Performance Evaluation of multi-hop Routing in Software Defined Network

Thesis submitted in partial fulfillment of the requirements for

the award of the degree of

Master of Engineering

In

Computer Science and Engineering

Submitted By

Paluck

(801632035)

Under the supervision of
Dr. Sushma Jain
Assistant Professor

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

THAPAR INSTITUTE OF ENGINEERING AND TECHNOLOGY
PATIALA – 147004

JUNE 2018

CERTIFICATE

I hereby certify that the work which is being presented in the thesis entitled, **"Performance Evaluation of multi hop routing in Software Defined Network"**, in partial fulfillment of the requirements for the award of degree of Master of Engineering in *Computer Science and Engineering* submitted in Computer Science and Engineering Department of Thapar Institute of Engineering and Technology, Patiala, is an authentic record of my own work carried out under the supervision of ***Dr. Sushma Jain*** and refers other researcher's work which are duly listed in the reference section.

The matter presented in the thesis has not been submitted for award of any other degree of this or any other University.

Signature

(Paluck)

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

(Dr. Sushma Jain)
Assistant Professor
Computer Science and Engineering Department

ACKNOWLEDGEMENT

The successful completion of any task would be incomplete without acknowledging the people who made it possible and whose constant guidance and encouragement secured the success.

First of all, I wish to acknowledge the benevolence of omnipotent God who gave me strength and courage to overcome all obstacles and showed me the silver lining in the dark clouds with the profound sense of gratitude and heartiest regard. I express my sincere feelings of indebtedness to my guide **Dr. Sushma Jain** for their positive attitude, excellent guidance, constant encouragement, keen interest, invaluable co-operation, generous attitude and above all their blessings. She has been a source of inspiration for me.

I am grateful to **Dr. Maninder Singh**, Head of Department and **Dr. Ashutosh Mishra**, P.G. Coordinator, Computer Science and Engineering Department, Thapar Institute of Engineering and Technology for the motivation and inspiration for the completion of this thesis. I will be failing in my duty if I don't express my gratitude to **Dr. S.S. Bhatia**, Senior Professor and Dean of Academics Affairs in the institute for making provisions of infrastructure such as Library facilities, Computer Lab equipped with internet facility immensely useful for the learners to equip themselves with latest in the field.

Last but not the least I would like to express my heartfelt thanks to my parents and my friends who with their thought provoking views, veracity and whole hearted cooperation helped in doing this thesis.

Paluck

(801632035)

ABSTRACT

From the origin of the Internet, data transmission over the network uses routing algorithms to get some directions. The novel path labeling techniques does not follow any multipath methodologies. In case of any busy event, the sender node must wait before forwarding data to the destination node. This results in higher delay and lesser throughput. To overcome these types of problems, an optimal routing based on certain network parameters is proposed to compute the cost for every routing cycle. This method also adopts multipath in case of any congestion or busy network, which avoids network performance to degrade. Software Defined Networks (SDN) are intelligent networks where the routing decisions are not based on some traditional algorithms in which different sizes of networks are designed using different network topologies and devices. The cost factor is computed each time while routing the data from source to destination. The goal of routing is to achieve the minimum cost path. Various topologies like spanning tree, ring tree, and dual tree are evaluated on the basis of factors like bandwidth usage, cost, and link efficiency. The main focus is to evaluate the performance of various topologies in terms of routing. The SDN routing is the solution to handle all the drawbacks of traditional routing protocols. The performance of spanning tree, ring and dual tree are compared with each other and the outcomes of spanning tree are better than a ring and dual tree in terms of cost. The bandwidth is more efficiently utilized in spanning tree, but the links are heavily used in spanning tree, which results in some chances of congestion. The average cost of the dual-tree is better than the performance of spanning tree.

TABLE OF CONTENTS

LIST OF FIGURES

Figure No.	Description	Page No.

LIST OF TABLES

Traditionally, the information packet had to follow the fixed path whereas SDN (Software Defined Network) provides the numerous paths to perform a single operation and hence the dynamic property can provide the continuous connection throughout as shown in Fig 1.1. For Example: - when one end is calling another end over the network then check the routing table for IP port and MAC. When it is connected and then the packets are being sent. When the path is closed with traffic and then connection gets cut and it is disconnected because the switch doesn’t have any programmability. Therefore, the rules cannot be changed dynamically and the connection is terminated. On the other hand, in SDN, Switch is connected to a controller which controls the action of the switch and the controller can be programmed dynamically which control the action of the switch and giving programmability to switch. Switch contact the controller to perform the action of the switch and will not do anything on its own. So, when the connection gets cut, then SDN switch receives an appropriate command from controller and SDN switch now takes an alternate path as per controller instruction and re-route the packet and they can talk over the network without any jitter.

OSPF (Open Shortest Path First) protocol periodically sends local link state information to all the other switches in the network. Collected information is being processed by software running on switch CPUs to calculate global routes. When a network link is to be failed, the protocol propagates updated link state information to calculate the new another set of routes within seconds. In the past few decades, routing which is reliable, widely adopted by the internet users. The main reasons for this are that the protocols themselves are implemented as classified software with thousands of lines of code written by multiple vendors. These implementations were proprietary and network operators neither access the internals for visibility, nor could modify their implementations to achieve the customized routing goals. Traditional network switches and programmed switches are separated by the functional components. Programmable switches discover the network, routes the network using distributed protocols in the control plane and use the routes which are to be computed to forward packets in the fast data plane.

The operator neither involves a high-level abstract view of the network nor she has a high-level interface to configure routing directly at that level of abstraction. Therefore, the summarize network behavior was very opaque to the network operator to debug deviation from expected behavior (like packet losses, routing loops etc.) which in turn made it difficult to either change this behavior (to achieve flexibility of routing packets based on different objectives, like traffic engineering etc. In the recent years, the rise of problems that are lack of visibility and flexibility geo-distributed data centers are elaborated. The increasing demand on data centers workloads that meant that the data center network had to be fast enough to include ever increasing compute capacity on short notice. In addition, the changing nature of data center workloads meant that route data traffic around them at rapid timescales in order to fully exploit network capacity without cost prohibitive over-provisioning and the operators had to quickly identify bottleneck network link. It means that network operators were willing to trade these protocol implementations for alternative architecture which is more manageable i.e., those that solve the twin problems of lack of visibility and flexibility that are Traditional Networking and Software-Defined Network.

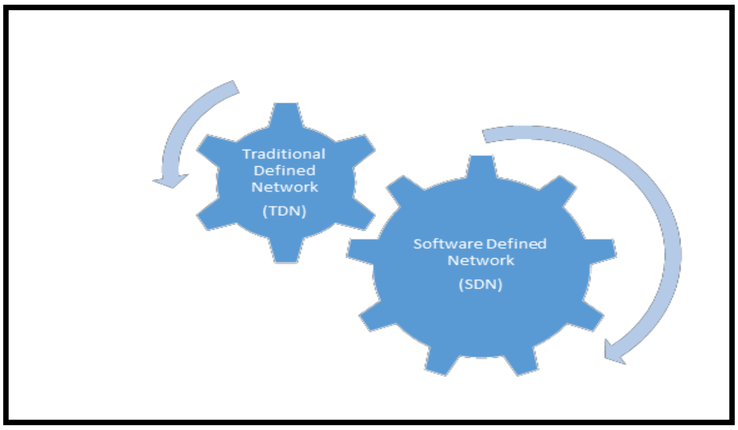

Figure 1.1 Evolutions of Networks

1.1 Software-Defined Network

Software Defined Network (SDN) is a system design where the control plane is decoupled from information plane and that is specifically programmable. The worldview of Software-Defined Networking (SDN) proposes neatly isolating the control plane from organizing switches into a centralized server known as the controller. That simply forward the packets in the data plane which is switches using commands sent by the controller. In addition, the switches may even send the events to the controller regarding the arrival of flow counters and the specific packets etc. The controller typically sends the commands to the switches in response to those events. This decoupling of the control plane and forwarding plane is the key to better the network management in SDN as compared in Table 1.1. There are three important architectural that changes the separation enables. First, the operator control of the entire network from one place, the controller or poll each switch independently using different interfaces get a centralized view instead of having to configure. Second, switch functionality is abstracted into a much simpler match-action forwarding plane model (which dictates how to classify packets based on their header match) instead of the code complexity that comes with running distributed protocols. Third, the separation and simplification of the forwarding plane enable a simple, vendor-agnostic control interface, unified and like Open Flow protocol across a heterogeneous set of network elements from the different equipment, vendors are included. Based on all the above architectural changes, SDN includes two important properties efficiency and flexibility. Instead of the network operator can use the central controller to customize the network behavior according to her needs and tweaking these protocols to indirectly do her bidding. This means that she can flexibly route the network traffic on some specific paths. She can perform different actions on packets like forwarding the packet, header modifications without the use a separate protocol for each such function. In addition, the visibility of the global network and a control interface across multiple devices makes for more efficient decision making. Overall, today SDN is seen as an alternative to traditional protocol implementations because of the increased visibility and flexibility. For example, Google's Software-defined WAN called B4 is a result of the frustration with rigid. One of the key appeals of B4 for the Google was the ability to efficiently change control at software speeds, ability to simply rely on control software testing before ensuring that the network behaves as expected when deployed in production and ability to rely on a central control software that can be flexibly customized to their needs (e.g., a WAN traffic engineering solution). Over the years, in addition to large service providers like the SDN paradigm is also adopted by switching vendors like Cisco, Juniper, etc. Google who built their own SDN switches, which started rolling out support for Open Flow-style control interface to their switches.

Table 1.1 Traditional Network v/s Software Defined Network

Traditional Network	Software Defined Network
Traditional networks are inflexible and static networks. Which are not useful for business venture They possess little flexibility.	These are programmable networks during development time which can help new business ventures through flexibility.
Traditional network work using protocols.	Software Defined Network uses API's to configure as per need.
Traditional networks have distributed control plane.	Software Defined Network have a logically centralized control plane.

1.2 SDN Architecture

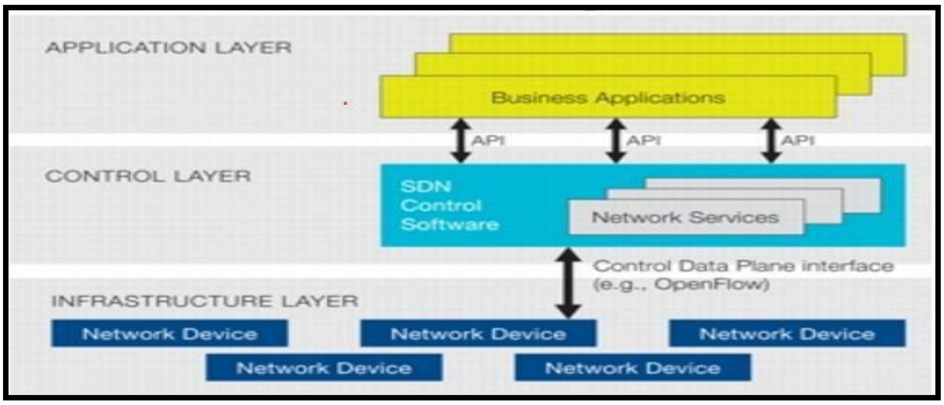

Figure 1. 2 SDN Architecture

In SDN Architecture there are three layers like a framework layer, control layer, the application layer as shown in Figure 1.2. In the framework layer, it is an establishment layer that comprises of both physical and virtual system gadgets, for example, switches and switch. All the system gadgets will execute the open stream convention to actualize movement sending rules. In Control Layer comprise of unified control plane that is decoupled from physical foundation to give brought together view to the whole system the layer will utilize open stream convention to speak with framework layer. In Application Layer comprise of system administrations and application and organization instruments that are utilized to associate with the control layer. It furnishes an open interface to speak with another layer in engineering. SDN components are: -

- **Control Plane**: Control plane provides the abstract view of the network
- **North-bound Application Protocols**: It represents the software interface between the controller and applications.
- **East-West Protocols**: It acts as a control that manages the interaction between the multi-controller environments.
- **Data Plane and south-bound protocols**: It forwards the communication from the controller to the network infrastructure.
- **Open flow**: Open flow Protocol which manages the southbound interface of the SDN architecture.

SDN is characterized by two distinguished features: -

- Decoupling of control and data plane.
- Programmability on the control plane

1.3 Advantages of SDN

- **Reduced working expenses:** It is brought together overseen; operational effectiveness and which has better equipment utilize all empower diminished expenses.

- **Reduced equipment administration and expenses:** We may expand the life of existing equipment since we have moved the choice of the SDN controller.

- **Enhanced security:** Management made more perplexing in view of virtualization, having a brought together system controller which can take the reins of control back to its hand's experts' and make a control of the main issue to disseminate security and arrangement data reliably all through the undertaking.

- **Centralized organize provisioning:** SDN can adjust the heap and appropriate the activity all the more productively to keep the chokepoints, which can enhance the application execution.

1.4 Applications and Services of SDN

- ***Security Services:*** Within the network layer, the current virtualization ecosystem supports specific virtual services and incorporating functions like an environment which is capable of minimizing risk and responding to incidents much more rapidly. When a violation occurs, every second is analytical in stopping the attack and the requirement is the capability to find the attack which protects the other network components. As the network layer is more analytical and as the modern organization becomes even more computed. We check more sophisticated advanced persistent threats and more attacks. Powerful security services are integrated into the SDN layer which helps to create an environment that is capable of responding to alter.

- ***Network Intelligence and Monitoring:*** Within the data center, SDN technologies are serving one of the most critical layers. The architecture of the Network is more complex and have to handle more data which mean to know what is flowing through the environment. All of these challenges are modest when we have a network intelligence layer. However, we integrating these technologies into the SDN architecture. Hypervisor integration, Traffic flow, port configurations alerting and even optimization can be integrated into monitoring technologies and network intelligence. Most of all, these parts of agile systems which will further help to monitor network traffic between the cloud ecosystem and data center.
- ***Compliance and Regulation-Bound Applications:*** Major cloud vendors are offering the capability to work with compliance-bound workloads. Now, these organizations have the option of extending architectures which were originally very limited because of regulations into distributed environments. This is where SDN can help. Network traffic traveling between network points, switches, and even hypervisors can all be controlled in SDN architecture. This layer abstracts virtual functions and hardware controls. This powerful layer can then span various locations, virtualization points.
- ***High-Performance Applications:*** Virtualization has allowed the delivery of rich applications like graphic design software and CAD engineering. Traditionally, these workloads needed bare-metal architectures with their own connection. However, with virtualization, applications are streamed and VDI can help to create powerful desktop experiences. At the network layer, we also see the integration of SDN into application control. Segmenting heavy traffic, creating powerful QoS policies, securing confidential data, and even creating

threshold alerts. All of these functions within SDN can help to support high-performance, rich applications which are being delivered with virtualization.

- ***Distributed Application Control and Cloud Integration:*** One of the best advantages of SDN is its capacity to stretch out over the whole server farm. This kind of readiness incorporates circulated areas, cloud, and the whole association. SDN takes into account basic system activity to go between different areas, paying little heed to the kind of fundamental system design. By abstracting basic system controls you take into consideration the less demanding development of information between server farm and cloud areas. Since SDN is a type of system virtualization, we can utilize intense APIs to not just incorporate with a cloud supplier we can control particular system benefits also. This enables you to granularly deal with your workloads while keeping your business deft.

1.5 Open Challenges

While SDN is a paradigm for the network design, the practice of SDN is included with many challenges. When arising some of the challenges due to architectural shifts forcing a rethink about how specific objectives like efficiency should be achieved, some are due to deployment issues with the new promises like flexibility. These challenges guarantee a relook at the best practices for networking at many levels in an SDN control plane, forwarding plane, and the intertwining control interface. In this section, we focus on three important aspects of SDN that needed a fresh study recently control interface reliability, control plane flexibility, and forwarding plane efficiency.

- ***Efficient Network Utilization:*** Networks have included multiple paths which are connecting to the network endpoints to avoid congestion when running at high utilization. Datacenter networks have multi-rooted tree topologies which are included with more equal cost paths that must be used effectively in order to achieve the high bisection bandwidth. This is the reason administrators ordinarily utilize arrange stack adjusting plans that spread approaching activity stack on the numerous ways in the system. Generally, even before the coming of SDN, administrators utilized a heap adjusting system called square with cost multi-way steering (ECMP), which spreads activity by relegating each stream to one of a few ways at arbitrary. Be that as it may, ECMP experiences the corrupted execution if two long-running streams are relegated to similar way. ECMP additionally doesn't respond well to interface disappointments and leaves the system underutilized or congested in unbalanced topologies. The paradigm of SDN gives a new opportunity to look at data plane efficiency by using the controller to deploy sophisticated traffic engineering algorithms. In particular, a controller can exploit global visibility into the congestion levels on various paths (say, by polling switch flow counters) and then can push commands to switches that steer traffic along multiple paths optimally. For example, schemes such as use switch monitoring techniques to collect flow counters solve the traffic engineering (TE) problem centrally at the controller given this information and then push routing rules to switches. However, compared to ECMP, which would make load balancing decisions in the data plane at line rate, control plane timescales are too slow to implement load balancing. Example Switch Rule Table efficiently uses the available network capacity. The controller-based centralized TE schemes take on the order of minutes to react to changing network conditions, which is too slow for networks running volatile traffic. For example, in data centers, most flows are short-lived, interactive, mice flow whose lifetimes span a few milliseconds. In this case, the flows either have to delay until the central TE decision is enforced or simply use stale paths in the data plane, both of which adversely affect application responsiveness. This warrants a new look at designing effective load balancing schemes for SDN that are responsive to volatile traffic at data plane timescales in order to be effective and adhere to SDN principles of visibility and flexibility without resorting to opaque and rigid vendor-specific ASIC implementations.

- ***Flexibility:*** As mentioned earlier, SDN achieves one of its fundamental objectives of flexible control by sending prioritized match-action rules to the switches using an interface like Open Flow. The match part describes the header pattern of packets that should be used to process this rule. The corresponding action may be either to forward the packet out of a switch port or to change certain header fields, or to drop it entirely, etc. The rules are prioritized in order to disambiguate when a packet matches multiple rules in the table. It shows an example Open Flow table with prioritized rules having match patterns and corresponding actions. In the cutting edge equipment switches, rules which are put away in exceptional equipment memory called

Ternary Content Addressable Memory (TCAM) which can contrast an approaching bundle with the examples in the greater part of the principles in the meantime. Be that as it may, switches bolster generally few principles, in the little thousand. This is a request of greatness not as much as the ordinary number of sending rules pushed to switches today. For instance, around 500k IPv4 sending rules are put away in switches of the web future switches will bolster bigger control tables, however, TCAMs still present a major exchange off between manage table size different concerns like power and cost. TCAMs present around 10 times more prominent power utilization, contrasted with traditional RAM. Refreshing the principles in TCAM is a moderate procedure the present equipment switches which could without much of a stretch compel an extensive system with dynamic approaches. Therefore, commodity SDN switches have limited space to enforce fine-grained policy rules which undermines the promise of flexible control. The challenge here is to come up with a solution that works transparently with current controllers and switches without having to wait for the next generation of advances in switch hardware.

- ***Reliability**:* Traditionally, network operators had to do a lot of work to manually configure various routers in the network with a myriad of parameters to run distributed routing protocols. However, once configured, switches running these protocols can discover link failures and automatically adjust routing around them. In contrast, the centralized controller in SDN is a single point of failure, which is unacceptable. Now, they need to understand and deal with the myriad issues related to consistent replication and fault-tolerance of controller instances in order to implement a logically centralized controller. Additionally, this cannot simply deploy traditional software replication techniques. Steady controller keeping upstate is just piece of the arrangement; one should likewise guarantee that the switch state is taken care of reliably amid controller disappointments to give unified. This is on account of the switch has state identified with coordinate activity administer tables, bundle supports, connect disappointments, and so forth. Frameworks don't reason about this switch state; they have not thoroughly examined the semantics of handling switch occasions and executing switch charges under disappointments. For instance, while the framework could move back the controller express, the switches can't without much of a stretch "move back" to a protected checkpoint. All things considered, what does it intend to rollback a bundle that was at that point sent. The option is for the new ace to just rehash summons, yet these orders are not really idempotent. Since an occasion from one switch can trigger summons to different switches, the synchronous disappointment of the race controller. If these issues are not carefully handled, the network can witness erratic behavior like performance degradation. In the meantime, running an agreement convention including the switches for each occasion would be restrictively costly, given the interest for fast bundle preparing in switches. Then again, utilizing appropriated capacity to reproduce controller state alone (for execution reasons) does not catch the switch state accurately. Consequently, after a controller crash, the new ace may not know where to continue reconfiguring switch state. Essentially perusing the switch sending state would not give enough data about every one of the charges sent by the old ace (e.g., Packet Outs, Stat Requests). Given this, the test is to manufacture a blame tolerant controller runtime that deals with ensuring value-based semantics to the whole control circle that involves gathering events, processing them and the subsequent issue of commands. This way, even under failures, the physically replicated control instances behave as one logical controller. An additional challenge is to remove the burden from the network operator of having to reason about failure and consistency issues and instead write control programs for just one controller while the runtime takes care of correctly replicating it to multiple instances. While there are several challenges plaguing the current implementations of SDN, there is little doubt that the basic architectural vision behind SDN is sound and desirable in practice. In this section, we discuss how we can take advantage of recent advances in hardware and software data planes and past techniques from replicated state machines to tackle the three challenges mentioned earlier — efficiency, flexibility, and reliability. There are several opportunities for handling challenges.

- ***Programmable Data planes for Efficient Utilization*** used in order to have efficient data plane forwarding that exploits multiple network paths, we need to be able to infer global congestion information and then be able to react to it at data plane timescales. This means we need the ability to export and process link level utilization information in the data plane itself instead of using the switch CPU. In addition, we need the ability to dynamically split traffic flows at fine granularity (in order to avoid adverse effects of collisions) and route them instantaneously based on the previously gathered information. In this context, the recent rise of programmable hardware data planes fits our requirements perfectly. As opposed to a programmable control plane which dictates which rules to send to switches, programmable data planes allow the operator to specify how hardware resources like TCAM, SRAM, packet buffers, etc. should be distributed into multiple tables and registers in a packet processing pipeline. This way, the operator can not only customize the match-action rules but also every stage of the packet processing pipeline in the data plane. This results in a data plane that provides a sophisticated hardware platform that can be customized 'in the field' for a wide variety of data plane algorithms without having to wait for vendor approved ASIC upgrades. In a programmable data plane, the switch consists of a programmable parser that parses packets from bits on the wire. Then the packets enter Programmable data plane model ingress pipeline containing a series of match-action tables that modify packets if they match on specific packet header fields. The most important aspect of the model that is of interest to us is that each table can access stateful memory registers that can be used to read and write state at line rate. This feature can be used to export network utilization values onto packets flowing through a switch. The neighboring switches that receive this packet can then store this information in their own local memory and use it to decide where to send the next set of packets flowing through them. In this thesis, we will try to exploit such data plane architectures that allow for global congestion visibility and stateful packet processing entirely in the data plane. This will mean much faster reaction times for a load balancing scheme, on the order hundreds of microseconds, which matches the round trip time in modern data center networks. At the same time, we aim to configure the data plane using a vendor-agnostic programming interface that can customize a heterogeneous set of data plane targets in a way that adheres to the basic principles of SDN - visibility, and flexibility.

- ***Software Switching for Fine-Grained Policies*** Limited TCAM availability in switch hardware leads to difficulties implementing a truly flexible control policy. Luckily, the movement has a tendency to take after a Zapf appropriation, where by far most of the activity coordinates a moderately little division of the standards. Subsequently, we could use a little TCAM to forward by far most of the activity, and depend on elective information ways for handling the rest of the movement. Late advances in programming exchanging give one such an appealing option. Running on product servers, programming switches can process parcels at around 40 Gbps on a quad-center machine and can store vast govern tables in primary memory and (to a lesser degree) in the L1 and L2 reserve. Likewise, programming switches can refresh the run table in excess of ten times quicker than equipment switches. But, supporting trump card decides that match on numerous header fields is saddling for programming switches, which must fall back on moderate preparing in client space to deal with the primary bundle of each stream. Thus, they can't coordinate the "torque" of equipment switches that give many Gbps of bundle handling (and high port thickness). In this manner, in view of the Zipf idea of the measure of movement coordinating switch rules, we will endeavor to utilize a blend of equipment and programming changing to bring the better of the two universes – high throughput and expansive control space. In addition, we need to carefully distribute one rule table into multiple tables spanning across heterogeneous data paths so that the semantics of the single-switch rule table are preserved in the distributed implementation.

- ***Replicated State Machines for Reliable Controlling*** order to provide reliable control in the face of controller failures, we need to design failover protocols that not only handle controller state but also the switch state. In addition, we need to provide a simple programming abstraction where the network operator need only write a program for a single controller while the runtime manages proper replication. 11 In this context, a natural choice for such a simple abstraction is that of a replicated state machine where a single state machine is replicated across multiple physical instances for fault tolerance. An example protocol that implements such an abstraction is the View stamped Replication protocol, a replication technique that handles crash failures using a three-stage request processing protocol and a view change protocol that on node failures, depends on a quorum to reconstruct the committed requests. In the case of SDN controller failure, we need to adopt such replicated state machine protocols for control state replication and then add mechanisms for ensuring the consistency of the switch state. In particular, we need the protocol to provide transactional semantics for the entire control loop that is triggered for each network event: event input replication, event processing at each instance and executing resulting commands at the switch. Instead of involving all switches in a consensus protocol, we need to design a lightweight replication protocol that keeps the overhead on the switch runtime low while ensuring the correctness of the transactional semantics. These are some needs of SDN:

- **Orchestration**: Should be proficient to control and deal with a large number of gadgets with one charge.
- **Programmable**: Should be proficient to change the conduct on the fly.
- **Dynamic Scaling**: Should be skilled to change size and amount.

Some of the initial ideas in this topic deal with the issue of understanding the different kinds of topologies that can be implemented like:

Bus topology: Bus topology is a network type in which every network device and the computer is connected to single cable and it has exactly two endpoints is called Linear Bus topology as shown in Figure 1.3. If the network traffic is heavy or nodes are more than the performance of the network decreases.

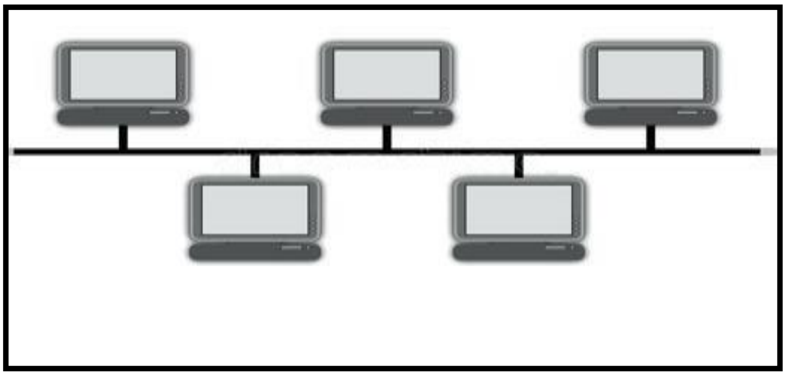

Figure 1.3 Bus Topology

Ring topology: It forms a ring as every computer is connected to another computer and with the last one connected to the first. Two neighbors are exactly for each device. Failure of one PC irritates the entire system as shown in Figure 1.4

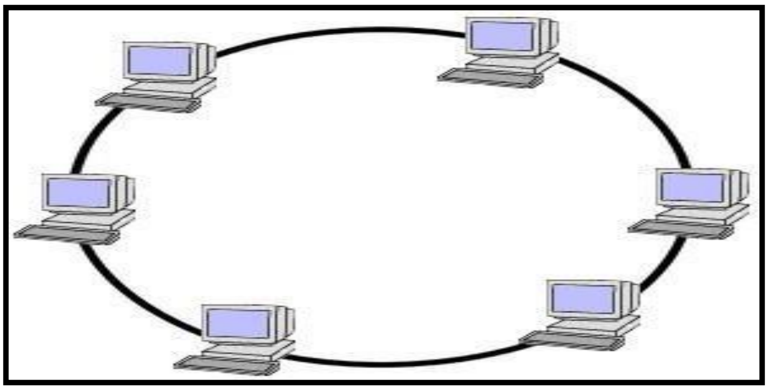

Figure 1. 4 Ring Topology

Star topology*:* In which all the computers are connected to a single hub through a cable and that hub is the central node and all others nodes are connected to the central node is called star topology as shown in Figure 1.5. If the center hub damage, then the entire system is ceased on the grounds that every one of the hubs relies upon the center.

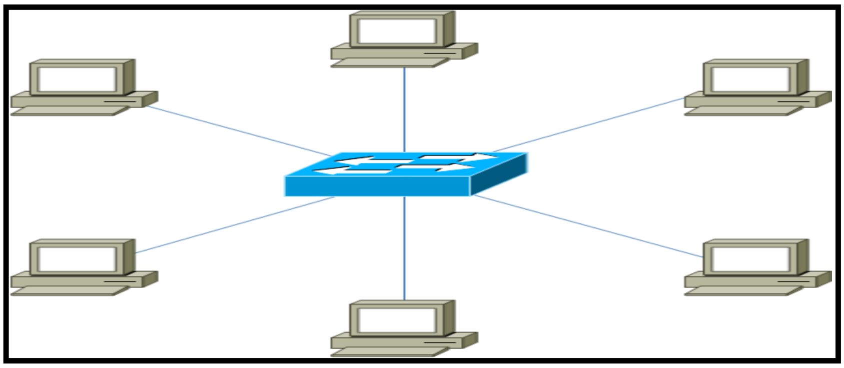

Figure 1. 5 Star Topology

Mesh topology: It is a point-to-point connection to other nodes or devices. All the network nodes are connected to each other is called mesh topology. Installation and configuration are hard as shown in Figure 1.6.

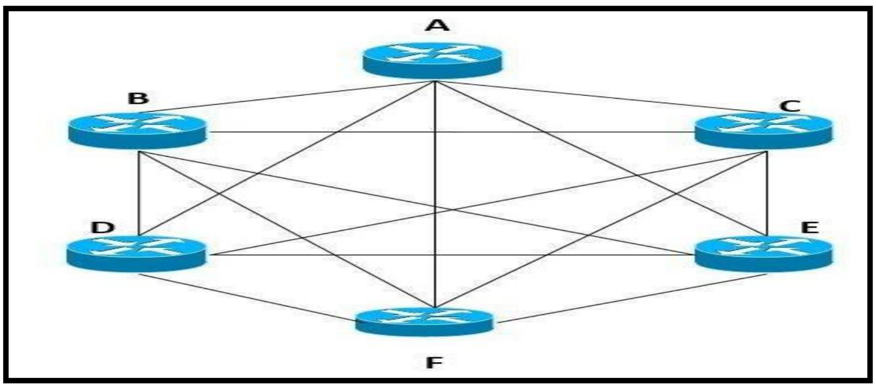

Figure 1. 6 Mesh Topology

Tree topology: It has a root node and all other nodes are connected to it forming a hierarchy. It is also called hierarchical topology as shown in Figure 1.7. It should at least have three levels to the hierarchy. If more nodes are added maintenance is hard.

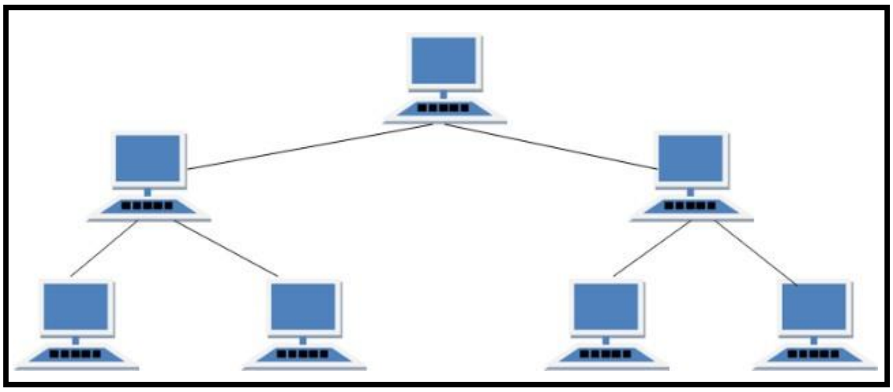

Figure 1.7 Tree Topology

1.6 OPNET

OPNET gives numerous builds identifying with interchanges and data handling, star viding high use for displaying of systems and circulated frameworks. OPNET gives a Graphical determination of model wherever conceivable, models are entered by means of graphical editors. These editors give a characteristic mapping from the exhibited structure to the OPNET show assurance. OPNET gives four gadgets rang editors to assemble another reenactment appear. These editors, the Network, Node, Process and Parameter Editors, are dealt with in a dynamic form, which reinforces show level reuse. Models created at one layer can be used by another model at a higher layer. All OPNET multiplications, therefore, join support for examination through a progressed natural debugger.

OPNET (Optimized Network Engineering Tool) gives an exhaustive change condition supporting the showing of correspondence orchestrates and passed on structures. Both direct and execution of showed systems can be dismembered by performing discrete event amusements. The OPNET condition joins instruments for all periods of an examination, including model plan, reproduction, information accumulation, and information investigation. OPNET is occasion based (discrete time) organize reproduction device. The product is effective yet, in addition, requesting at any rate for the designers. Using just reenactments is very direct. Demonstrating is done for the most part in three levels Network Domain (top level) and Upper levels shroud the unpredictable structure of the lower level segments, which can be absolutely undetectable to the end client initially is Node Domain and other is Process Domain .Different layers and functionalities can for all intents and purposes be displayed as precisely as required with outer model access (EMA) and outside framework area (ESD) functionalities.OPNET incorporates different extra modules and apparatuses for facilitating the use of the product.

A typical introduction of the OPNET test system (OPNET Modeler) is given. OPNET is expansive and capable programming with a wide assortment of potential outcomes. Empowers the likelihood to reproduce whole heterogeneous systems with different conventions. Advancement work was begun in 1986 by MIL3 Inc. (these days OPNET Technologies Inc.) Originally the product was created for the requirements of military, however, it has become a world driving business to organize reproduction apparatus OPNET is an abnormal state occasion based system level reenactment device. Recreation works at "bundle level" Originally worked for the reproduction of settled systems. OPNET contains a colossal library of exact models of economically accessible settled system equipment and conventions. These days, the potential outcomes for remote system reproductions are likewise wide. Exact radio transmission pipeline arranges for the demonstrating of the physical layer (radio interface). The test system has a ton of probability, however, there exists regularly an absence of the ongoing remote frameworks. A great part of the work considering new advances must be finished without anyone else. OPNET can be utilized as an exploration apparatus or as a system plan/investigation instrument (end client). The limit for the use is high for the engineer, however low for the end client. OPNET comprises of abnormal state UI, which is developed from C and C++ source code hinders with a colossal library of OPNET particular capacities. Hierarchical structure, demonstrating is partitioned into three principle areas Network space, Networks, sub-systems, organize topologies, land facilitates, versatility Node Area Single system hubs e.g., switches, workstations, cell phone. Process area, Single modules and source code inside system hubs (e.g., information movement source display) With OPNET it is additionally conceivable to run outside code segments (External System Domain, ESD.

Chemerinsky [1] had proposed that the test platform with the software-only simulation, this exploration proposed another equipment tuned in imitating: to copy an SDN and utilized the System-In-The-Loop (SITL) innovation of OPNET to construct the SDN equipment insider savvy copying stage. The imitating test stage incorporates the sensor organize part, all of which will utilize the product recreation, together with the transmission arrange and the cloud server farm both of which will utilize the genuine system. This research demonstrates that the recreation test stage of equipment on the up and up can accomplish vitality productivity under the request of QoS.

Ramdhani *et al.* [2] had done a model reflection strategy that effectively changes the system gadgets in an SDN-based system to one virtualized switch show. While significantly lessening the model execution time and empowering the constant reenactment capacity, our preoccupied display additionally protects the conclusion to-end sending conduct of the first system. To accomplish this, we first characterize bundles with a similar sending conduct into littler and disjoint Proportionality Classes (ECes) by dissecting the Open Flow rules introduced on the SDN gadgets. We at that point make a diagram show speaking to the sending conduct of every EC. By navigating those diagrams, we finally build the principles of the enormous change model to effectively save the first system's end-to-end sending conduct.

Chih *et al.* [3] had discussed the algorithm that Emulators and test systems give a simple method to lessen equipment needs in tests. Hence, arrange analysts utilize applications that enable them to imitate or reproduce systems, as Opnet in Software Defined Networks. In this research and looked at the exploratory outcomes gained when a virtual framework is made by using Opnet versus a certified executed framework which had taken a gander at them moving the Maximum Transmission Unit (MTU) on Internet Protocol shape 4 (IPv4) packages. Ethernet, Fibres Distributed Data Interface (FDDI), and Wireless Local Area Network 802.11 (WLAN 802.11) MTUs have been utilized as a part of our exploratory tests and worked with various connection capacities and produced movement with various transfer speed.

Izumi *et al.* [4] implemented an adaptive multipath routing scheme for the disaster-resistant storage systems. In this scheme which select the multipath routes dynamically based on network resources under the disaster situations for active data transmission. In this research, which shows the implementation and design of this scheme based on the Software Defined Network (SDN), which evaluate its effectiveness through the various simulation experiments.

Fu *et al*. [5] suggested that the fractional disjoint multipath steering has been to a great extent sent in systems. For example, remote sensor systems, adaptation to internal failure and beating single way directing in stack adjust and so forth. This research thinks about the SDN as the plans and actualizes disjoint multipath directing, inquire about the foundation and contrast it with the conventional single brief way steering calculation. It additionally depicts that the differently preferred standpoint of disjoint multipath directing in part originates from the heap furthest reaches of a connection in systems.

Jo *et al.* [6] had designed the fat tree topology with multipath ability has been utilized as a part of different new server farm systems (DCNs) for adaptation to internal failure and expanded data transfer capacity. Customary steering conventions can't completely use the gave transmission capacity in that sort of systems and have some help for multipath directing a handy arrangement, which exploits the rising programming characterized organizing worldview. The calculation depends on a local controller to gather essential system state data keeping in mind the end goal to settle on an improved directing choice.

Pascal *et al*. [7] had done the Machine Learning Techniques (MLT) in Software Defined Networks (SDN). Applications created by the gadgets are various in nature, for every application data transmission and defer prerequisites change. The streams in the system vie for an obliged asset, for example, data transmission or low inertness way, a shrewd stream steering calculation turns into a characteristic request.

Oliveira *et al.* [8] had evaluated to assess the SDN copying device called Opnet After playing out a few tests and recreations, deduced in that research that regardless of the restrictions of the device in connection to the constancy of execution between the reproduced and the genuine condition, the Opnet instrument has incredible significance for SDN look into. In addition, the limit of fast and improved prototyping, the immaterialness security, the likelihood of sharing outcomes and instruments at zero cost are certain variables that assist researchers to help in their exploration.

Fontes *et al*. [9] had done new means for arranging virtualization and programmability propelling the way systems can be composed and worked, including client characterized includes and tweaked conduct even at run-time. In any case, implies for quick prototyping and trial assessment of SDN for remote conditions are not yet accessible. This examination presents Opnet-Wi-Fi as a device to copy remote Open Flow/SDN situations permitting high-devotion tries that reproduce genuine systems administration conditions.

Keti *et al*. [10] had considered the use of Opnet to model the behavior of SDN networks which provided a brief tutorial on the installation and use of Opnet in a virtual machine (VM) environment and discuss the requirements for installing the Open Daylight open source SDN controller. Scalability is discussed, both in terms of host server limitations and virtual network response times and provided a framework for using this approach to create and simulate SDN networks for both educational and research purposes.

Godanj *et al*. [12] had proposed the SDN is a stage for quick system prototyping. It can run unmodified system application code on little systems and additionally extensive systems. It is another option to run SDN probes copied systems. Genuine frameworks are exceptionally excruciating to reconfigure. Test systems are a decent option however same source code can't be sent on genuine equipment. There are execution issues on Opnet. The difficulties previously Opnet are to demonstrate systems of substantial scale with common sense execution.

Keti *et al.* [13] implemented the scope of development in networking. The customary systems administration design can't oblige the present necessities proficiently. The increment of cell phones, virtualization, security, proficient Big-Data administration and high caliber of administrations has lead SDN as a promising engineering. Likewise, its effect is expanding at a sharp rate step by step. As PC organize become bigger and more perplexing, there is a requirement for another straightforward sort of approach to configuring them. SDN has risen as a promising system design.

Qilin *et al*. [14] proposed an assessment of an SDN imitating apparatus called Opnet is led. Tests were led to contemplate Opnet constraints identified with the recreation condition, asset capacities. The execution of Opnet apparatus for copying SDN systems was assessed. During this examination numerous capacities of Opnet emulator in the

SDN worldview assessment was secured, from the formation of fundamental topologies with reference controller to the capacity of association with remote controllers.

Wang [15] had compared and evaluated the correctness, performance, and adaptability of EstiNet Open Flow simulator, EstiNet Open Flow emulator, and Mininet Open Flow emulator over an arrangement of matrix systems. The well-known Floodlight Open Flow controller is utilized with no alteration to control the re-enacted/copied Open Flow switches made in these instruments. With respect to copying and found that Opnet emulator created unusual outcomes that can't be clarified over some system sizes.

Hu *et al*. [16] recommended a heap adjust arrangement plot by taking the advantages of the worldwide system perspective of SDN. Highlights incorporated into that examination are transmission dormancy, transmission jumps data transfer capacity usage proportion and parcel misfortune rate. As the same with the customary system, way excess innovation in SDN viably gives the vigor and steadiness to arrange. This procedure gathers data transfer capacity usage proportion, transmission bounces, parcel misfortune rate, transmission inertness and incorporate them by Artificial Neural Network to demonstrate the way stack condition.

Nam *et al.* [17] implemented the MPTCP (Multi-Path TCP) helps application organize execution by gathering the data transmission in various ways. In any case, it might cause terrible execution because of the enormous number of out-of-arrange bundles particularly when the ways have distinctive deferrals. To determine this issue, then suggested that powerfully include or erase MPTCP ways with the use of programming defined organizing (SDN). The fundamental thought is to track the accessible limit of associated ways and pick the most proper ways relying upon differing system conditions. To demonstrate the achievability of our approach, make an SDN stage utilizing Opnet.

Koerner *et al*. [18] proposed a type of load dispersion component on the in advance of said organize layers by utilizing a flow based load imparting method to layer two multi-ways and which will exhibit an approach, utilize case and execution assessment too. The fundamental point will demonstrate this is expanding the throughput in a rack-to-rack information trade situation.

Wang *et al.* [19] had designed the objective of this research is to design and implement a computing technique that improves energy efficiency at the Cloud Data Centre. An algorithmic-based approach is to minimize the energy consumption at the Data Centre. The proposed approach utilizes one of the most important technologies in the areas of namely Software Defined Networking (SDN) using Open Flow technology and the server virtualization research area.

Guillen *et al.* [20] implemented the Distributed Storage Systems (DSS) has considerably risen in the past years, alongside the need for effective data transfer from storage to storage. Even if the current network infrastructure can reliably handle the more amounts of traffic, networking techniques have not been changed for several years, leading to an under-use of resources. In this research, presented a pragmatic approach for multipath routing in DSS, which is based on Software Defined Networking (SDN) which uses parallel links at the edge-side.

Toufga [21] designed the vehicular systems are one of the foundations of an Intelligent Transportation System (ITS). These are required to give the system availability to moving vehicles while supporting the diverse ITS administrations, some with extremely stringent prerequisites as far as dormancy and unwavering quality. Those which consider both system innovations to offer the multi-way steering, way splitting, load adjusting or for a superior nature of the experience of ITS administrations expect clearly independently controlled systems.

Azzouni1 [22] had introduced a dynamic routing framework for software defined networks (SDN) entirely based on machine learning and Neural Networks. Currently, SDN/Open Flow controllers use a default routing based on Dijkstra's algorithm for shortest paths and provide APIs to develop custom routing applications. The Route that achieves the same results as the most efficient dynamic routing heuristic but in much less execution time.

Chavula *et al.* [23] had provided the different opportunities for dynamic and flexible traffic engineering. This research discusses how Education Networks (NRENs) and Ubuntu Net Alliance National Research which can reduce inter-NREN latencies through the implementation of SDN-based traffic engineering bandwidth utilization and apply network metrics in the selection of inter-NREN paths. Additionally, this research looks at the utility of applying Reinforcement Learning to path selection, using network data obtained through an SDN controller.

Singh *et al*. [24] had designed the software defined network, or then again SDN based systems are being sent in testbed systems, as well as underway system and adaptation to internal failure is a standout amongst the most attractive properties in the creation systems, there is very little investigation in giving adaptation to non-critical failure to SDN-based systems. The objective of this exploration is to build up a blame tolerant SDN engineering which can powerfully recuperate from the issues and scale to extensive system sizes. This examination exhibits the SDN blame tolerant framework which recoups from numerous connection disappointments in the information plane.

Rezende *et al.* [24] had designed the modules to an Open Flow controller and MP-Routing, which decides an arrangement of ways, proposed an arrangement of new counters to Per Port and Per-Flow granularity levels of Open Flows specification. Traffic is pushed through a the single way in spite of the presence of elective ways in systems. In this examination, which is occupied with breaking down the help for checking system movement and for provisioning of multi ways in programming characterized networking (SDN) given the solid stage it accommodates experimentation of new arranged arrangements.

Banfi *et al.* [25] had suggested that multipath sending, which comprise of answers for execution change, both at the collector side, by modifying out-of-arrange bundles in a devoted in-organize support and at the sending side through multipath booking calculation. In which executed a model with ordinarily accessible innovation and afterward assessed it in both genuine and imitated systems. Their outcomes indicate predictable throughput enhancements and utilize the accumulated way limit.

Rocha [26] had designed to help an alternate number of system setups through the arrangement of information ways between physical hosts for SDN. Computational demonstrating and improvement is a plausible other option to quantify the most various computational issues previously it's prototyping. Build up the imperative toolset called Mini-TE (Mini-Traffic Engineering) which perform activity designing over computational models of server farm topologies and need to assess the viability of our approach by utilizing Opnet through the different arrangement of investigations.

Gupta *et al.* [27] had developed the issue of prototyping and assessing new SDN-based applications precisely, everywhere scale, and in a way that empowers simple interpretation to genuine controller stages like POX and NOX and made the plan, execution and utilization of fs-sdn, and show its ability via doing a progression of trials utilizing fs-sdn and the Opnet stage in almost indistinguishable configurations.

Paolucci *et al*. [28] had suggested a novel component for the cutting edge optical systems and examinations the entanglement of optical light-way calculation. Any single light-way calculation does not act regularly under the distinctive movement situations, as every calculation/heuristic talk about the hunting space particularly. In this exploration had proposed a novel calculation determination philosophy for the new age of programming characterized the optical system. Optical systems administration was considered for both the parent and youngster calculation component. The fields that are coordinated to be utilized with the product characterized organizing were considered in outlining the calculation choice component.

Gupta [29] had described the different advantages of SDN for some system administration applications like traffic designing, benefit affixing, topology reconfiguration which includes the tending to of complex improvements which are integral to these intricacy that will probably illuminate the advance of SDN applications utilizing abnormal state deliberations for recovering the streamlining necessities from which can efficiently produce best arrangement which demonstrated the SOL which gives, the preferable adaptability over the custom enhancement answers for different applications.

Dilmaghani [30] had reviewed the research which requires a highly reliable, resilient, and secure network infrastructure. Efficient network management, lower cost, and smaller size are among other requirements. The need is well recognized and the U.S. Navy has invested in emerging technologies to explore how new technologies are developed which can improve the integration withstand the barrage of attacks. In this research, which explained how Software Defined Networking (SDN) which can improve the performance by providing the various access to the certain types of network traffic in the bandwidth constrained environments.

Aceto [31] had presented a novel instrument for estimating the data transmission in SDN systems which take the upside of the SDN design and assemble an application over the Network Operating System. Their application can give data transfer capacity usage over the system joins, which track the topology of the system, it can condense the accessible transmission capacity between any two focuses in the system which approved our strategy utilizing the famous Opnet organize copying condition utilized NOS called Floodlight.

Sandri *et al*. [32] had reviewed the centers around assessing the utilization of MPTCP and it is a system convention intended to forward packets through disconnected ways. Systems ordinarily utilized the Equal-Cost Multipath Protocol (ECMP) to part flows through particular ways which could enhance the throughput in shared bottlenecks by sending packets from a same transport convention association through numerous ways that could approve the approach in a tested where shared bottlenecks in the connection can happen 18.

Mijumbi *et al.* [33] had reviewed the state of the art in programming characterized organizing giving a verifiable point of view on integral innovations in natural deficiencies which gave the best approach to SDN. The SDN design was considered alongside prevalent protocols and the current reproduction. Programming characterized organizing (SDN) was one of the relatively new networking paradigms, simplify the network management by decoupling the control plane and the information plane and time network programmability.

Jimenez *et al.* [34] had discussed the Software Defined Networks (SDN) had become a new way to make different topologies. Some emulators were present like Opnet, which allowed an SDN in a single personal computer, but there was a lack of works which show its performance compared with real cases. In this research which shows a performance comparison between Opnet and a real network when multimedia streams were being delivered.

Pang *et al.* [35] had designed an efficient flow a novel booking approach which powerfully sets up ways for mice streams independently, in view of the current connection workload. Their distinction plans mice streams with a proactively introduced weighted multipath directing calculation which modifies the diverse way weight as per the different connection usage and which adjust the activity in an SDN system which enhanced the administration productivity and framework execution.

Patil *et al*. [36] developed the IP multicast which has been utilized to help the gathering of correspondence semantics in the different Internet-based disseminated applications, it's sending in cloud Data Centre Networks had been restricted because of its higher assets steadiness and versatility issues, which debases the utility of the cloud. Programming Defined Networking (SDN) had empowered the multicast capacities to defeat these impediments. In this exploration introduced a self-ruling, adaptable and powerful middleware arrangement called SDN-based Multicast (SDMC), which gave both systems stack mindful and switch memory-proficient gathering correspondence semantics in DCNs.

Zhou1 *et al*. [37] had suggested that a versatile and viable calculation for controller stack adjusting completely in view of circulated engineering, with no concentrated part. This calculation is running as a module of SDN controller. On one hand, it acknowledges a movable load gathering limit in order to diminish the overhead of trading messages for stack accumulation, and then again it can make race region and approach keeping in mind the end goal to reduce the choice postponement caused by organizing transmission. In this examination, made the model framework on floodlight to build up their plan and test the execution of our calculation.

Liao *et al.* [38] had implemented a dynamic load-adjusting instrument to decrease the reaction time by modifying rules in the SDN stream table. Their plan can suit the paired tree contingent upon the heap on every server. At the point when a server is out of limit, it could exchange the overabundance movement load to servers with low load. Recreations demonstrate that their plan adjusts server stack as well as expands the server utilization

Jiang *et al.* [39] had designed the fast inflation of a number of clients associated with the network, it is more important to assign the incoming request equally among all the servers. These traditional network platforms were very difficult to manage due to different challenges like interoperability, upgrade that require protocols and techniques hard-coded into the underlying network platform. This research gave the comprehensive critical survey on different load balancing strategies used in SDN technology.

Neghabi *et al*. [40] had developed all the reviews of the stack adjusting components that have been utilized as a part of the SDN reliably in view of two classes, nondeterministic and deterministic. Likewise, this exploration speaks to some benefits and shortcoming considered of the chose stack adjusting calculations. Be that as it may, the colossal difficulties of these calculations have been examined, so better load adjusting methods can be connected by these searchers in the future. Due to limit arrange assets and one of the focuses that must be considered is stack adjusting issue that serves to pass on information movement among various assets with a specific end goal to amplify the unwavering quality and effectiveness of system assets.

Zhang *et al*. [41] proposed that Programming Defined Network is an engineering with the center the idea of isolating the sensing plane and the control plane and a unified controller was acquainted with deal with the system hardware to acknowledge adaptable control of system movement. Innovation which utilized the SDN which gave a decent stage to application-arranged system advancements to enhance organize asset utilization, reduce working expense and rearrange organize administration. In this exploration directing turns out to be more adaptable by just changing the substance of stream tables.

Wang *et al*. [42] had suggested that the momentum inquires about on SDN which are chiefly centered around the wired system, while the product characterized remote sensor arrange is advanced in a couple of investigates, however just at the phase of advancing ideas and models. In this exploration, which had proposed a multi-jump remote system in new SDN steering plan. The usage of that convention was depicted in detail. Then construct the model with OPNET and afterward recreate it. The reenactment came about demonstrated that the proposed directing plan could give the most brief way and its system lifetime was longer than existing calculations (OLSR, AODV) when activity stack is more heavier.

Jing *et al*. [43] had implemented the requests of capacity, sending a lot of energy information to extend. Adjusting to this progressions, the software characterized organize, which has high programmability and effectiveness, is joined with the power correspondence arrange which has changed the design of the conventional system. In this examination which considers the directing system in view of Software Defined Network which proposed a hereditary subterranean insect province (GAC) steering calculation, that can consolidate the upsides of subterranean insect settlement calculation and hereditary calculation. At long last, in the wake of looking at the GAC directing an algorithm with two keen steering techniques comes about to show is enhanced and in addition the effectiveness, with the goal that the execution of energy correspondence system can be upgraded utilizing these calculations.

Jia *et al.* [44] suggested that the SDN courses of action are possible in a collection of settings including Wide Area Networks (WANs). Regardless, the far-reaching spread deferments between the controller and switches raise a couple of stresses for WANs, for instance, execution obstacles, longer directing time for streams and affectability of the controller position. Finally, a model was executed as a proof of the procedure in OPNET test framework. Exploratory results demonstrated that the approach can bring critical execution increments, for instance, the reducing of state scattering and the affectability of controller course of action in SDN-based WANs.

Azmi *et al.* [45] had explored on SDN mostly centers around wired system while programming characterized remote multi-jump arrange is advanced in a couple of examines, yet just at the phase of advancing models and ideas. In this research, proposed a novel steering convention connected SDN in remote multi-bounce organize. The execution of the convention is given in detail, and OPNET is utilized to construct the model and carry on the recreation try. Countless investigations are performed to look at the key parameters of various systems.

3.1 Gap Analysis

To get the path between two ends, various traditional routing approaches are used over the network. Still, traditionally existing schemes lack in searching optimized path.

The SDN provides a better way to routing and handles existing problems. The network topology plays an important role in network performance. There are many topologies like a dual tree, ring and spanning tree and so on. The major gap is to analyse the performance of all these topologies to check which is best in terms of performance for a given input. The existing work has not explored the various QoS (Quality of Service) parameters. Therefore, the defined approach used for selection of optimal paths on the basis of different QoS parameters i.e. bandwidth, delay, and load..

3.2 Problem Statement

The objective of the proposed approach is to evaluate the performance of two routing protocols i.e. OSPF and RIP, for SDN. This evaluation should be done theoretically and through simulation. The simulator used for the proposed approach is compared with various simulators on the basis of accuracy [9]. Hence the simulator used in the proposed approach is better than others to get the optimized path using OSPF and RIP protocols.

3.3 Objectives of the Proposed Work

The objectives of the proposed approaches are

- To design various scenario using different topologies.
- To evaluate the performance of OSPF protocol on the basis of various QoS parameters like traffic sent and network convergence.
- To compare the performance of topologies for different network size.

All the switches are configured in such a way that they can send and receive data to each other. The various scenarios are designed using various topologies like a dual tree, ring and spanning tree. The placement of switches is done as per designed topology. The main aim of work is to evaluate the performance of various topologies in term of routing using various performance parameters like cost, delay, load, and bandwidth. Various routing paths are selected on the basis of cost which is defined as:

$$Cost = custom\_db\_cost + fixed\_cost + (cost\_per\_db * data\_rate\_in\_kb) + (cost\_per\_km * distance\_in\_km)$$

Demonstrates the use of the spanning tree, dual tree and ring backbone design actions for performing topology design. It also demonstrates the use of a custom link price database file to specify the link costs between each node pair.

4.1 Spanning tree

The spanning tree design action is used to construct a minimum or near-minimum cost one way-connected network topology. The spanning tree can be parameterized to be port constrained, to use a specific root node, to be the shortest path tree or minimum cost tree etc. Generally, a spanning tree is not the best choice for an actual network topology, but it is useful to provide a baseline for the cost of a connected network.

4.2 Dual trees

The dual trees design action constructs a minimum or near-minimum two-connected topology. The algorithm works by first constructing a spanning tree and then building a second spanning tree among the leaf nodes of the first tree. The end result is a topology where at least two node-disjoint paths between each node pair. There are several parameters on the dual trees design action for controlling the root node, the type of spanning tree, link sizing, etc.

4.3 Ring backbone

The ring backbone design action is used to construct a topology based on a ring. By default, the ring backbone is constructed as a Multi-length Chordal Ring design. The Multi-length Chordal Ring topology is a 3-connected design (at least 3 node-disjoint paths between each node pair). The ring backbone design action finds a near-minimum cost topology with the specific graph properties. The design action can be parameterized to construct a simple ring (2-connected) or various chordal ring graphs. The ring backbone topologies are symmetric and as a result are good choices for networks that require high reliability and insensitivity to traffic fluctuations.

link_pricer_custom_db

This design action is used by other design actions to price candidate links between source-destination node pairs. The link's prices are based on the endpoints, the link model and (optionally) the link data rate. The link_pricer_custom_db uses a comma-separated value (CSV) text file to specify the cost values. Two example files are provided with this project to define the format and show an example of its use. These files are:

- *Link_pricer_custom_db_example.gdf*- It defines the file format.
- *greenfield_design_link_pricer_custom_db.gdf* - It used for pricing.

These files are used to change the scenario sites. The sites scenario is used as the starting point for the topology design. Each design is based on the sites scenario. The README in the sites scenario describes the steps in the design process. The _ref projects are available for comparison. Use this scenario for each topology design. The network contains 13 nodes. There is a full mesh of IP demands in the network. The network does not contain any links as shown in Figure 4.1.The goal is to design 1-connected, 2-connected and 3-connected topologies and compare the costs/performance of each link. The custom database file defines the costs for DS3 links between all node pairs in this network and also shows total cost, number of links, connectivity, and hop information by using OSPF routing algorithm.

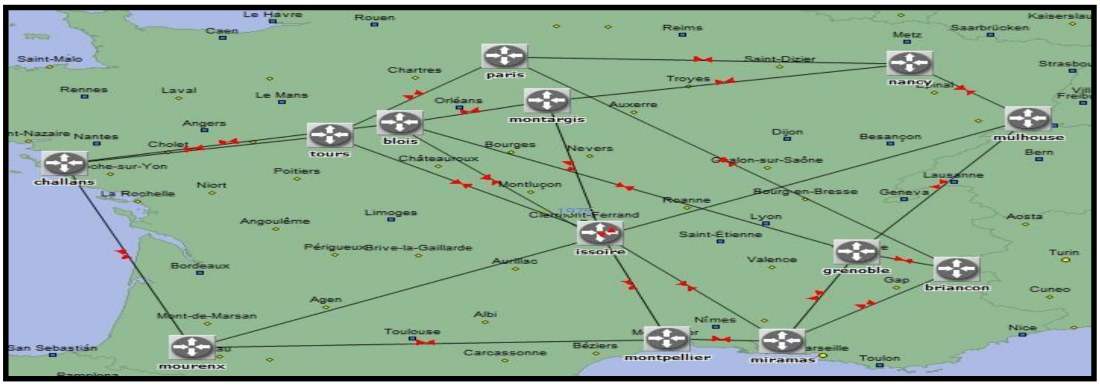

Figure 4.1 Transmissions of Data over the Network

The Figure 4.2 shows that the ring has a maximum number of links, hence find the shortest path between the nodes.

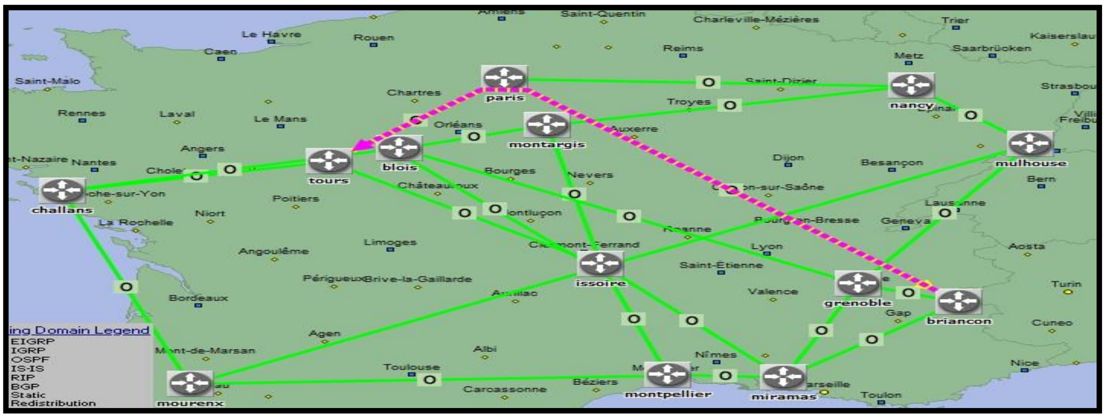

Figure 4.2 Analyze the Traffic during the Data Transmission in Ring

The Figure 4.3 shows how the packet is routing through various nodes and the proposed approach providing optimized path using the ring topology

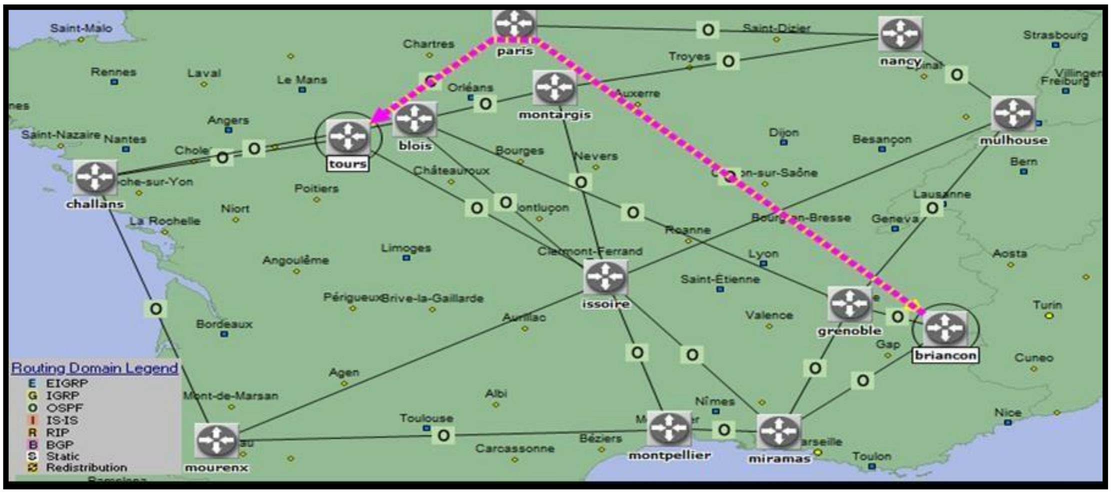

Figure 4.3 Optimization of Shortest Path between the Nodes in Ring

The Figure 4.4 shows that spanning tree has a minimum number of links and doesn't contain any loop like ring topology. Due to the usage of minimum links, it will result in congestion points.

Figure 4.4 Analyze the traffic during the data transmission in Spanning Tree

Figure 4.5 shows the data transmission over the network using dual-tree topology. The dual-tree has a maximum count of links and generally will show the result.

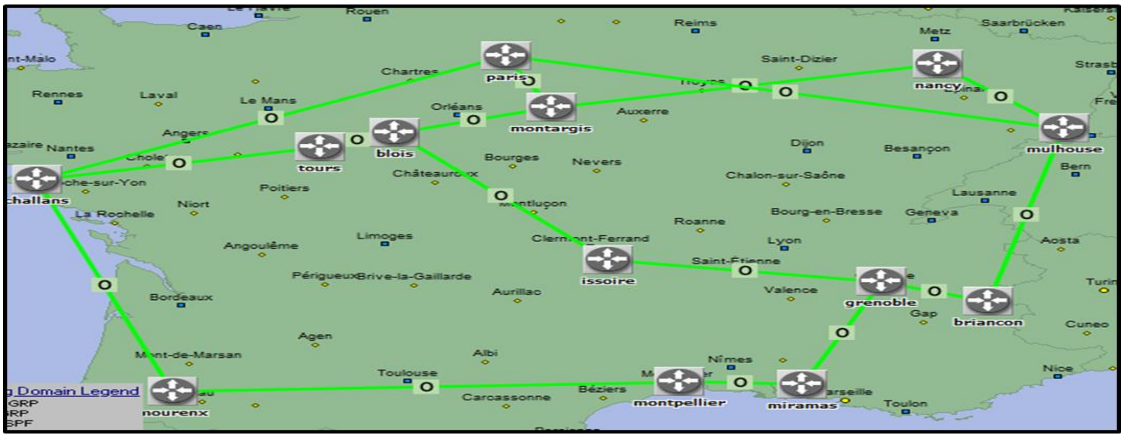

Figure 4.5 Analyze the traffic during the data transmission in Dual Tree

The performance of various topologies is evaluated on basis of various parameters i.e. cost, bandwidth and network size as shown below. Bandwidth is the amount of data that can be transmitted in a fixed amount of time. In which we evaluate the performance of different network size and check which topology give a better result.Therefore, when the nodes are increased then performances like cost is reducing in all topologies are compared to the previous scenarios.

Table 5.1 Comparison on the basis of cost, bandwidth, network size (13 nodes).

Parameter	Spanning Tree	Dual Tree	Ring Tree
Bandwidth Efficiency	2.9	2.6	5
Bandwidth Consumed	4401 x 10^5 bps	4401 x 10^5 bps	11632 x 10^5 bps
Maximum link Utilization	9.3	9.4	28.1
Value of reused link	2249.43	5705.06	4578.04
Cost of reused link	2249.43	5705.06	4578.04
Cost of topology change	4049.43	5856.06	5893.31
Average cost per link	3374.95	3448.38	34400.62
Cost of new link	180.1	315.12	6766.91

Table 5.2 Comparison on the basis of cost, bandwidth, network size (26 nodes)

Parameter	Spanning Tree	Dual Tree	Ring Tree
Bandwidth Efficiency	37.5	22.4	14.3
Bandwidth Consumed	5376×10^5 bps	4212×10^5 bps	3846×10^5 bps
Maximum link Utilization	98.4	50.0	39.3
Value of reused link	2075.43	5645.43	4662.80
Cost of reused link	2645.43	5645.43	4662.80
Cost of topology change	2645.43	228279.55	678880.65
Average cost per link	165.34	1087.45	2514.73
Cost of new link	0.0	0.0	6322.56

Table 5.1 and Table 5.2 depicts that average cost per link is lowest for spanning tree and highest for a ring topology. The cost of adding a new link in spanning and the dual tree is nil whereas quite higher in ring tree. The bandwidth factors for spanning tree are quite better in contrast with other. The cost for the reused link is higher for dual-tree, but still, the average cost per link is lower than the ring tree. The performance of spanning tree is best in terms of routing among other two topologies.

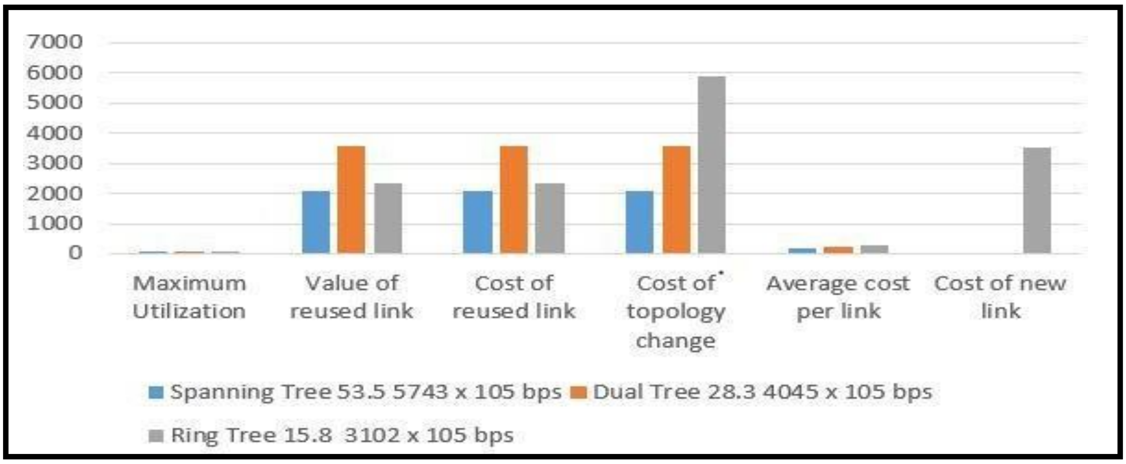

Figure 5.1 Comparison of various topologies on the basis of cost and bandwidth

The Figure 5.1 depicts the optimization of shortest routing path on the basis of various parameters of spanning tree, dual tree and ring tree. The outcomes on the basis of *cost of reused links* and *cost of topology change* parameters of spanning tree is better than other topology.

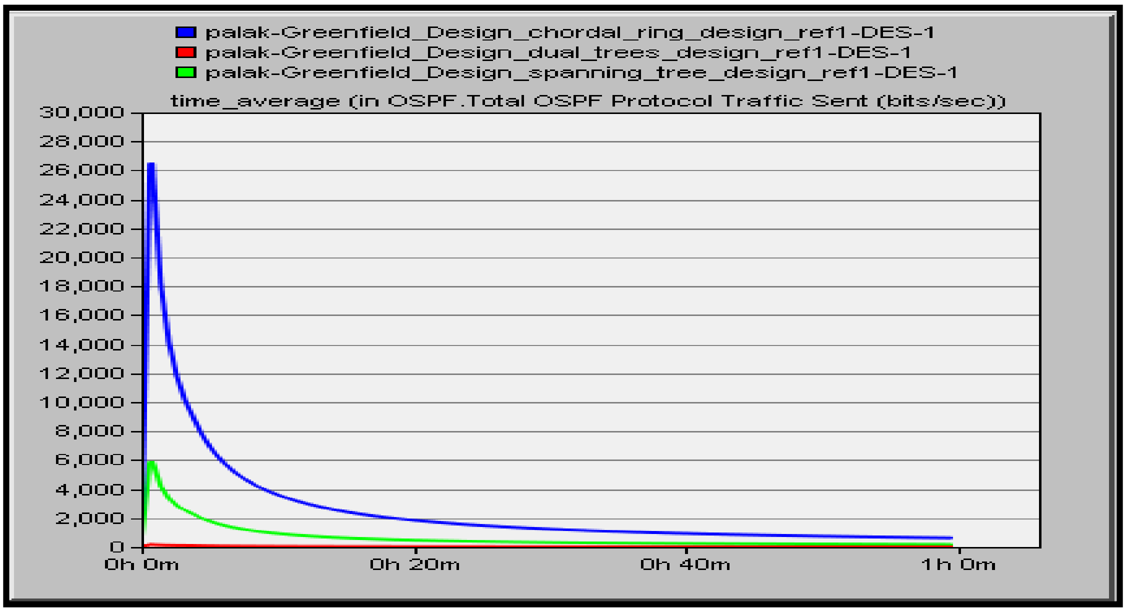

Figure 5.2 Comparison of various topologies on the basis of bandwidth in OSPF

In Figure 5.2, x-axis represents the time and y-axis represents the traffic sent and the packet sent in the form of bit per second(bps). Initially, a large amount of packets are transferred on the link and then packets are scattered and used different paths to reach the same destination. It shows the movement of packets with respect to time over the network. In case of a ring topology, traffic sent rate is more as compared to both spanning and tree topology.

6.1 Conclusion

The main focus is to evaluate the performance of various topologies in term of routing. The SDN routing is the solution to handle all the drawbacks of traditional routing protocols.

The performance of spanning tree, ring and dual tree are compared with each other and the outcomes of spanning tree are better than a ring and dual tree in terms of cost. The bandwidth is more efficiently utilized in spanning tree, but the links are heavily used in spanning tree, which results in some chances of congestion. The average cost of the dual-tree is better than the performance of spanning tree.

6.2 Summary

The SDN has the power to simplify the control of enterprise network management. Different vendors are making different claims, but the most commonly cited advantages of software-defined networking are traffic programmability, greater agility, the ability to create policy-driven network supervision, and implementing network automation. The SDN has the ability to allow networks to keep pace with the speed of change.

Software-defined Networks are intelligent networks where the routing decisions are not based on traditional routing algorithms like shortest path, Dijkstra routing algorithms and so on. In which, the scenarios of variable sizes are designed using different network topologies. The cost factor is computed every time during the data transmission from source to destination. The goal of routing is to achieve the minimum cost. OPNET simulator version 14.5 is used for this work.

6.3 Future Scope

The spanning tree uses more nodes to optimized the optimal path between two ends and sometimes leads to congestion. Hence, the work can be extended to reduce the congestion while using a large number of nodes in spanning tree.

In the proposed work, only two routing protocols i.e. OSPF and RIP, are used to evaluate the performance of the network. The count of routing protocols can be increased to evaluate the better performance and increase the number of parameters to get the deep knowledge of the optimal path over the network.

References

[1] E. Chemeritskiyand R. Smelansky, "Semi-Physical Simulation for Software Defined Network,"*In the Proceedings of International Science and Technology Conference (Modern Networking Technologies) (MoNeTeC)*, Moscow, pp. 1-6, 2014.

[2] M. Ramdhani, S. Hertiana and B. Dirgantara, "Simulation of Software Defined Network as one big switch”, *In the Proceedings of 4th International Conference on Information and Communication Technology (ICoICT)*, Bandung, pp. 1-6, 2016.

[3] Y. Lei, K. Wang and Y. Hsu, "Analysing the Performance of Software Defined Networks v/s Real Networks," *In the Proceedings of European Conference on Networks and Communications (EuCNC)*, Paris, pp. 365-369, 2015.

[4] S. Izumi, A. Edo, T. Abe and T. Suganuma, "An Adaptive Multipath Routing Scheme Based on SDN for Disaster-Resistant Storage Systems," *In the Proceedings of 10th International Conference on Broadband and Wireless Computing, Communication and Applications (BWCCA)*, Krakow, pp. 478-483, 2015.

[5] M. Fu and F. Wu, "Investigation of Multipath Routing Algorithms in Software Defined Networking," *In the proceedings of International Conference on Green Informatics (ICGI)*, Fuzhou, pp. 269-273, 2017.

[6] E. Jo, D. Pan, J. Liu and L. Butler, "A simulation and emulation study of SDN-based multipath routing for fat-tree data center networks," *In the Proceedings of the Winter Simulation Conference*, Savanah, GA, 2014, pp. 3072-3083, 2014.

[7] S. Pasca, S. Kodali, and K. Kataoka, "AMPS: Application-aware multipath flow routing using machine learning in SDN," *In the Proceedings of Twenty-third National Conference on Communications (NCC)*, Chennai, pp. 1-6, 2017.

[8] R. Oliveira, C. Schweitzer, A. Shinoda and L. Prete, "Using Opnet for emulation and prototyping Software-Defined Networks," *In the Proceedings of IEEE Colombian Conference on Communications and Computing (COLCOM)*, Bogota, pp. 1-6, 2014.

[9] R. Fontes, S. Afzal, S. Brito, M. Santos and C. Rothenberg, "Opnet-WiFi: Emulating software-defined wireless networks," *In the Proceedings of 11th International*

Conference on Network and Service Management (CNSM), Barcelona, 2015, pp. 384-389, 2015.

[10]F. Keti and S. Askar, "Emulation of Software Defined Networks Using Opnet in Different Simulation Environments," *In the Proceedings of 6th International Conference on Intelligent Systems, Modelling and Simulation*, Kuala Lumpur, pp. 205-210,2015.

[11]S.Rout, B.Sahoo, and A.Jena,” Load Balancing in SDN using effective traffic engineering method”, *In the Proceedings of International Conference on Signal Processing and Communication(ICSPC)*, pp. 484-489, 2016.

[12] I. Godanj, K. Nenadić, and K. Romić, "Simple example of Software Defined Network,"*In the Proceedings of International Conference on Smart Systems and Technologies (SST), Osijek*, pp. 231-238, 2016.

[13]F. Keti and S. Askar, "Emulation of Software Defined Networks Using Opnet in Different Simulation Environments," *In the Proceedings of 6th International Conference on Intelligent Systems, Modelling and Simulation*, Kuala Lumpur, pp. 205-210, 2015.

[14]M. Qilin and S. Weikang, "A Load Balancing Method Based on SDN," *In the Proceedings of Seventh International Conference on Measuring Technology and Mechatronics Automation*, Nanchang, pp. 18-21,2015.

[15]S. Wang, "Comparison of SDN OpenFlow network simulator and emulators: EstiNet vs. Opnet," *In the Proceedings of IEEE Symposium on Computers and Communications (ISCC)*, Funchal, pp. 1-6, 2014.

[16]F. Hu, Q. Hao, and K. Bao, "A Survey on Software-Defined Network and OpenFlow: From Concept to Implementation," *In the Proceedings of IEEE Communications Surveys & Tutorials*, vol. 16, no. 4, pp. 2181-2206,2016.

[17]H. Nam, D. Calin and H. Schulzrinne, "Towards dynamic MPTCP Path control using SDN," *In the Proceedings of IEEE NetSoft Conference and Workshops (NetSoft)*, Seoul, pp. 286-294, 2016.

[18]M. Koernera, O. Kaoa,” Evaluating SDN Based Rack-to-Rack Multi-path Switching for Data-Center Networks”. In the Proceedings of International Conference on Future Networks and Communications, pp 488-510, 2016.

[19]L. Wang, A. Anta, F. Zhang, J. Wu, and Z. Liu, "Multi-resource energy-efficient routing in cloud data centers with network-as-a-service," *In the Proceedings of IEEE Symposium on Computers and Communication (ISCC)*, Larnaca, pp. 694-699, 2015.

[20]L. Guillen, S. Izumi, T. Abe, T. Suganuma and H. Muraoka, "SDN implementation of multipath discovery to improve network performance in distributed storage systems," *In the Proceedings of 13th International Conference on Network and Service Management (CNSM)*, Tokyo, pp. 1-4, 2017.

[21]S.Toufga,"An SDN hybrid architecture for vehicular networks" *In the Proceedings of IEEE Communications Surveys & Tutorials*, vol. 16, no. 4, pp.181-206,2016.

[22]A. Azzouni, R. Boutaba and G. Pujolle, "NeuRoute: Predictive dynamic routing for software-defined networks," *In the Proceedings of 13th International Conference on Network and Service Management (CNSM)*, Tokyo, pp. 1-6, 2017.

[23]J. Chavula, M. Densmore, and H. Suleman, "Using SDN and reinforcement learning for traffic engineering in UbuntuNet Alliance," *In the Proceedings of International Conference on Advances in Computing and Communication Engineering (ICACCE)*, Durban, pp. 349-355, 2016.

[24]A.Rezende, S. K. Singh, T. Das and A. Jukan, "A Survey on Internet Multipath Routing and Provisioning," *In the Proceedings of IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2157-2175, Fourth quarter 2015.

[25]J.Guillaume, D.Banfi,"Endpoint-transparent Multipath Transport with Software Defined Networks", *In the Proceedings of 13th International Conference on Network and Service Management (CNSM)*, Tokyo, pp. 1-6, 2017.

[26]L. Rocha, "Framework for traffic engineering of SDN data paths", *In the Proceedings of 13th International Conference on Network and Service Management (CNSM)*, Tokyo, pp. 1-6, 2017.

[27]M.Gupta and P.Barford,"Fast, accurate simulation for SDN prototype", *In the Proceedings of 13th International Conference on Network and Service Management (CNSM)*, Tokyo, pp. 1-4, 2017.

[28]F. Paolucci, F. Cugini, A. Giorgetti, N. Sambo and P. Castoldi, "A Survey on the Path Computation Element (PCE) Architecture," *In the Proceedings of IEEE*

Communications Surveys & Tutorials, vol. 15, no. 4, pp. 1819-1841, Fourth Quarter 2013.

[29]E. Gupta, ”Simplifying software-defined network optimization using SQL”, *In the Proceedings of Seventh International Conference on Measuring Technology and Mechatronics Automation*, Nanchang, pp. 18-21,2015.

[30]R. Dilmaghani, ”Evaluation of Software-Defined Networking for Mission Operations”, *In the Proceedings of 13th International Conference on Network and Service Management (CNSM)*, Tokyo, pp. 1-6, 2017.

[31]G. Aceto, V. Persico, A. Pescapé and G. Ventre, "Sometimes: Software-defined network-based available Bandwidth Measurement In MONROE," *In the Proceedings of Network Traffic Measurement and Analysis Conference (TMA)*, Dublin, pp. 1-6,2017.

[32]M. Sandri, A. Silva, L. A. Rocha and F. L. Verdi, "On the Benefits of Using Multipath TCP and OpenFlow in Shared Bottlenecks,” *In the Proceeding of IEEE 29th International Conference on Advanced Information Networking and Applications*, Gwangiu, pp. 9-16, 2015.

[33]R. Mijumbi, J. Serrat, J. L. Gorricho, N. Bouten, F. De Turck and R. Boutaba, "Network Function Virtualization: State-of-the-Art and Research Challenges," *In the Proceedings of IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 236-262, First quarter 2016.

[34]J.Jimenez and O. Romero, ”Study of Multimedia Delivery of Software Defined Network”, *In the Proceedings of IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2157-2175, 2015.

[35]J. Pang, G. Xu and X. Fu, "SDN-Based Data Center Networking With Collaboration of Multipath TCP and Segment Routing," *In the Proceedings of IEEE Access*, vol. 5, pp. 9764-9773, 2017.

[36]P.Patil and A.Hakiri,” Scalable and Adaptive Software Defined Network Management for Cloud-hosted Group Communication Applications”, *In the Proceedings of the 10th International Conference on Utility and Cloud Computing*, pp 20-29,2016

[37]Y. Zhou *et al*., "A Load Balancing Strategy of SDN Controller Based on Distributed Decision," *In the Proceedings of IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications*, Beijing, pp. 851-856,2014.

[38]W. Liao, S. Kuai and C. Lu, "Dynamic Load-Balancing Mechanism for Software-Defined Networking," *In the Proceedings of International Conference on Networking and Network Applications (NaNA),* Hakodate, pp. 336-341, 2016.

[39]C.Jiann and Y. Hong,"Load Balancing Multiple Controllers Mechanism for Software Defined Networks"*In the Proceedings of International Conference on Wireless Personal Communications*, pp. 336-341, 2016.

[40]A. Neghabi, N. Navimipour, M. Hosseinzadeh and A. Rezaee, "Load Balancing Mechanisms in the Software Defined Networks: A Systematic and Comprehensive Review of the Literature,*" In the proceedings of IEEE Access*, vol. 6, pp. 14159-14178, 2018.

[41] Zhang, Jie& Zeng, Deze&Gu, Lin & Yao, Hong & Fan and Yuanyuan," On Rule Placement for Multi-path Routing in Software-Defined Networks", *In the Proceedings of IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2157-2175, 2015.

[42]J.Wang, P.Zhai, Y. Zhang,"Software Defined Network Routing in Wireless Sensor Network", *In the Proceedings of International Conference of Cloud Computing, pp*. 3-11,2016.

[43]S. Jing, W. Muqing, B. Yong, and Z. Min, "An improved GAC routing algorithm based on SDN," *In the Proceedings of IEEE International Conference on Computer and Communications (ICCC),* Chengdu, pp. 173-176,2017.

[44]X.Jia and Z. Zhang," Using path label routing in wide area software-defined networks with OpenFlow"*In the Proceedings of IFIP International Conference on Network and Parallel Computing (NPC),* pp. 281-290,2016.

[45]A. Azmi, A. Muamer and N. Mohammed," Software Defined Network in Wireless multihop network" *In the Proceedings of IEEE Access,* vol. 5, pp. 9764-9773, 2017.

Appendix A
List of Publications

[1] Paluck and S. Jain, "*Performance Evaluation of multi-hop routing in Software Defined Network*", In the Proceedings of IEEE Second International Conference on Computing Communication Control and automation (ICCUBEA), held at Pune, Maharashtra.

Appendix B

Plagiarism Report

Performance Evaluation of Multi Hop Routing in Software Defined Network

ORIGINALITY REPORT

12%	7%	6%	6%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	gdeepak.com Internet Source	2%
2	Naga Katta, Haoyu Zhang, Michael Freedman, Jennifer Rexford. "Ravana", Proceedings of the 1st ACM SIGCOMM Symposium on Software Defined Networking Research - SOSR '15, 2015 Publication	1%
3	pdfs.semanticscholar.org Internet Source	<1%
4	Submitted to Centennial College Student Paper	<1%
5	Josiah Chavula, Melissa Densmore, Hussein Suleman. "Using SDN and reinforcement learning for traffic engineering in UbuntuNet Alliance", 2016 International Conference on Advances in Computing and Communication Engineering (ICACCE), 2016 Publication	<1%