Republic Of Iraq
Ministry of Higher Education & Scientific Research
Iraqi Commission for Computers & Informatics
Informatics Institute for Postgraduate

Software Defined Networking Architecture and Design based on Energy Conserving Model

Submitted to the Informatics Institute for Postgraduate Studies at the Iraqi Commission for Computers and Informatics as a Partial fulfillment of the Requirements for the degree Master of Science in Computer Science

By
Bassam Noori Shaker

Supervised By
Asst. Prof. Dr. Mohammad Najm Abdullah

2017 1438

بسم الله الرحمن الرحيم

يَرْفَعِ اللَّهُ الَّذِينَ آمَنُوا مِنكُمْ وَالَّذِينَ أُوتُوا الْعِلْمَ

دَرَجَاتٍ وَاللَّهُ بِمَا تَعْمَلُونَ خَبِيرٌ

صدق الله العظيم

سورة المجادلة الاية (11)

Dedication

To my beloved family . . . .

My Mother ,My Father

My Brothers and sisters

To those close to myself "My wife and children"

To my dear country "Iraq"

Bassam

Acknowledgement

First and foremost, praise be to ***Allah*** *for His guidance that has enabled me to pass all the difficulties I have got to complete my thesis.*

I am so grateful to my supervisor ***Asst. Prof. Dr. Mohammad Najm Abdullah*** *for his advice, wisdom and his guidance for improving my work.*

I would like to thank all my teachers and the staff of the ***Iraqi Commission for Computers and Informatics.***

Last, but not least I would like to thank ***my family****, without them, I could not have reached this in my life. They have helped me to pass all my difficulties easily.*

Bassam Noori

Abstract

As consumed energy of information and communication technology (ICT) has been increasing rapidly in the last years Relative to the developments in this field, energy- efficiency problem becomes an important topic for the most network researchers.

In the conventional network, it is hard to define a cooperative way for reducing energy consumption for all network devices (switches and routers) because the control functions distributed over them, on the other side, a new network architecture, software defined network(SDN) arise with new features such as a centralized controller.

This thesis aims to produce a model for reducing the energy consumption by using dynamic modification method. Re-routing the traffic from the default path such as the shortest path to another one that serves our objective function which is minimized the number of active links and in the same time satisfy the constraint such as link capacity and the amount of traffic demand represents the essence of this work.

The proposed model has been tested using different types of network topologies such as Abilene backbone network and data center topology (fat tree, two tier, three tier) and comparison it with Dijkstra algorithm.

Through the results obtained from the simulation operation, we find the model achieves energy saving nearly 64% and 19% relative to full active network and Dijkstra algorithm in case of fat tree topology, 64% and 7% for two-tier topology, and for the third case three tier datacenter topology its achieves about 54% of energy saving.

Contents

Chapter 1 “Introduction”

Chapter 2 “SDN Theoretical Concepts”

List of Figures

List of Tables

List of Abbreviations

API	Application Program Interface
BIP	Binary Integer Programming
CLI	Command Line Interface
FNSS	Fast Network Simulation Setup
ICMP	Internet Control Message Protocol
ICT	Information and Communication Technology
ILP	Integer Linear Programming
Iperf	Internet Performance
LLDP	Link Layer Discovery Protocol
LPI	Low Power Idle
NOS	Network Operating System
NP	Network Performance
ONF	Open Networking Foundation
QoS	Quality of Services
RTT	Round Trip Time
SDN	Software Defined Network
SWN	Software Wireless Network
TCAM	Ternary Content Access Memory
TCP	Transmission Control Protocol
UDP	User Datagram Protocol

List of Symbols

$P_{chassis}$	amount of power consumed by the chassis of active switch
P_{switch}	total power consuming by the switch.
W	amount of energy consumed by the port
$Port_i$	active port i
N	total number of active ports.
$P_{network}$	Total power consumption by the network
n	total number of network links
l	link in the network
TOPO	Network topology
S	Set of nodes
L	Set of the links
C_l	capacity of the link
C_f	traffic volume

Chapter 1

Introduction

Chapter One

Introduction

1.1 Background

In recent years there has been interesting on the subject of energy efficiency in the electronic devices because reducing energy consumption be companied with him many benefits such as saving money, reducing footprint, minimizing risk [1].

The rapid development of the content, services, and applications that offered by the Information and Communication Technology (ICT) in general and especially in networks domain, is basically the reason of the increased need for larger storage volume and high-speed data transfer, which growing significantly the energy consuming [2].

current network with its infrastructure is unable to keep up with vast developments of ICT since new technologies require more flexibility, scalability network resources In order to face environmental and economic problems. So its desire to found new network technology that contains new different aspects of solving contemporary problems such as energy consumption.

The new network paradigm is represented in Software Defined Network (SDN) where the principle of these networks is based on the separation of the control functionality in the network from the forwarding packets functionality. This will make the network more flexible in control and management operations[3].

Energy efficiency linked to the provision and operation of the necessary resources to meet the user's needs from the required services at the lowest energy consumption. In another word the amount of energy that required for produce a functional unit. Networking devices designed to operate at full capacity for both in peak time and in non-peak time to support services to the users. There are set of principles for energy management in telecommunication systems starting from planning phase that requires for setup the network by choosing the appropriate location for network resources, and principle of new investment energy-aware technologies in network components such as silicon and modify network operations according to the workload .

For getting energy – aware network operations we need more flexible network in both sides control and management than current network because new network applications and services require resources with more dynamic anticipation and a higher degree of flexibility. In the context of founding new network architecture with more flexibility for energy-efficient purpose Software Defined Network (SDN) represent the solution [4].

SDN Allowing users to manage network equipment using software by taking the intelligent out of the switches and routers and locating it in a central point called the controller. New SDN architecture facilitates the adaption of the network forwarding devices for energy-efficient.

SDN architecture consist of three layers and two interfaces:

- Data plane layer: construct from forwarding devices (switches, routers) as in the traditional network but without control functionality, data plane (infrastructure layer) responsible for forwarding incoming traffic to the next node based on matching process between the incoming order from the upper layer (controller) and stored information in the flow table that implemented inside the forwarding devices.
- Control plane layer: represent the entity that responsible of controlling forwarding traffic decisions inside the data plane, the controller embodied as a software that implemented in the external

central logic device for that reason it could be developed without needed for new hardware[5].

- Application layer: is the top layer of SDN architecture and its construct from a set of application such as (network traffic management application and loading balancing application) that responsible for managing the network through control layer.
- Southbound interface: represents the connector between the controller and the forwarding devices and use for defining the instruction set within the flow table.
- Northbound interface: represents the connector between two layers application and the control layer and use for providing the first layer with an abstract view of the forwarding layer .

1.2 Problem Definition

This work deals with the problem of energy consumption in computer networks, which is reflected in reducing consumed energy by devices involved in these networks without affecting the quality of service provided by this network.

1.3 Literature Survey

Below is a summary of researchers works for solving energy conservation problem in the SDN-based network.

Yadi Ma and Suman Banerjee, in 2012 [6] take advantage of the memory search process in the Ternary Content Access Memory (TCAM) to achieve their goal of reducing energy consumption, where the TCAM venders design the memory as contiguous same size of blocks. These blocks used for saving the information so for any incoming packet matching operation will arise and all blocks will activate for this operation. The researchers produce a smart pre-classifier technique that lead for energy saving by reducing the active blocks(more active bloke more energy consumption) so TCAM blocks will use pre-classifier for consulting at the first and then matching a lower number of blocks this will help in reducing the number of active block that needed and consequently reducing the energy consumption.

Kalapriya Kannan and Subhasis Banerjee, in 2014 [7] investment compacting the TCAM,which is an electronic circuit that is used for saving switch information such as pre-defined rule in case of openflow switch, for minimizing the size of flow entries in open flow switches this minimizing will help in reduction of energy consumption due to less search in TCAM blocks.

Frederic Giroire et al, in 2014 [8] they used southbound interface (OpenFlow) to achieve their goal in reducing the energy consuming in SDN, by minimizing the rules number in the flow table that holds in the open flow switches by TCAM memory which is energy consumer and expansive, the flow table managed by OpenFlow protocol.

The researcher uses Integer Linear Programming (ILP) for optimizing rule placement for a small network and using links capacity and rule space as a constraint. For large network they use a heuristic algorithm that begin with compute the available routing under the constraint of link capacity and rule space when the routing table is full and there is no more space for additional rules they minimize it by choosing the port with high traffic to set as default port and this will help in reducing the number of implemented rules in the switch. Next, the algorithm selects the most less loaded links and switch-off them and modulate the traffic that passes through these links to another link.

Radu Carpa et al, in 2014 [9] deals with the problem of reducing energy consuming in the network by putting the ports and nodes that connected to this ports on both ends of the links in the sleeping mode in case of unused this link and active again whenever the links needed. The researchers divide the problem into three stages in the first one choosing the candidate links for changing the status of them and in the second stage selecting the new path by avoiding the using one, and third stage is push the rules of selected path from the controller to the nodes. They testing them work on two different backbone network namely Germany 50 network and Geant network 22 with dynamic traffic matrix.

Tran Manh Nam et al, in 2015 [10] they presented a module to solve the energy consumption problem based on principle of dynamic adaption for energy saving, where the module profiling the network devices and ports of the switches under different link rate according to the energy consumption, then they embedding this profile in the process of routing and topology optimization for achieving best results.

The energy consuming in the NW is result from the total energy consumption of the active forwarding devices, number and capacity for each port per device, the amount of traffic which pass though the devices and because the module support different venders forwarding devices so it will depending on the energy profile that support the module. The model divided in to two sub module first one reside in the controller of network and act as monitor for the statistics of the switches such as switch and link stat and amount of traffic, this information is exchanges between network devices and controller via open flow protocol. The second module responsible of generating the traffic.

Deze Zeng et al, in 2016 [11] introduces a proposed heuristic algorithm for energy efficiency in datacenter network based SDN architecture by minimizing the number of active switches depending on determining the paths for each flow demand in the traffic matrix by satisfying two constraints first one the link capacity and the other TCAM size.

the researchers widely used multiple path routing in their algorithm, in the case of traffic demand is less than link capacity so it will divide into the sub-flows and routed them through a number of paths from source to destination points to accomplish their goal. they examined their algorithm on the fat tree datacenter with two types 8 and 4 pods and compare their result with shortest path algorithm.

Fahimeh Alizadeh Moghaddam and Paola Grosso, in 2016 [12] developed a model that is based on Integer Linear Programming(ILP) in his work by defining the problem of energy consumption as linear function, the network constraint(bandwidth,) is predefined to meet the variable and the value of this function, this method used for scheduling the incoming traffic flow to the appropriate paths which meet the objectives of method.

The objectives function for this(ILP) method are reducing energy consuming in the NW by putting in the sleeping mode unused switches, and prefer links that already used on the sleeping one.

Mahshid Rahnamay et al, in 2016 [13] modulate an optimization framework that help in solving energy consumption problem by determine best set of edges to be switch-off for energy aware purpose (cost reduction) while settlement the amount of traffic that will be forwarded prior to the available resources (communication and energy) of the network .

The researchers developed a heuristic algorithm to solve the optimization problem of that framework where they testing this algorithm on Abilene backbone network by generating ten shortest paths between each two nodes in the network and take the distance between these nodes(which has effect on the delay time for delivery the traffic for example) and energy cost as constraints. The algorithm is depends on the assumption that the network in its start state is full active and links capacities of the network can satisfy all traffic demand.

Ying Hu et al, in 2016 [14] address the problem of energy consumption from viewpoint of the SDN controller placement, by taking into consideration the delay time of the links propagation between the controller and the switches that belong to him and the number of switches that connected to the controller which represented the load of the controller as constraints.

They used Binary Integer Programming (BIP) model to accomplish their goal (reducing energy consumption) and selecting most suitable controller placement that maximize energy saving for small network where each node could be a controller, after selecting the most a proper one the reset of nodes (switches) connect to this controller and each switch will connect to only one controller, for saving the energy the idle link can be modify to sleep mode by sleeping the ports corresponding to him. table(1.1) is categorized the related works.

Table(1. 1): Literature Survey Categories

category	approach
compacting TCAM	Pre-classifier [5]
	compacting[6]
rule placement	optimizing rule space [7]
	Controller Placement[13]
traffic aware energy	Segment routing[8]
	power profile[9]
	Joint Optimization[10]
	Linear programming[11]
	resource-aware[12]

1.3.1 Summary of the Literature Survey

From the above researcher's works the following remarks are concluded :

1. The researcher's solutions for the energy consumption problem applied other in the SDN controller or in the forwarding devices.
2. The researcher solution to the problem could be categorized into three types namely (compacting TCAM, rule placement, traffic aware energy).
3. Most of the solutions based on putting the unused nodes and links in the network in the sleeping mode or turning off them.

1.4 Aim of the Work

This work is intended to:

1- Analysis previous works of researchers within the same field of this work (energy aware in SDN network).
2- Give a brief study of the SDN networks its benefits, challenges and the most difference with the traditional network.
3- Design energy saving model for SDN networks
4- Constructing a model for reducing energy consumption in SDN network and testing this model by using mininet simulator.
5- Checking network performance by using set of tools such as ping and iperf.

1.5 Thesis Contribution

In this thesis, a new proposed model is introduced for reducing the energy consumption in the SDN networks by taking advantage of the remaining capacity of the links after forwarding the traffic .

1.6 Thesis Outline

This thesis is arranged in five chapters, the rest chapters Included the following:

Chapter two, introduces an overview to the new SDN technology and explains its history, architecture, benefits and challenges that face this technology and finally the main concepts of energy saving over SDN networks.

Chapter three reviews the proposed model that is used for reducing the energy consumption in SDN network by using dynamic modification method .

Chapter four, evaluate the proposed model by perform experiments with different types of network topology such as Abilene, fat tree datacenter topology, two-tier datacenter topology.

Chapter 5, contains the conclusions that obtained from this work followed by suggestions to the future works.

Chapter 2

SDN Theoretical Concepts

Chapter Two

SDN Theoretical Concepts

2.1 Introduction

This chapter gives an overview of the SDN technology and main concepts of energy saving. Starting with a brief history of the beginnings of SDN computer networks, next clarified the main idea of this new technology which enable computer networks be programmable , Next takes a walk through on SDN architecture as bottom-up layered approaches stating how this architecture construct from three layers (planes) namely data planes, control plane and applications plane and two connectors between these plane southbound interface and northbound interface, Also explain the main challenges and benefits on this network and the main network performance parameters and essential metrics to measures these parameters .

2.2 History of SDN

The general term programmable in the network branch mean simplifying the management and configuration of the network and used to cover many ideas suggest over time.

Ethane is a project proposed by Martin Casado, who is a researcher in Stanford university under the name clean stat project. This project represents the beginnings of developing SDN idea, Ethane project aims to build a powerful new network architecture for enterprise with advance characteristic, more softly management, tough security and enable the administrators to push and tailor security policies for flow easily and dynamically by centralized controller. The researchers of clean stat project debate that network management and security have integrated relation because both of them based on right manage to the policy which responsible of control connectivity. Ethane realized this goal by pairing simple switches based on flow and centralized controller in charge of managing the routing traffic of flows in these switches through secure connection[15] [16].

Nick McKeown in 2008 proposed the main concepts of Openflow to facilitate the programmability of the network based on Ethane clean stat project idea and implemented as a software and hardware [17].

2.3 SDN Definition

Software Defined Network (SDN) is a new network architecture that most appropriate for modern dynamic and high bandwidth applications because of it more manageable , dynamic, and cost-effective from traditional architecture [18].

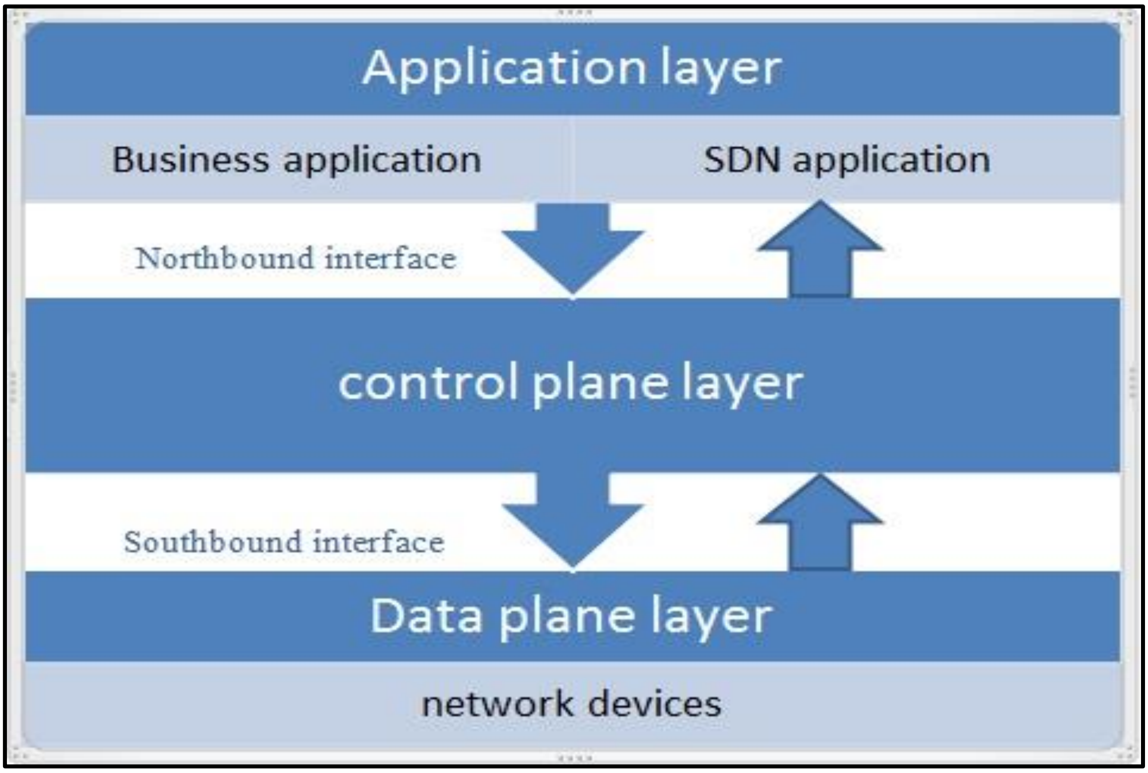
SDN architecture was built on idea of decoupling the network forwarding and controlling functionality and this will facilitate the network control to be programming directly and forwarding devices to be abstracted for services and network applications[19].

- Easy programmability: control operation of the network can be directly programmed because of stripping the controlling function from the forwarding device.
- Agile: separation control from the underlying infrastructure facilitate to the network administrator dynamically modify network traffic for special management purpose.
- Centralized management: brain of the network (controller) preserve a global view of the whole network because is logically centralized which seems as a single switch to the application and policy engines

- Programmatically configured: software defined network allow to network managers by using dynamic SDN programs modifying, managing, securing and enhancing network resources quickly.

2.4 SDN Architecture

Based on SDN definition that early mentioned, can clarify the SDN components as set of different planes or layers (data , control, application) which represent the SDN architecture as shown in the figure (2.1) each one of this planes has its own functionality, and the communication between them can be done through open standard interfaces . Next, we will depict these layers based on bottom-up approach [20].



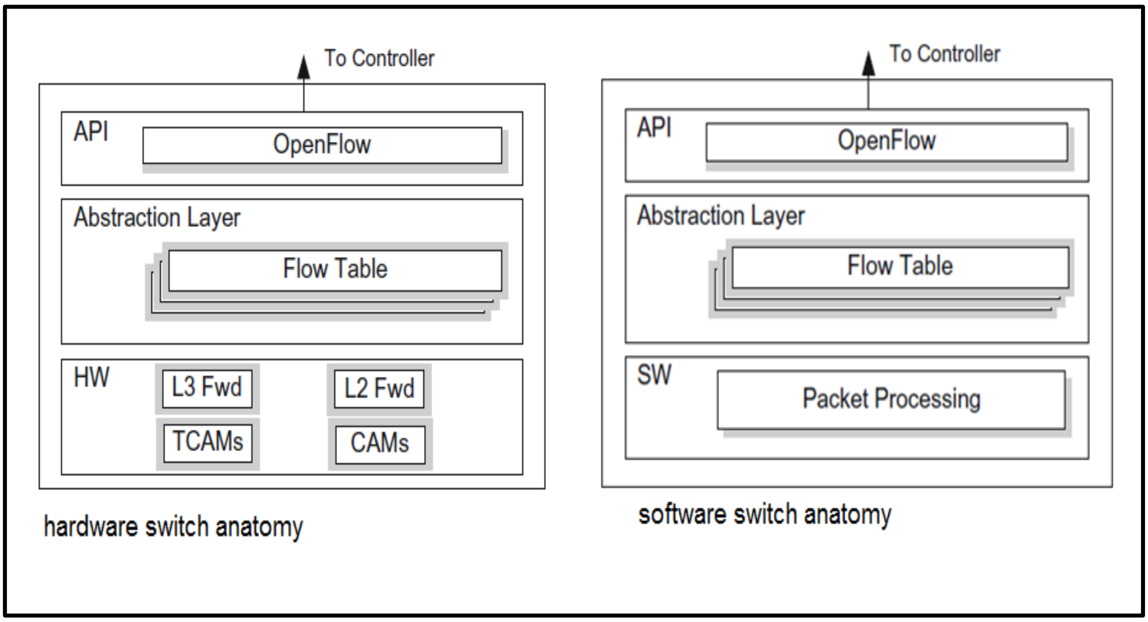
Figure(2. 1): SDN Architecture

2.4.1 Data Plane

This plane represents the forwarding devices (switches and routers) plus a set of instructions that could be defined by Application Program Interface (API) . SDN network devices same as traditional network devices with main difference in their functionality which are forwarding the packets based on a prior decision taken by the higher plane. this means that the control functionality removed to the external entity and make it logically centralized.

The data plane interact with the upper plane (control plane) through a standard interface (OpenFlow) in another word the network brain (control) and applications are built (conceptually) above open and standard interface. This interface allows to the controller to dynamically programming different forwarding devices.

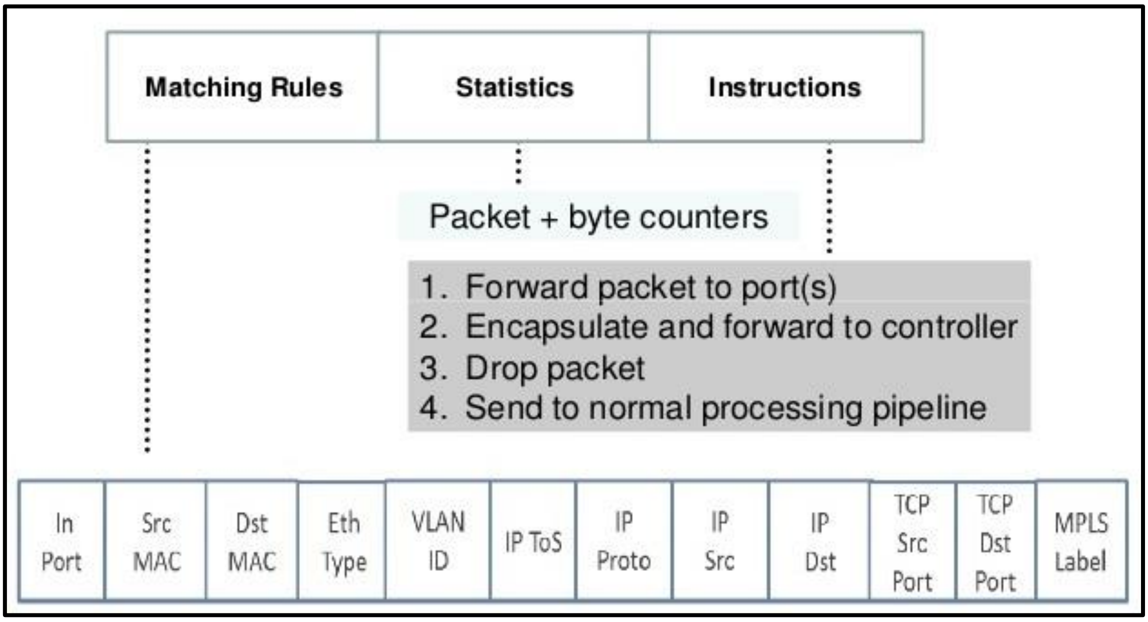
Forwarding device of SDN data plane consist of API that is used for interact with the controller , abstraction layer and traffic (packet) processing function that will be software in case of virtual switch and embodied as a hardware for traffic processing logic in case of physical switch as show in the figure (2.2)[21].



Figure(2. 2):Hardware & Software Switch

The abstraction layer takes a shape as one or more than one flow tables its main function allowing the device to decide the appropriate action for the coming packet, depending on its contents. This action may be forwarding the packet to a specific switch port or flooding it to all ports or dropping the packet [22].

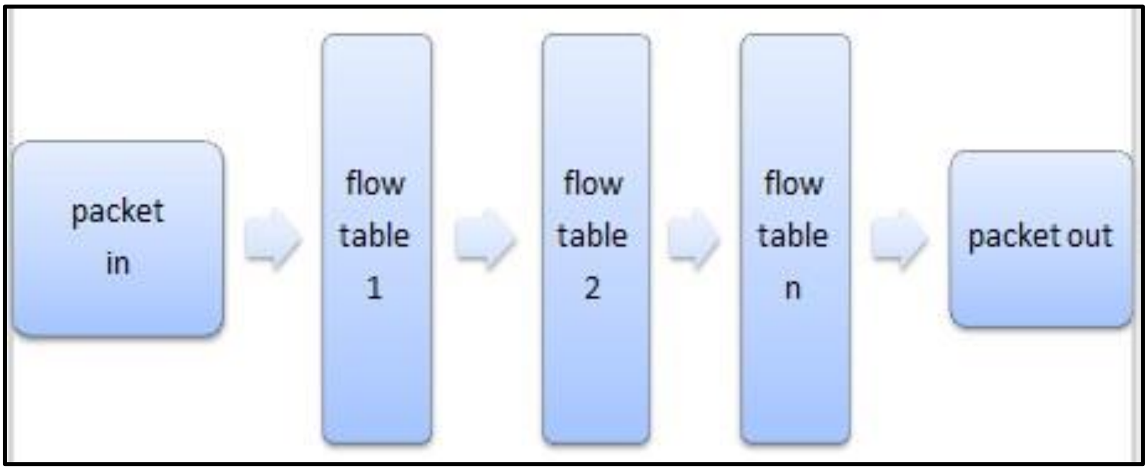
Flow table is a data structure that resides in the high-speed data plane in open flow switch. It contains the determined the forwarding behavior, packet handling behavior of that open flow switch. Open flow table has one or more of flow entries each flow entry has set of component, figure (2.3) describe the flow table which component consist of three entries (match fields , action, and priority) in addition to these fields it contains a counter and timer.



Figure(2. 3): Flow Table Entries

In some cases, the packet may matchless any entries in the flow table so table-miss flow entry founded to solve this cases by the set of actions such as sending the traffic to the controller, re-forwarding the packet to the next table or drop the packet.

In a case of the Open Flow switch contain more than one flow table, tables are sorted as pipeline and the packet process operation among these tables called pipeline processing as shown in the figure (2.4) [23].



Figure(2. 4):Pipeline processing

2.4.2 Control Plane

SDN controller is the often name of the control plane and also called Network Operating System (NOS), with SDN architecture the management of the network exchange from distributed fashion in the traditional architecture to the centralized one because of the controller connected to all forwarding devices in the lower plane[2]. The main functions of the controller are :

1- Provide an abstraction view of underlying infrastructure to the applications plane so that these applications will have the ability to communicate with the SDN devices (switches, routers).
2- Carry out policy decisions (load balancing, forward, routing) of the administrators.
3- Controlling of all data plane devices that construct network infrastructure [24].

The logical position of the controller help in solving many problems that deep-rooted in the distributed fashion such as faster responding to the node or link failure because failure node directs connected to the controller.

The controller has an overview of all network and this makes avoiding loop process much easier.

There are different types of SDN controller based on the type of programming language that used for implantation , such as pox controller which is build base on python language or flooding light which is implemented in java language and even could use c-language in case of NOX controller. All of them are open source controller and there are also commercial one such as HP and NEC [25].

2.4.3 SDN Application Layer

SDN technology provides the users with the programmable platform that facilitate for them building SDN applications for solving urgent network problems such as quality of services and routing management. Network applications have a top plane in SDN architecture, communicate with the controller via API called northbound interface. The main duty of these applications is managing the traffic inside the network devices by modifying the flow entries in them by using southbound interface [25].

2.5 SDN Challenges

Although SDN represents the most promises solution for traditional network problems, but the new technology suffering from some challenges that impact on its performance, below some of these challenges.

2.5.1 Reliability

Network operating system is considerate the prevention tool from manual error but it splitting nature of the controller make it suffering from the single point of failure problem.

The traditional network can handle the failure in network devices(nodes) easily by re- routing the traffic to another node so can ensure the connection continuity, for that reason we can say the traditional network has reliability. However, in SDN network lack of a standby controller, there is a single controller manage the whole network in case of fail happen in the controller all network will down. Many proposed solution founded for solving the reliability problem all based on exploiting the controlling functionality such as supporting multiple paths routing in case of link failure happen[26].

2.5.2 Scalability

Besides the advantage of separating the controller idea, some drawbacks arise such as scalability problem. SDN controller could be a bottleneck in case of scaling up the network by increasing the numbers of the switches, bandwidth and hosts, so the controller may not be able to respond to all requests sufficiently.

Also, another scaling problem concerning in the networks based on SDN architecture is the flow table that implemented inside the open flow switches which needs to configure instantaneously at each time of setup a new flow by the controller and number of flows. this controller need to accomplish may exceed millions per second in a large network these operations will cause additional overhead.

2.5.3 Security

Security issues in SDN differ from traditional one in many aspects prior to the implementation the architecture of each one of them . With SDN the domination of the controller on all infrastructure devices of the network and direct accessibility of application to the control plane by the northbound interface represents an excellent choose for attackers.

Three categories of security challenges can be proposed prior to networks based SDN-related to the position of attack(switch, control, channel)[27] .

- **Switch security challenge:** In this category, the challenges pointed to vulnerabilities in the network infrastructure devices (switches). Because switches contain flow table and important information about traffic routing , management of the network so it will be a good target for attackers. Also attackers can aimed the network forwarding devices them self and trying to get an access for the network (physically , virtually) or connect to the network through legal host by compromise it in order to destabilize network devices.
- **Controller security challenge:** In this category, the challenges pointed to vulnerabilities in control plane in SDN networks. Although the centralized controller an efficient and inelegant idea for dealing with the traffic flow of the network, controller will be a bottleneck for the network and essential target for different types of attacks like denial-of-services because its position (middle plane) and it has an abstract view for all network infrastructure. So if attacker dominate on the controller he will compromised all networks forwarding device (data plane) .
- **Channel security challenge:** In this category, the challenges pointed to vulnerabilities in the channel of connection between the architecture component of SDN network also connections between the administrator and network devices. The attacker can exploit from there is no trusted connection between the controller and forwarding devices for compromise the network as an example for this type of attack is man in the middle [5].

2.6 Benefits of New Technology SDN

The key point in SDN architecture is decoupling the control from the data plane, bring with it new benefits for the communication network. For example improving configuration, optimizing network performance and support inventions.

2.6.1 Improving Configuration

Management network in general and particular improving configuration operations considerate most important process especially when we need to modify the network topology by adding or removing devices. This re-configuration operation require modifying whole network. There are various vendors of network devices each one of them with specific management interfaces for these devices with a different set of command this variety make the manual configuration operation be error prone and boring so with the traditional network suffering from many challenges in design and dynamic adjustments .

The amazing architecture of the SDN especially the logical centralized of the network operating system provides the controller with an abstract view for all forwarding devices. This lead to making management of the network be easy because the configuration can be made from a single point and no need to configure each device individually[28].

2.6.2 Optimizing Network Performance

Maximize employment of the network devices represents the main goal of network operations. With traditional networks improving performance focused on a part of the network or some important services that provided to the users. These networks will have suboptimal performance because they depending on the local information without considerate the whole network layers. The logical position of the controller in the middle of the SDN plans (data and application) enable it to has a global view of the network. All exchange information between these layer pass through it this allow the optimization performance be globally in SDN architecture.

With appropriate algorithms that can be run centrally helps in reducing the challenges to improving optimization problems and making these problems programmable and as a result could be developed and used softly [26].

2.6.3 Support Inventions

Many reasons faced the applications developers and researchers in legacy network domain such as a proprietary network infrastructure, tightly coupled between data forwarding, control functionality and standard vendors Command Line Interface (CLI) for configuration network devices. With flexible SDN architecture and enabling programmability of the network, SDN can support inventions .

2.7 Network performance (NP)

It's the quality of services that supported by the computer network infrastructure which can be obtained by analysis the network statistics such as numbers of packets that delivered per unit of time, NP can be represented as a quantitative process or as qualitative, from the customer's perspective it considerate as qualitative process (QoS that supplied to the customers by the network) [29].

2.7.1 Techniques for Performance Analysis

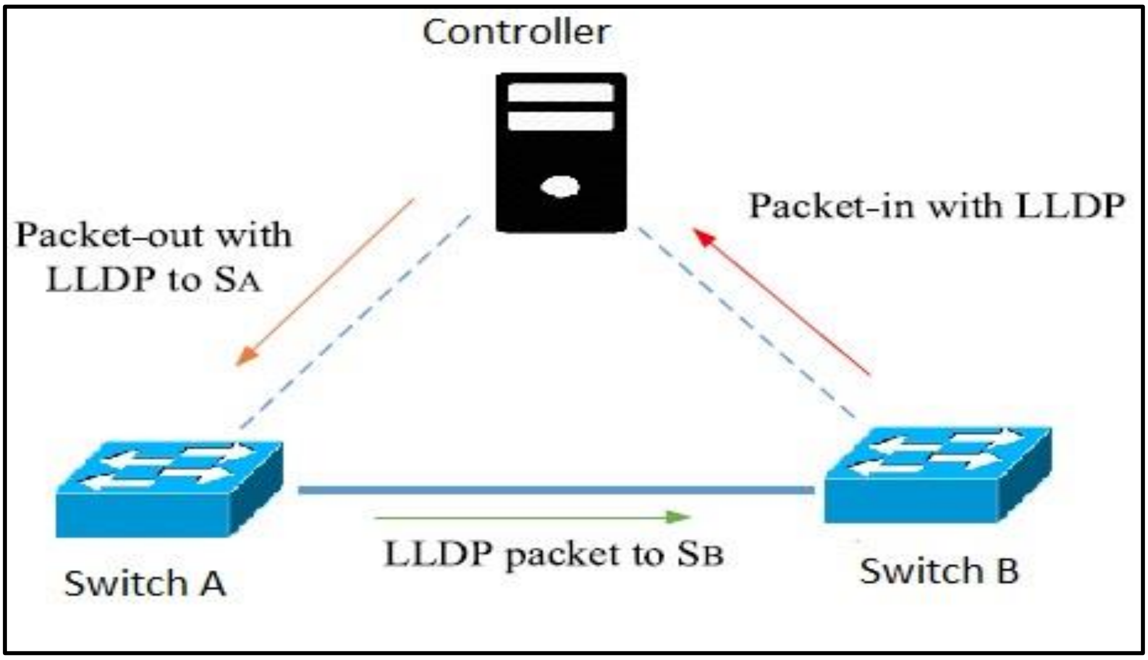
There are three techniques for network performance analysis first one **measurement** represents the basic approaches and can be accomplished other by software or hardware or both, so it will be expensive.

The second one based on building **a mathematical model** which represents as tools for testing the network before real implementation and solve it this approaches called (analytic modeling). The last one **simulation,** its most suitable approaches for studying network problems and checking the performance. It involves on mapping the specific real network aspect to a virtual model [30].

2.7.2 SDN Network Metrics Measurement

It is a set of metrics that are used to describe the state of the network and they considered one of the key elements in the process of network management. There are three types of metrics in the SDN networks topology, traffic, and performance metrics [31].

Topology metrics: This type of indicators point to the network details such as the nodes number, a number of connections between these nodes (links) in the network and state of the switches ports (active or not). Based on link layer discovery protocol (LLDP) the SDN controller discovers the global view of network topology figure (2.5) explain the link discovery process.



Figure(2. 5): Link discovery process

The controller send a messages (packets) for all forwarding infrastructure devices in the data plane after the switches received LLPD packets forwarded to all switches that have a direct link with it These devices, in turn, send messages to controller in format of help message (packet in) because they haven't contained forwarding information in the flow table, when the controller received these messages, analysis them and construct the network topology [32] .

Traffic metrics: Intended number or time spent by the packets to go through the network devices or ports. They are important in the analysis of the behavior of network users. In SDN environment there are two types of traffics the first one called control traffic which represent the traffic flow that constructed from transmitted data between controller and network ; the second called data traffic which represent the traffic flow that constructed from transmitted data between switches.

The Statistical information collected from network forwarding devices such as number and size of packets helping to represent the characteristics of the flow.

performance metrics: there are numbers of metrics that are used for checking network performance, types of used metrics usually depending on the use case . Listed below some of the most important metrics:

- capacity: refer to how much traffic can transfer system (link, path) cop and it's measured by bit per second or packet per second.
- throughput: the amount of transfer bit in time.
- delay: time taken to transfer the packet from source to sink.
- packet loss: refer to the number of dropped packets.

The two most important metrics are throughput and delay time, the first metric network bandwidth which also called throughput can be determined by amount of transferred bits in specific periods of time in the network . For instance network with a bandwidth 100 million bits/second , declared that ability of the network to deliver 100 million bits every per second and also we can considerate the bandwidth as time-consuming to transfer each bit of data .

The other metric latency (delay time)which represents the time taken by the message for traveling through two ends in the network (can also compute latency for a single link in the network). On the other hand (RTT) which is the abbreviation for Round Trip Time is the time taken by the message for traveling from source to the destination and back again [33].

2.7.3 Tools for Measuring Network Metrics

There are two of tools that used along metrics and both of them gave a complete picture on the capabilities of the network; iperf and ping.

Ping: is a standard tool used for checking remote device reachability and its part of the most modern operating system (windows, Linux, MAC) in addition, it may be a part of some software network devices such as routers.

Most ping tools used Internet Control Message Protocol (ICMP) in their works these tools compute the interval time from sending the request (message) to the specific address until getting the respond message. They came with options such as a number of sending request times, size the request, and waiting time for getting the replay. The standard output contains IP address for the target, time spent between sending and responding in the millisecond and in the case of unreachable target the output will be an error message.

Iperf : is another network metric tool used for measuring the throughput metric by creating other TCP or UDP datagram, it come with multiple options that help in measuring process such as determining the size of datagram in case of using UDP. Iperf came with two configuration options client on the side and server on the other side, the output of this tool is a report contain the time and amount of data that transferred between client and server [34].

2.8 Power Consuming Model for SDN Switches

In general, there are two main factors that effect in energy consumption in the computer network, device factor and traffic factor. The first factor (device) which include an amount of energy consuming by device components such as device chassis, line cards number , active ports and their configurations and Ternary Content Access Memory (TCAM) if exist.

The second factor (traffic) which represents in packet size, delay time for delivering the packet and amount of CPU for processing the traffic. The traffic factor can be neglected as compared to the amount of energy consuming relevant to the device factor[35].

For modeling the energy consumption based on a network device (open flow switch) the equation below define energy consumption model.

$P_{OFswich} = P_{chassis} + P_{OFT} + P_{control} + P_{configuration}$----------(2.1)

Where

$P_{chassis}$: energy of switch chassis.

$P_{configuration}$: energy of configuration(number of active ports, configuration of the line speed)

$P_{control}$: energy consumption of traffic controlling.

P_{OFT}: energy consumption for processing OpenFlow traffic.

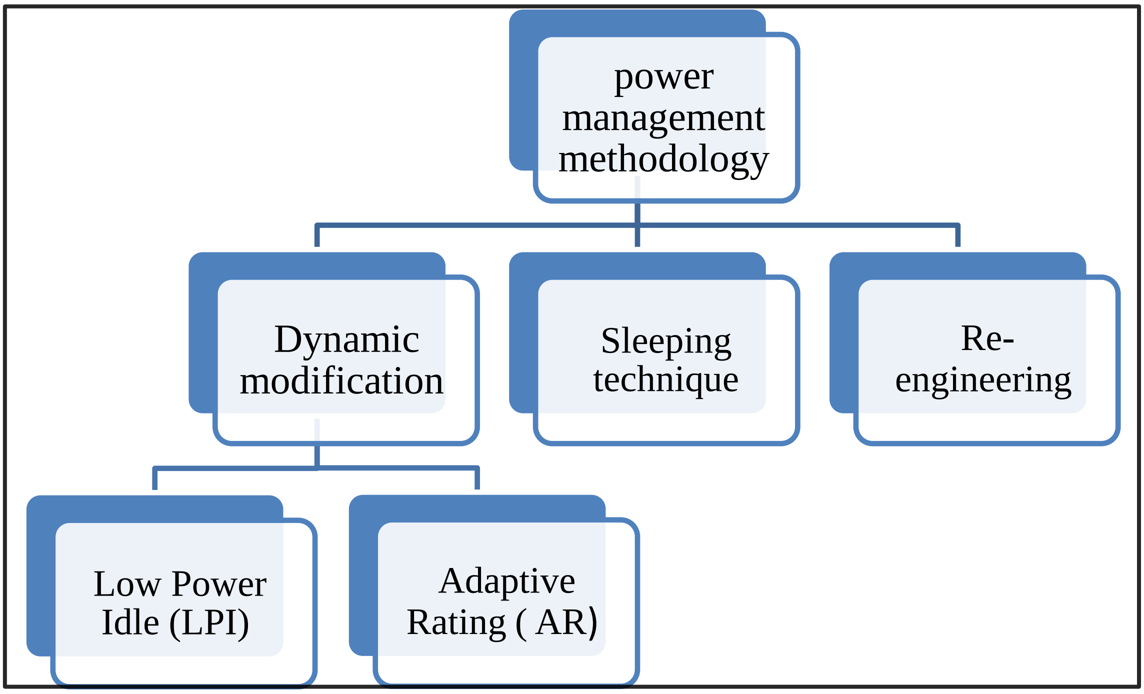
Consider to the (NEC PF 5240) OpenFlow switch and as shown in the table (2.1) in the case of the full active switch (chassis + ports) will consume about (122 W) in the case of 4 port switch (118W for chassis and 1W for each 1Gbps port) [36].

Table(2. 1): Power Consumption in NEC Switches

power chassis / W	port configuration	
	port speed	power consumption / W
118	1 Gbit	1
	100Mbit	0.4455

2.9 Network Power Management Methodologies

Because the network devices are considered as part of computer devices so the power management methodologies can be derived from enhancing the strategies that correspond to existing techniques in computer approaches. These technologies can be classified as re-engineering, dynamic modification and sleeping technique [35]. Figure (2.6) demonstrate the power management methodologies.



Figure(2. 6): Power Management Methodologies

2.9.1 Re-engineering Method

This method based on exploiting the new energy-aware technologies, devices, and equipment in the network architecture such as memories, protocols for improving energy consumption.

2.9.2 Dynamic Modification

This method allows modifying the capacity of the services based on instant traffic load and this can be done by two technologies named Low Power Idle and Adaptive Rating.

The Adaptive Rating (AR) allows reducing the device working rate dynamically and this will introduce a trade-off between device performance and power consumption. While with the Low Power Idle (LPI) technique allows switch-off some network component when they have no activity and re-work them again speedily as possible when the system needs their activity for power saving purposes.

In the case of the computer network, the most suitable component that helps in reducing a large amount of power based on (LPI) technique are nodes which represent by (routers, switches) and links between them.

2.9.3 Sleeping Technique

This technique aims to reducing the power consumption by putting some network component in sleeping mode in non-peak time and weak up them again when the system need employ them.

2.10 Energy Saving Methods in SDN

Energy saving can be done at different components that construct SDN architecture or can use SDN by meaning of power saving, solutions for reducing the energy consumption in SDN network can be classified either software based or hardware based.

The first solution (hardware – based) can be practically applied to the forwarding devices of the network, while the second solution (software-based) can be applied by the controller. Three methods founded for energy aware strategies in SDN network (compressing TCAM rules, traffic aware, rule placement) [37].

1-Traffic Aware Strategy:

Turning off some links in non – peak time (low traffic load intervals) and re _routing the packets to another link or setting CPU or forwarding ports in sleeping mode represent the main idea about this solution.

2-Compressing TCAM Rules:

Ternary Access Control Memory (TCAM) represents a significant part in every high-performance network forwarding device. In SDN network TCAM use to store rules in OpenFlow switch that are used in matching operation for incoming packets and take appropriate action based on this rules.

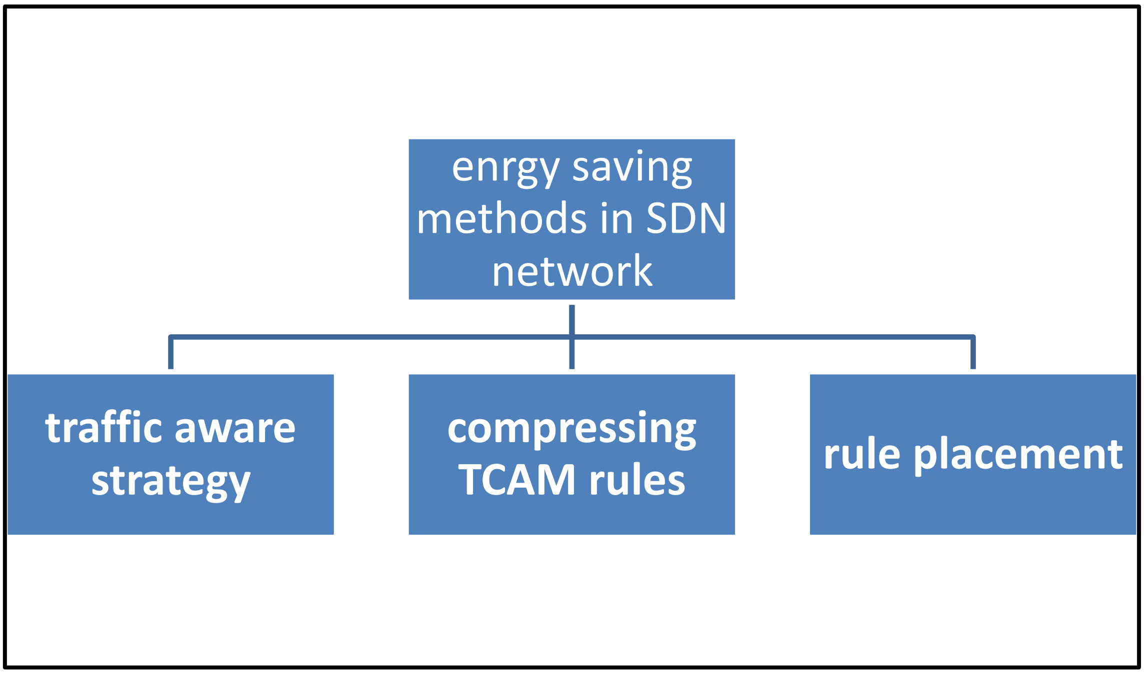
The most energy consuming in the TCAM came from number of search operations in the memory entries, so the trade-off between the performance and energy consumption is the most important characteristics of this memory. It offer high speed lookup operations (one cycle per search operation) and Conversely TCAM will be expensive and consume a large amount of energy .

Compression to the content or the rules represent the most suitable solution for energy consumption in the TCAM. The compacting operation means reducing the number of bits that are used for storing matching packet information.

3-Rule Placement:

Flow table which is implemented in the TCAM inside the forwarding switch represents the base for any routing decision in this switch, each flow table consists of a set of flow entry the matching operation against this entry depending on the matching rules that set and pushed to the switch by the controller.

Rule placement distribution across forwarding switches and enhancing these rules represent an objective function which can be exploited in energy saving in the SDN networks and the number of active switches, delay time, and link capacity represent the constraint of this objective function.



Figure(2. 7): Energy Saving Methods in SDN

2.11 Routing Algorithms

Communication network consists of number of nodes, in SDN network the node will be routers or switches and links that connect these nodes. Each node represent one of two sides either source (side that send the packets) or destination (side that receive the packets) [38].

The process of forwarding the packet along path between source and destination nodes called routing [39]. Routing algorithms are used for determine the most suitable path in the network for forwarding the packet based on algorithm class. The algorithm class either **user-oriented class** which is used for satisfy a specific users needs such as provide them with a good services or **network-oriented class** that work to provide most users with an acceptable service.

The optimal path is not necessarily be the path with the shortest distance between source and destination nodes but can be a path that saves the energy in the network by allowing the link to forwarding more than one flow and turn off the reset of links .

Dijkstra algorithm is one of the earliest Single-Source Shortest Paths algorithms [40]. Suppose we have a graph G (N,E) which consist of set of nodes N and set of weighted edges E (non-negative edges) and a single source node S. Dijkstra algorithm compute the shortest path to the destination node and in addition compute shortest path for all destinations nodes.

Chapter 3

Model Description

Chapter 3

Model Description

3.1 Introduction

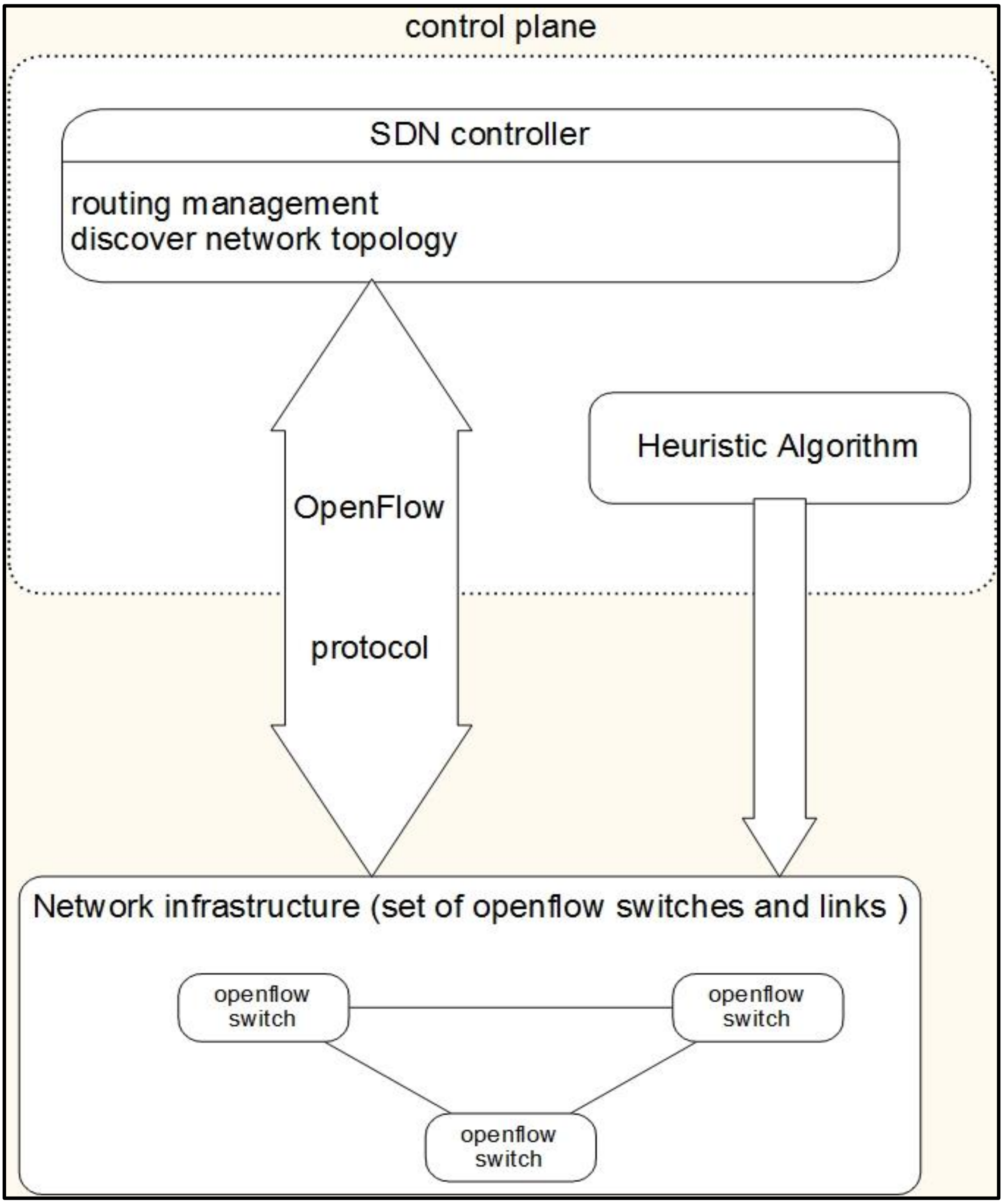
This chapter presents the proposed methodologies that used for reducing the energy consumption in SDN network by using dynamic modification method and explains why this method is used, we also explain the main component that is used to build this model, model architecture and the used algorithms.

3.2 Proposed Model Architecture

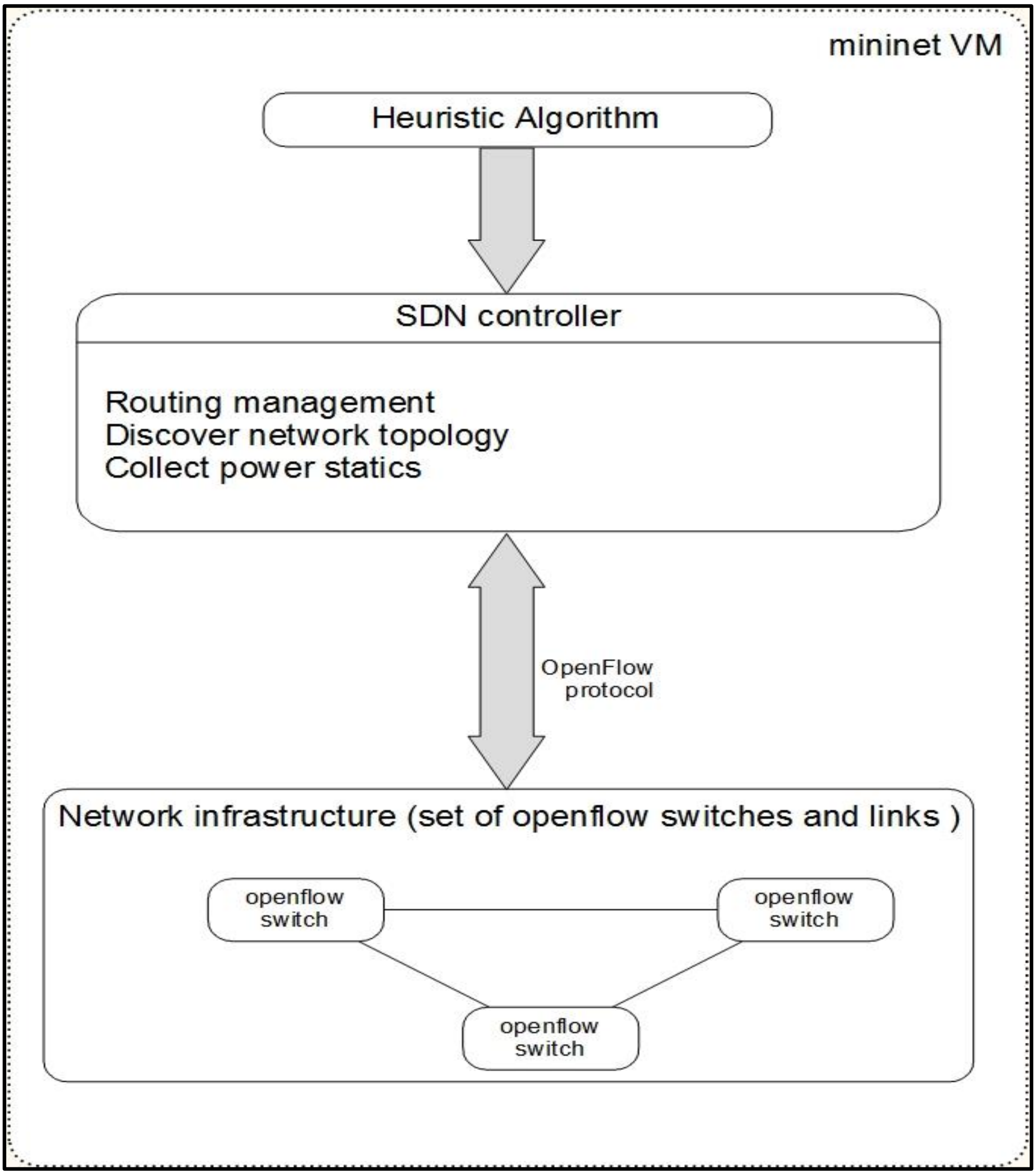
The architecture of the proposed model is shown in Figure (3.1) which consists of three major component: controller, heuristic algorithm and network infrastructure from SDN network perspective.

The heuristic algorithm with its input information such as network topology, traffic flow, build sub-network that is used for forwarding the traffic that satisfies the constraint (traffic flow volume and link capacity).
After the heuristic algorithm finishes its duty, outputs the virtual topology to the controller. After that, the controller changes the state of the switch and ports of data plane layer based on the virtual topology and then routing the traffics. The control plane connects with the OpenFlow switches in the data plane via OpenFlow protocol.

From the mininet simulator perspective, the heuristic algorithm cooperate with controller as scheduling algorithm that is used for determining the most suitable paths for forwarding the traffics and communicate with the controller to manage the routing and power management. Figure (3.2) explains proposed model architecture from the simulator perspective.



Figure(3. 1): Software Architecture from SDN Perspective



Figure(3. 2): Model Architecture form Simulator Perspective

3.3 Dynamic Modification Methodology

Based on energy saving methodology in chapter two in the article of network power management methodologies, find Low Power Idle (LPI) is most suitable for this project as explainedin the following reasons:

- The highest energy saving is obtained when network devices are really powered off (performance of devices equal to zero). On the contrary, other methods (Re-engineering, Sleeping technique).
- Network devices operating at maximum capacity in case of the unconditional amount of Energy Consumption.
- Programmability and flexibility of SDN networks in general and particularly the controller facilitate finding centralized algorithm solutions for the energy consumption problem.
- Unable to apply Re-engineering approaches because it requires developed special hardware.

3.4 Energy Saving Problem Statement and Formulation

The amount of energy consumption in the networks results primarily from the amount of energy consumed by active network devices, such as routers or switches. In the case of SDN network the devices will be OpenFlow switches. The energy consumption for each OpenFLow switch is affected by the following factors :

- Total active ports in the switch.
- Capacity of the port.
- The power consumed by switch chassis.

According to the factors above and by using NES Openflow switch model in chapter two and taking into consideration to the power that consumed just by switch chassis and active port , we can compute the power consumption as below:

$$P_{switch} = P_{chassis} + W \sum_{i}^{N} port_i \quad \text{------------------------------------(3.1)}$$

Where

P_{switch} total power consuming by the switch.

$P_{chassis}$ amount of power consumed by the chassis of active switch.

W amount of energy consumed by the port .

$port_i$ active port i.

N total number of active ports.

By taking assumption that each port operate at rate 1Gbit so it will consume 1Watt, the energy consumed by the active ports will be $\sum_{i}^{N} port_i$.

The total power consumption of the network will be the summation of the power of active switches and as below :

$P_{network} = \sum P_{switch}$ ---(3.2)

By submitted equation (3.1) in (3.2) the total power consumption of the network will be:

$P_{network} = \sum\left(P_{chassis} + \sum_i^N port_i\right)$.-------------------------------(3.3)

And because each two active port are connected by one link so the total power consumption of the network will be:

$P_{network} = \sum P_{chassis} + \ (2 * \sum_i^n l_i)$.------------------------------(3.4)

Where

$l_i \in L$, $i = \{1,2,3, \dots\}$

n total number of network links

l denoted to the link in the network.

3.5 Network Model

The network topology modulates as an undirected graph denoted by $TOPO = (S, L)$, where S is a set of nodes which represent the open flow switches, and L represent the set of links of the network, each single link $l \in L$ is used for forwarding the traffic between two switches via them ports , C_l represent the capacity of the link l. The process of forwarding the traffic among the switches by selected paths is managed by higher plane called control plane which represented by the controller.

For reducing energy consumption in the proposed model we depending on the dynamic modification approaches in general and specifically low power ideal (LPI). Minimizing number of active network elements (switches) and number of active ports per switch represent the objective function of this model and as explain in equation (3.5) to satisfy the network constraint (link capacity, traffic volume). Where the traffic demand should be less than each link capacity in the selected path, equation (3.6) clarify this point.

Objective function :

$min(\sum S_{active} + \sum L_{active})$ --------------------------------------(3.5)

Where

S_{active} denotes to the active switches.

L_{active} denotes to the active links.

Network constraint :

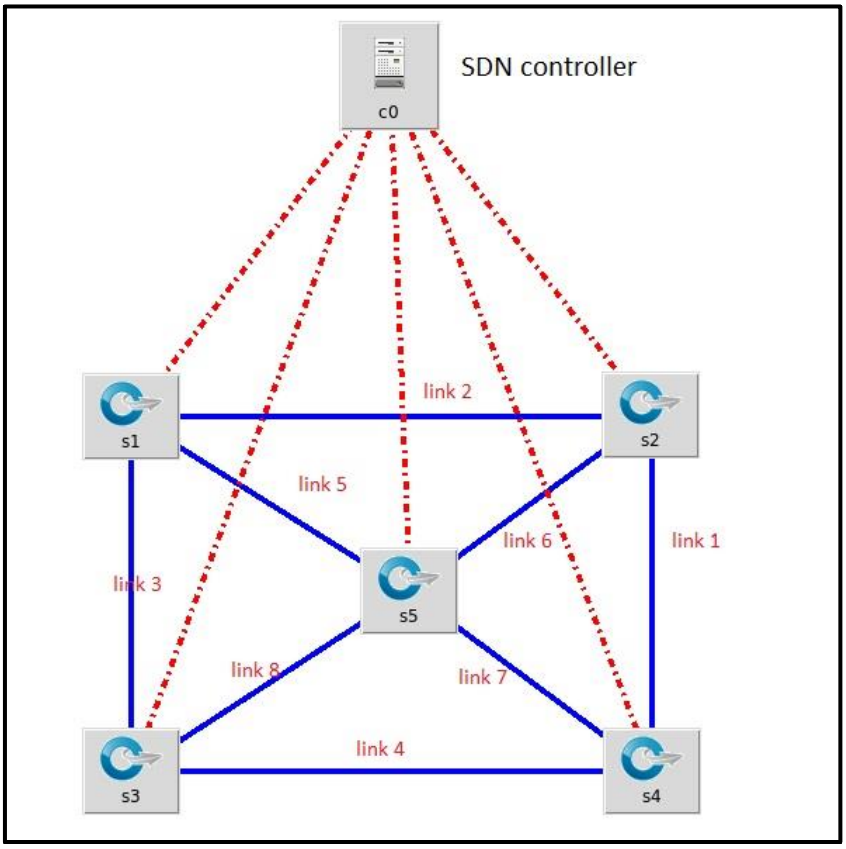
$C_f \leq C_{l,i}$ --(3.6)

Where

C_f denote to the traffic volume.

$C_{l,i}$ denote to the capacity of the link (i) because each path consist one or more of links.

each two hosts in the network could be connected via more than one path, we denoted these set of paths as $\boldsymbol{all_{paths}}$ and each of these paths consists of one or more links and nods such as in figure (3.3). Where there are more than one path between (S1) and (S2) like {(s_1, s_2), (s_1, s_5, s_2), (s_1, s_3, s_4, s_2),…. }. We assume that the constrain (3.6) of the network is satisfied. Then, the objective function for these three paths declare in the table (3.1)



Figure(3. 3): Network Model.

Table(3. 1): Objective Functions for Possible Paths

paths	nodes	no. nodes	no. links	objective function
path-1	s1,s2	2	1	3
path-2	s1,s5,s2	3	2	5
path-3	s1,s3,s4,s2	4	3	7

The most suitable path is (path-1)because it release the objective function of equation (3.1) which has minimum number of active switches and links. Consequently will consume less energy than other paths.

3.6 Work Details

Initially proposed model is divided into three stages preparation and traffic matrix generation, routing process and building virtual topology and computing the power consumption as shown in figure(3.4) .

3.6.1 Preparation & Traffic Matrix Generation

The first stage includes parsing of all the necessary settings, whether by console of the simulator or from an external file. These setting involves network topology, the method that is used for routing the traffic and constructing the virtual topology, determine the power model and amount of traffic demand.

The network is represented by a specific data structure that can be used to retrieve information for finding the most appropriate path between the two nodes according to the used algorithm, such as the shortest path algorithm (Dijkstra). This network differs from the virtual network topology the latter is incrementally formed during the next stage(routing process). The method for routing the traffic and the power model have to be established in this stage for using later .

Amount of the traffic for each node-pairs is important for traffic matrix constructing because the proposed algorithm depending on it, so it will establish in advance .

3.6.2 Building Virtual Topology

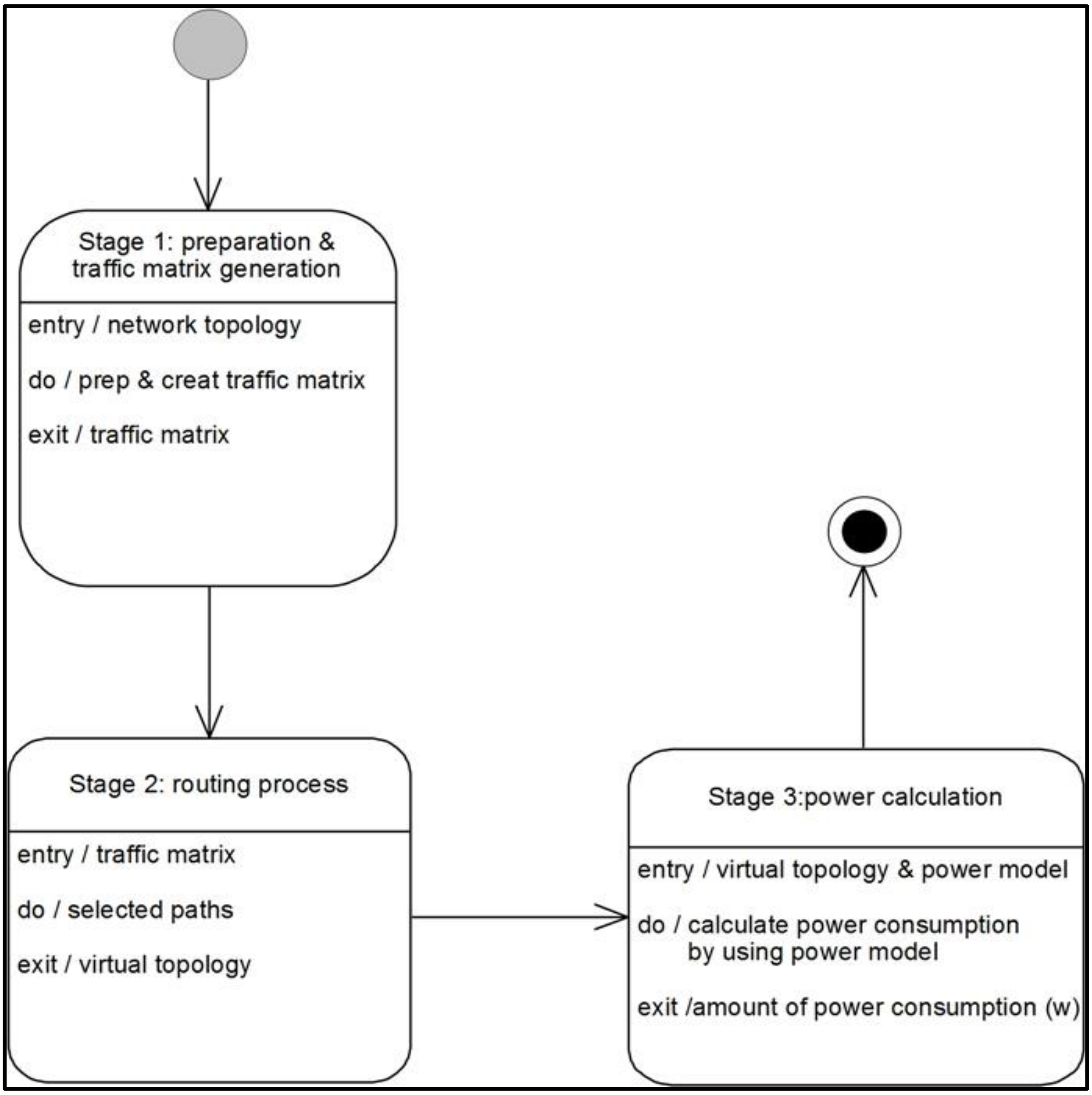
The virtual topology constructing incrementally after each forwarding request from the traffic matrix by proposed algorithm. Where each traffic will have a path that consists of number of links. When a new request (traffic) is introduced if there are no resources (paths) found for accomplish this request, a new path will arising for serving this request and Thus bring more energy to be consumed.

The amount of power savings varies depending on the method and the used algorithm, in this work we use Dynamic modification method that mentions early in chapter two with a proposed algorithms that base on off-line technique where it use history of previous forwarding traffic operations so it will repeatedly visit virtual network topology, On the contrary, on-line technique, where each request(traffic) will accomplish independently from the previous one.

3.6.3 Computing The Power Consumption

The purpose of this phase is to analyze the virtual network topology resulting from the previous stage at the individual level and determine the amount of energy consumption per component in this level.

In this model, the main component will be number of the switches in the virtual topology and number of active ports per switch according to the power model of NEC OpenFlow switch that mentioned early in cahpter two.



Figure(3. 4): UML State Diagram of Proposed Model

3.7 Heuristic Algorithm

It's difficult to find the optimal solution because of the diversity and the many factors that influence the reduction of energy consumption. Therefore the heuristic algorithm gave near to optimal solution according to the set of constraints such as the volume of traffic between the source and destination nodes and capacity of the links.

Heuristic algorithm decompose to the three sub-algorithm :

- Algorithm (A): This algorithm explain in figure (3.5) and figure(3.6), takes a traffic matrix and network topology as input where each entry in the traffic matrix consist of the pair of hosts associated with traffic demand. The algorithm initially generates all the possible paths between the source and sink nodes(candidate set) then select the most appropriate path(Path with least energy-consuming) as the objective function. Where the path is consist of a set of nodes and edges that connect between these nodes.

Input : traffic matrix ,network topology

Output : set of active switches ($\boldsymbol{total_{switches}}$)

10 start $\boldsymbol{total_{switches} = \emptyset}$, $\boldsymbol{count = 0}$.

20 for $\boldsymbol{count \leq length\ of\ traffic\ matrix}$:

30 generate all possible paths $\boldsymbol{all_{paths}}$ for traffic $\boldsymbol{traffic_{count}}$

40 for each $\boldsymbol{path} \in \boldsymbol{all_{paths}}$:

50 if $\boldsymbol{path}$ is valid then $\boldsymbol{valid_{paths}} \leftarrow \boldsymbol{path}$

60 for each $\boldsymbol{Valid_{path}} \in \boldsymbol{valid_{paths}}$:

70 compute path capacity $\boldsymbol{Valid_{capacity}} \leftarrow \boldsymbol{Valid_{path}}$

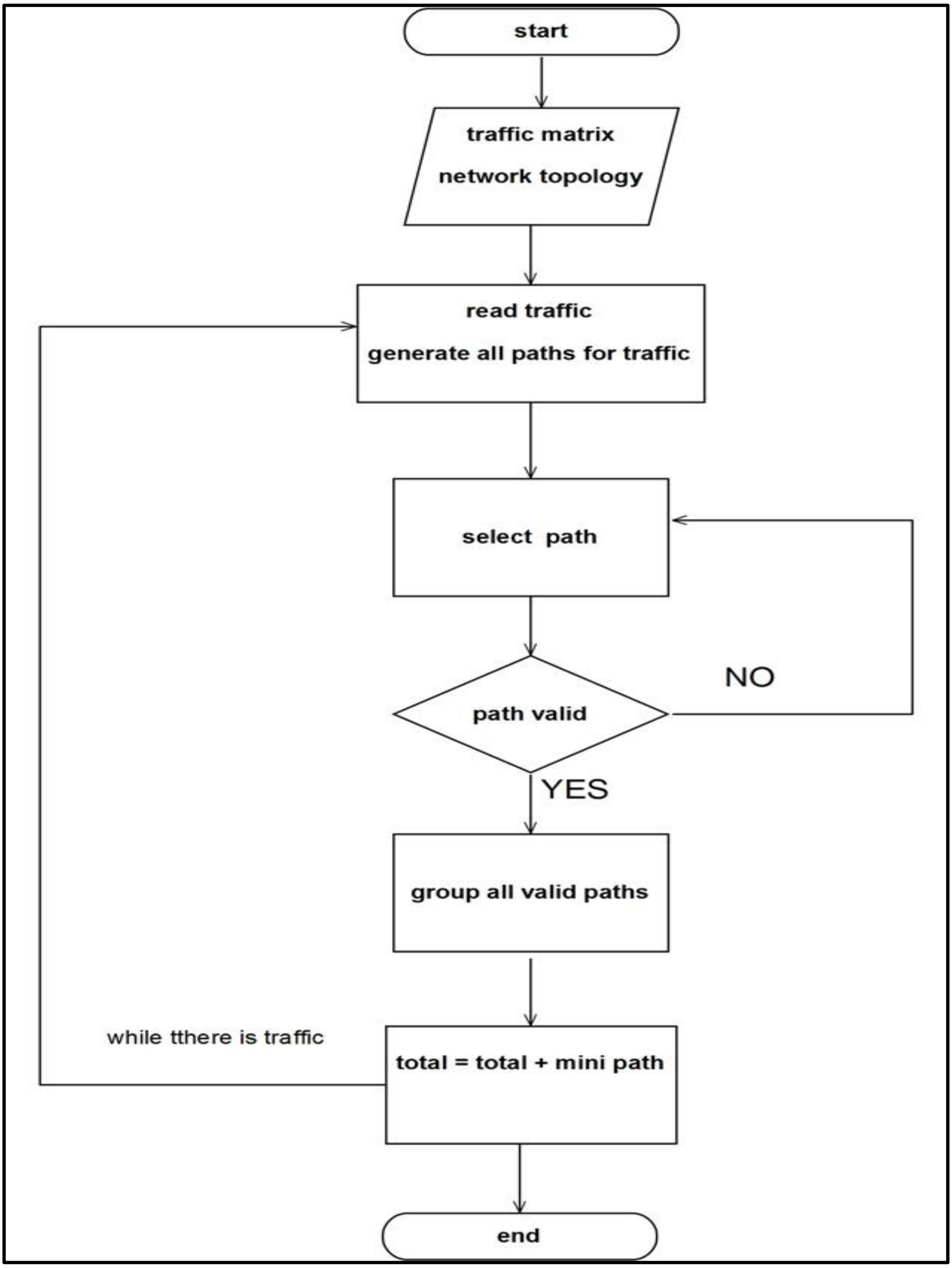
80 $\boldsymbol{correct_{path}} \leftarrow \boldsymbol{minimum}$ ($\boldsymbol{V_c}$)

90 $\boldsymbol{total_{switches} = total_{switches}} \cup \boldsymbol{correct_{path}}$

100 update topology ($\boldsymbol{correct_{path}}$)

110 $\boldsymbol{count}$ = $\boldsymbol{count}$ + 1

Figure(3. 5): Algorithm (A) Pseudo Code



Figure(3. 6): Algorithm (A) Flow Chart

- Algorithm (B): this algorithm acts as a filter for all possible paths between source and destination nodes. The output of this algorithm will be a set of paths in which the capacity of each link in these paths is greater or equal to the traffic. Figure (3.7) shows the pseudo code to this algorithm and figure (3.8) represents the flowchart.

Input : $\boldsymbol{path}$, traffic size of $\boldsymbol{traffic_{count}}$ ($\boldsymbol{traffic_{size}}$) from algorithm (A)

Output : Boolean (True OR False)

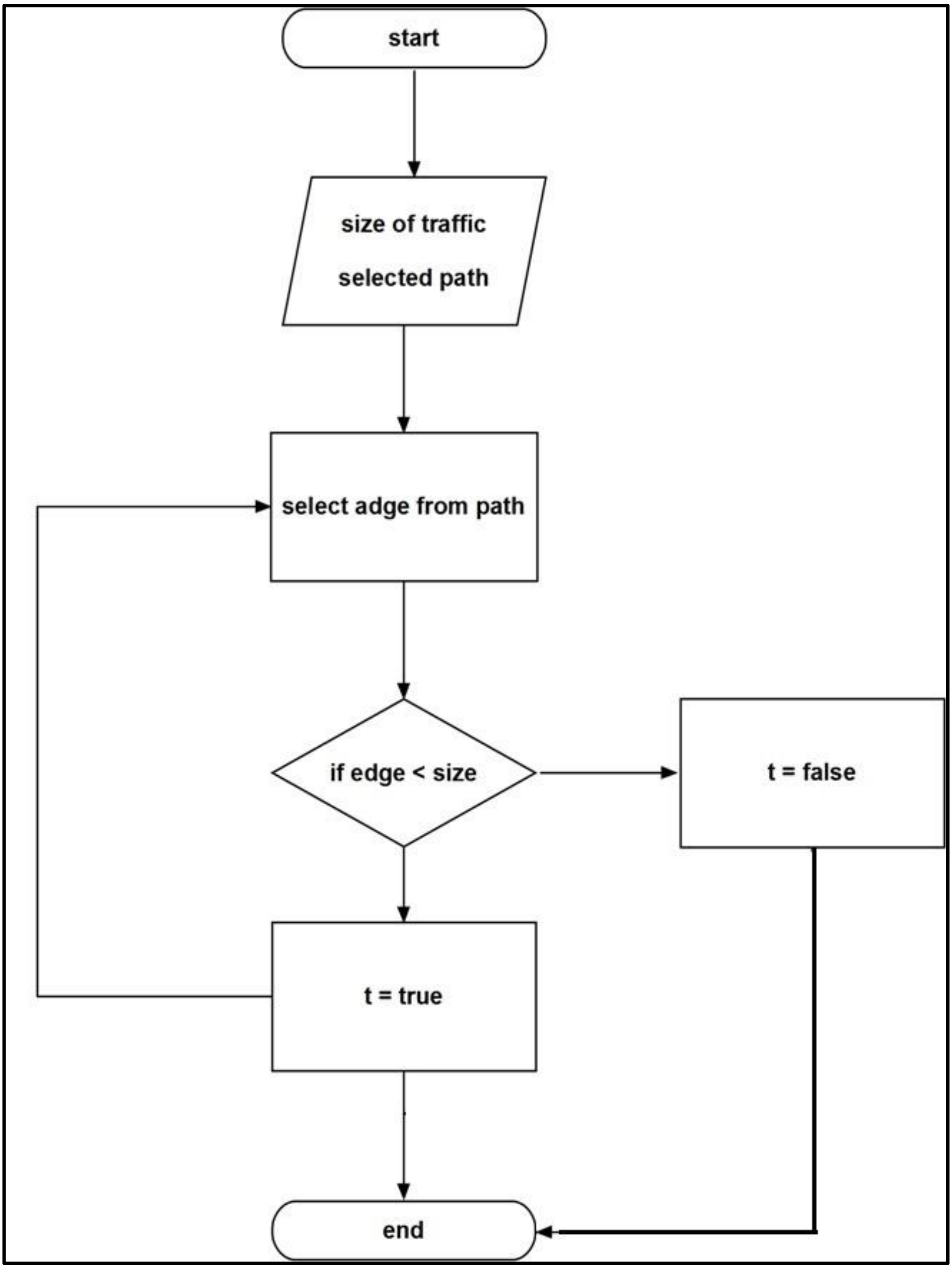
for each $\boldsymbol{edge} \in \boldsymbol{path}$:

    if $\boldsymbol{traffic_{size}} > \boldsymbol{edge_{capacity}}$ then

        return (False) break

    else return (True).

Figure(3. 7): Algorithm(B) Pseudo Code



Figure(3. 8):Algorithm(B) Flow Chart

- Algorithm (C): the goal of this algorithm is updating the links capacity of the virtual network topology after forwarding the traffic to the selected path and deactivate the unused paths. Figure (3.9) shows the pseudo code to this algorithm and figure (3.10) represent the flowchart.

Input : network topology , $\boldsymbol{correct_{path}}$, $\boldsymbol{traffic_{size}}$

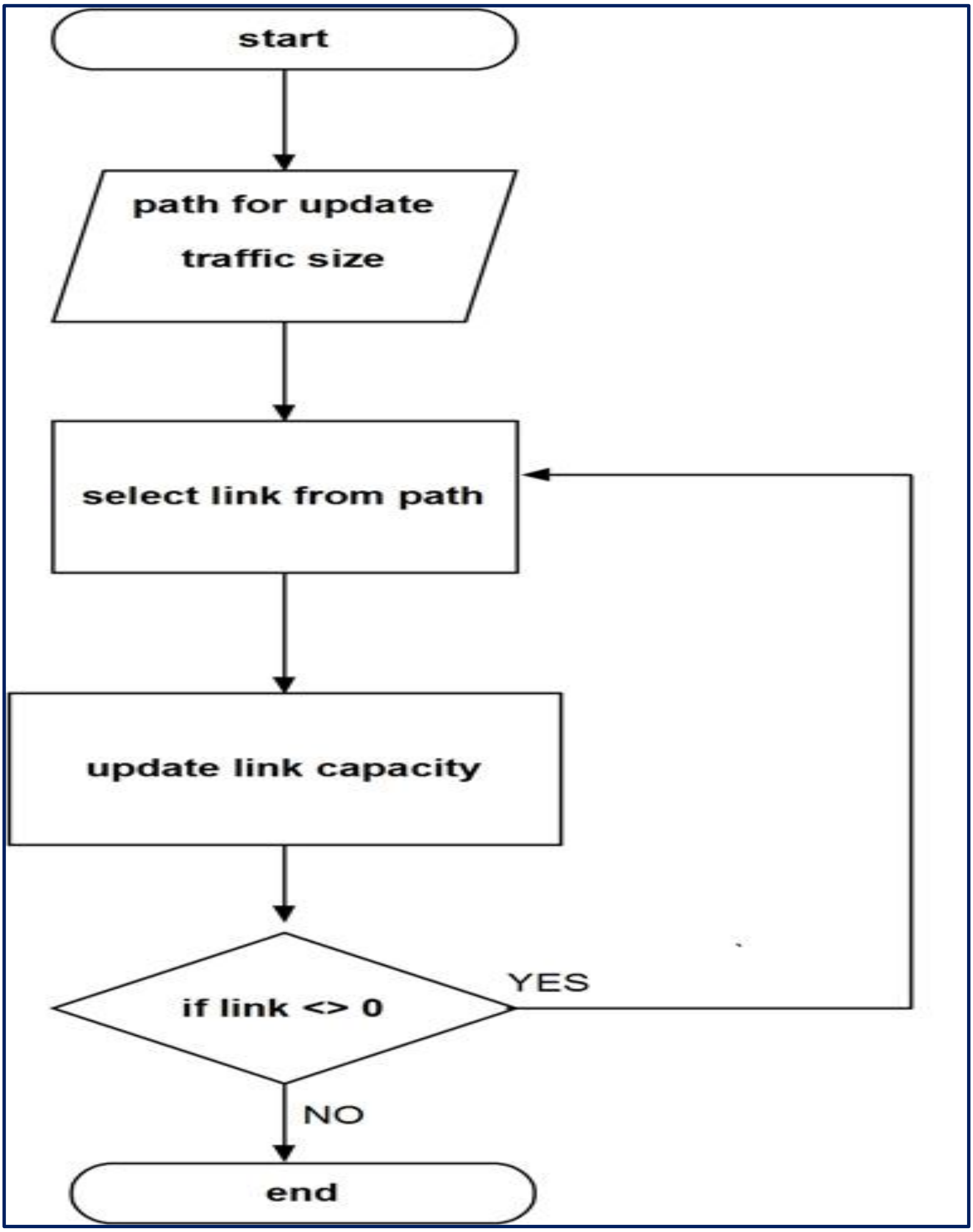
Output: updated network topology.

for each $\boldsymbol{edge} \in \boldsymbol{correct_{path}}$:

$\boldsymbol{edge_{scapacity}} = \boldsymbol{edge_{capacity}} - \boldsymbol{traffic_{size}}$

return network topology with new edges capacity .

Figure(3. 9):Algorithm(C) Pseudo Code



Figure(3. 10):Algorithm(C) Flow Chart

Chapter 4

Performance Evaluation

Performance Evaluation

4.1 Introduction

In this chapter, the evaluation of the proposed model is presented by performing experiments using different types of network topology such as Abilene, fat tree datacentre topology, two-tier datacentre topology and three tier datacentre topology. In addition to using special tools for testing the network performance such as iperf and ping, and clarify how this algorithm can reduce the energy consumption by reducing the number of active nodes comparing to the Dijkstra algorithm.

4.2 Simulation Setup

Different tools (hardware and software) are used for testing proposed model, python interpreter is used for generating the topologies and implementing the algorithm. Mininet simulator is used in executing pox controller and checking network performance by using ping and iperf tools . Table (4.1) clarify the topologies that used in experiments

Table(4. 1):Network topologies used in experiments

Scenario number	Topology name
1	Simple topology
2	Abilene NW
3	Fat tree 4 pod
4	2 tier datacentre
5	3tier datacentre

4.2.1 Simple Network

At the first we used a simple network as specified in table (4.2) with traffic matrix consist of two flows as shown in the table (4.3) after forwarding the traffic matrix through the network based on proposed algorithm. It is founded that the number of active nodes (switches) is 5 where first flow active 4 switches (0, 1, 2, 3) and the second flow active 4 switches (5,1,2,3) by using one new additional switch (switch 5).

Table(4. 2) : Simple Network Parameters

network topology	simple topology
number of nodes	6
number of links	8
link capacity	10MB
traffic matrix size	2×2
size per traffic	500 KB

The proposed algorithm is compared to Dijkstra algorithm which depending on number of hops to find shortest path. . It is founded that the number of active nodes (switches) is 6 where first flow active 4 switches (0, 1, 2, 3) and the second flow active 3 switches (5,4,3) by using two new additional switches (switch 5 and switch 4), and as shown in table (4.4).

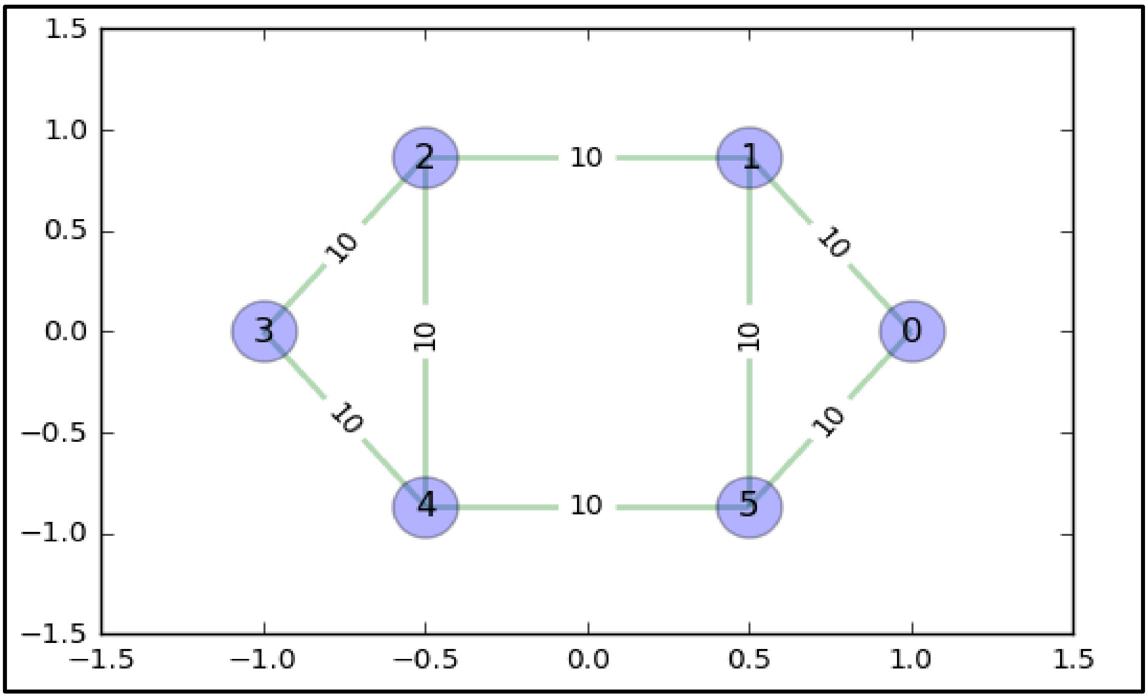
Table(4. 3): Traffic Matrix

source	destination	traffic size
0	3	100KB
5	3	100KB

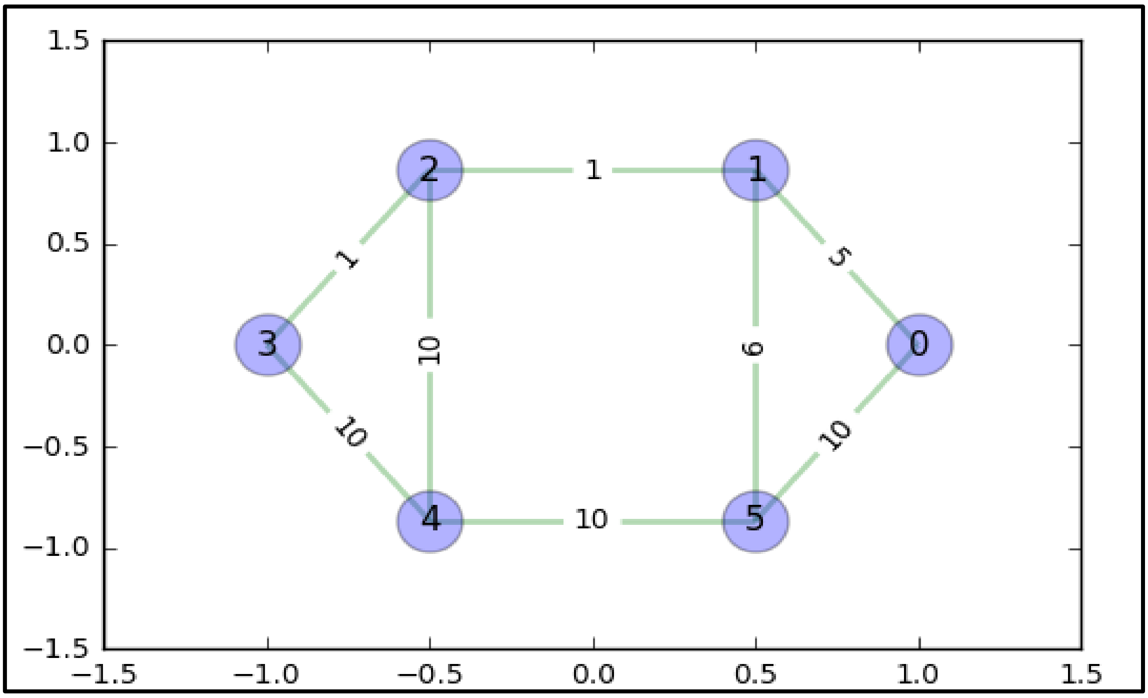
Table(4. 4): Active Nodes Based on Proposed and Dijkstra Algorithm

flow number	number of active switches		selected path	
	proposed algorithm	Dijkstra algorithm	proposed algorithm	Dijkstra algorithm
1	4	4	0, 1, 2, 3	0, 1, 2, 3
2	1	2	5, 1, 2, 3	5,4,3
total number active nodes			5	6

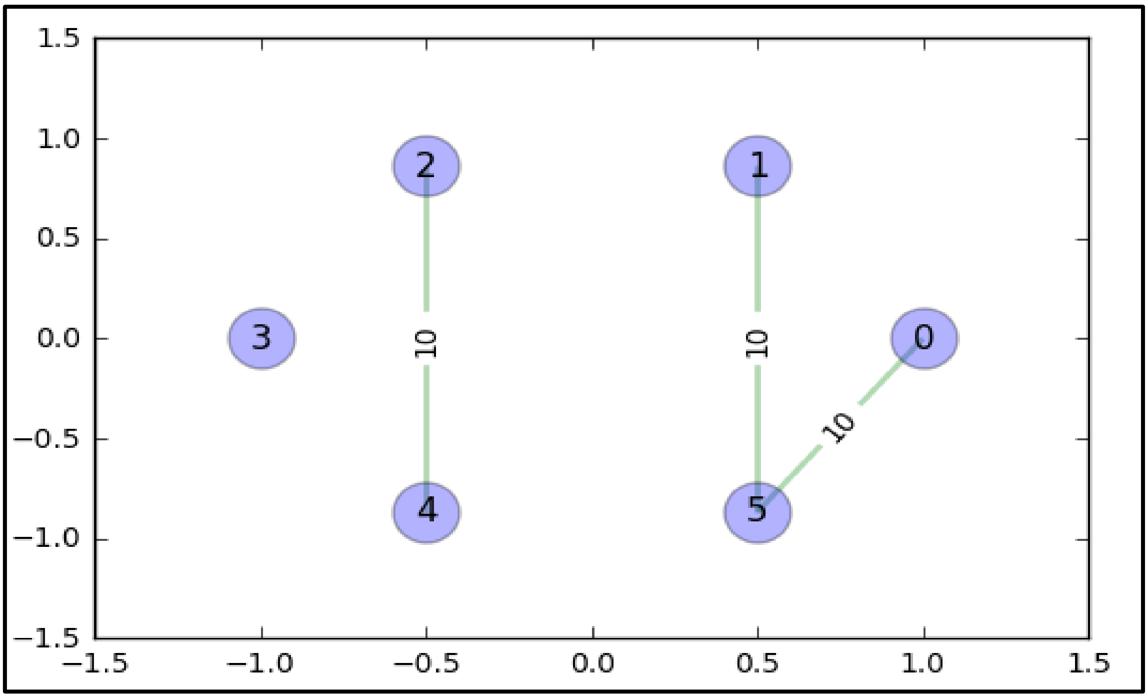
The proposed algorithm adaptively changing the link capacity according to the traffic size so multiple flows can share the same link. Figure (4.1) shows the original network before forwarding the traffic and, figure (4.2) shows the residual of links capacity of network after forwarding the traffic when proposed algorithm is used. Figure (4.3) shows the network after forwarding the traffic when Dijkstra algorithm is used and removing the used links, Figure (4.4) shows amount of energy consumption between proposed algorithm and Dijkstra algorithm.



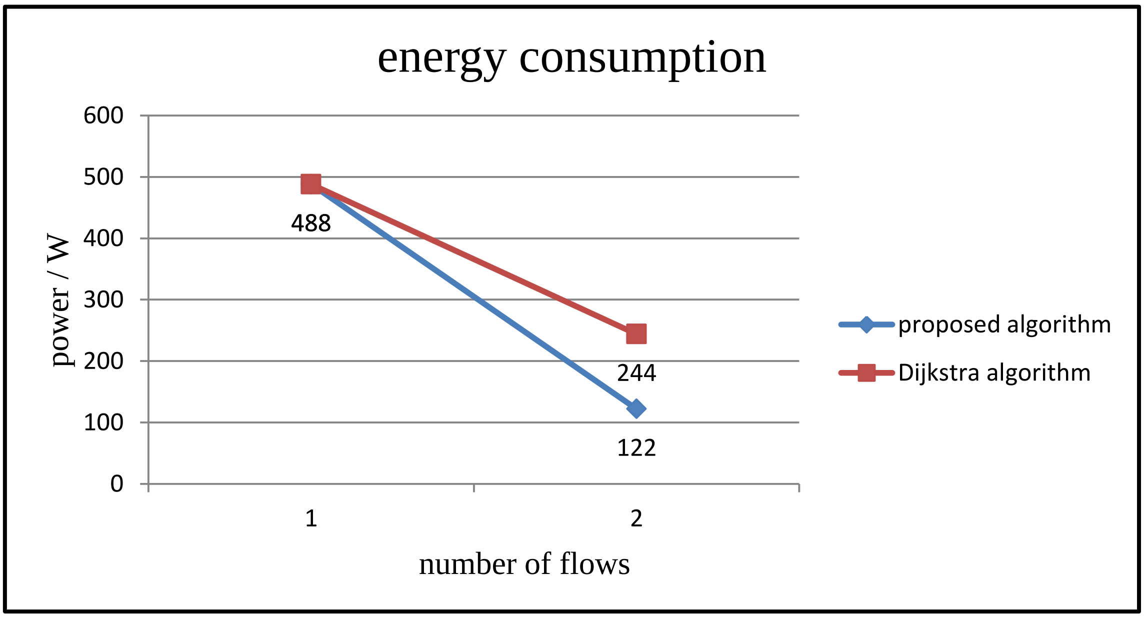
Figure(4. 1): Original Network Topology



Figure(4. 2): Links Capacity After using Algorithm



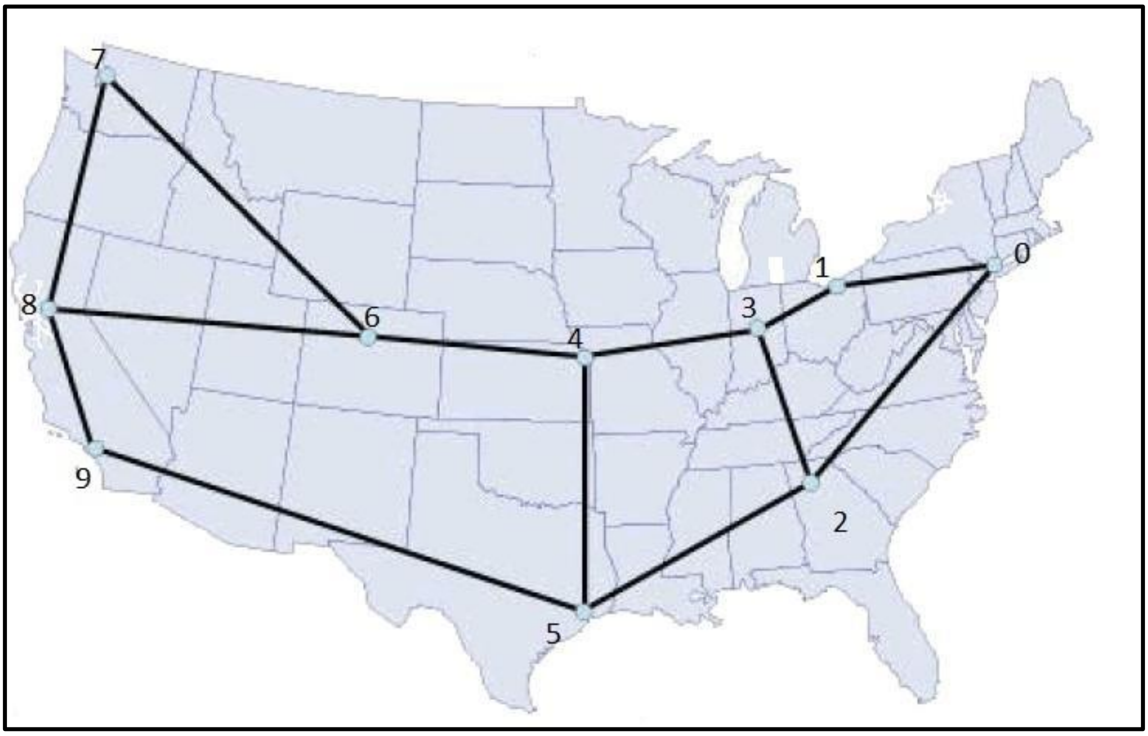
Figure(4. 3): Residual Links After Using Dijkstra Algorithm



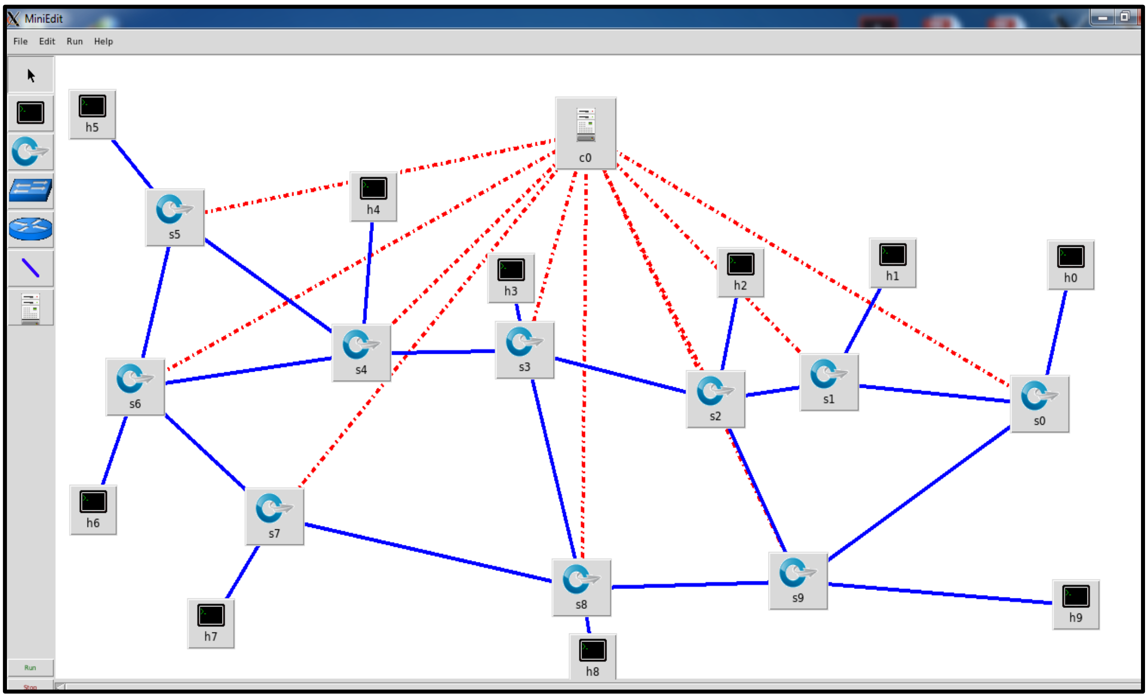
Figure(4. 4): Energy Consumption

4.2.2 Abilene Network

Abilene network is a high-performance backbone network. Figure (4.5) shown this core network topology. The Miniedit application with graphical user interface runs inside mininet simulator is used to constructed Abilene network topology. The constructed topology is described in figure(4.6)



Figure(4. 5): Abilene Network Topology



Figure(4. 6): Constructing Abilene NW in miniedit

The proposed algorithm is applied with simulation parameter as shown in the table (4.5) and traffic matrix in the table (4.6).

Table(4. 5): Abilene Simulation Parameter

network topology	Abilene topology
number of nodes	10
number of links	13
link capacity	10MB
traffic matrix size	2×2
size per traffic	100 KB

Table(4. 6): Traffic Matrix

source	destination	traffic size
0	6	100KB
9	6	100KB

First flow needs to active 5 nodes for performing the forwarding operation , while second flow need no active any additional nodes for forwarding the packets because the path already activated from previous flow by using proposed algorithm. On the other hand, if Dijkstra algorithm is used 3 new nodes need to be active for forwarding the second flow that represented in table (4.7).

Table(4. 7) : Active switches and selected paths

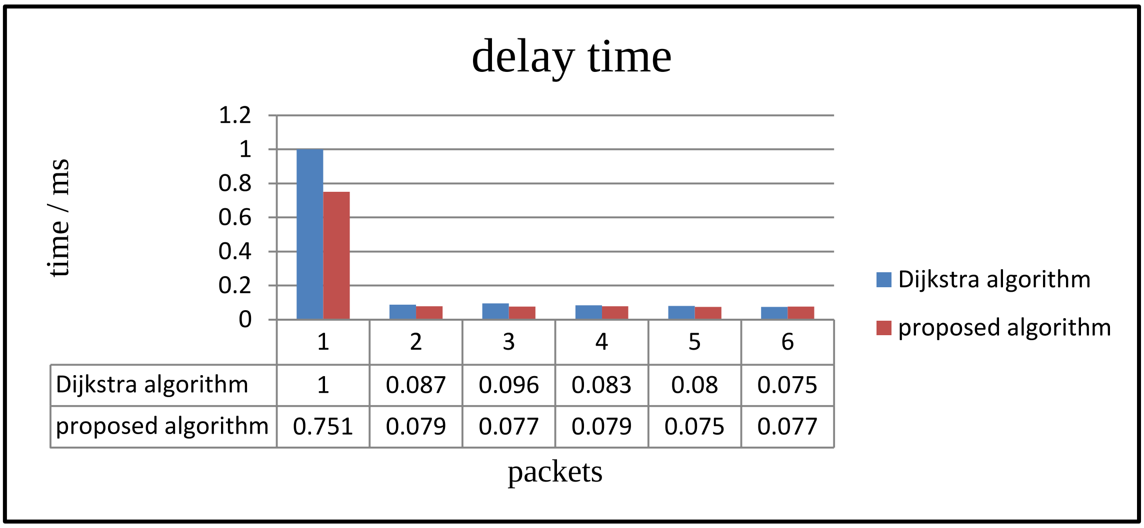
flow number	number of active switches		selected path	
	proposed algorithm	Dijkstra algorithm	proposed algorithm	Dijkstra algorithm
1	5	5	0, 9, 8, 7, 6	0, 9, 8, 7, 6
2	0	3	9, 8, 7, 6	9, 2, 3, 4, 6
total number active nodes			5	8

Table (4.8) shows the amount of energy consumption by using both algorithms prior to the active switches.

Table(4. 8): Amount of Energy Consumption

selected path		energy consumption	
proposed algorithm	Dijkstra algorithm	proposed	Dijkstra
0, 9, 8, 7, 6	0, 9, 8, 7, 6	610 w	610 w
9, 8, 7, 6	9, 2, 3, 4, 6	0 w	366w
total number active nodes		5	8

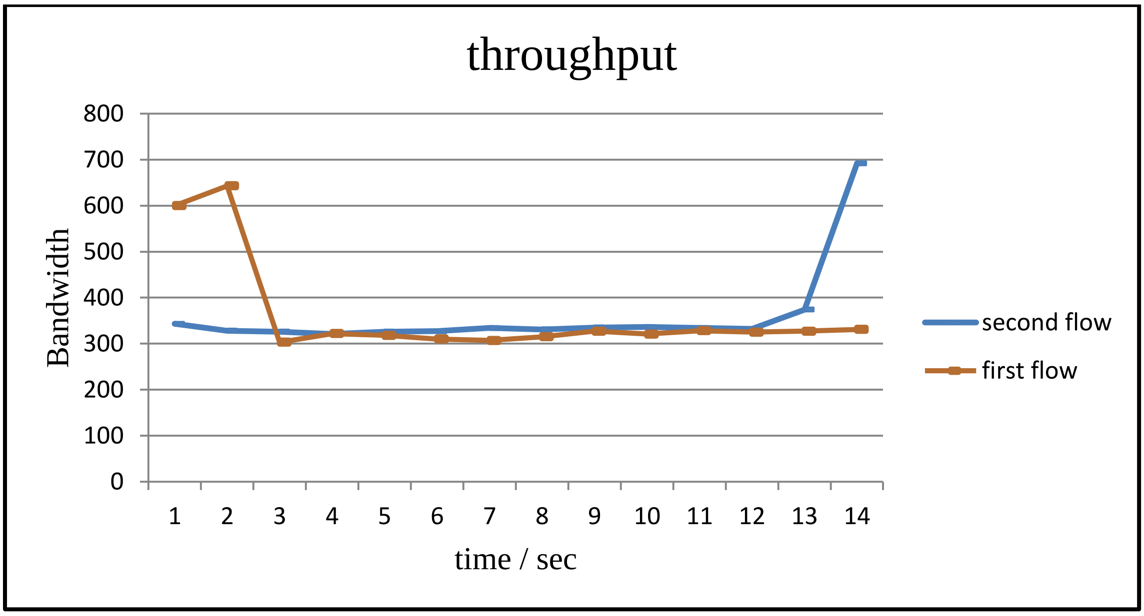
Figure(4.7) explain the delay time for transferring the packets between the source and destination hosts by using two algorithms it shown that the first packet has delay time greater than other because its consulate the controller after that the controller establish the path and all other packets pass through the path without consulting the controller.



	1	2	3	4	5	6
Dijkstra algorithm	1	0.087	0.096	0.083	0.08	0.075
proposed algorithm	0.751	0.079	0.077	0.079	0.075	0.077

Figure(4. 7): Delay Time

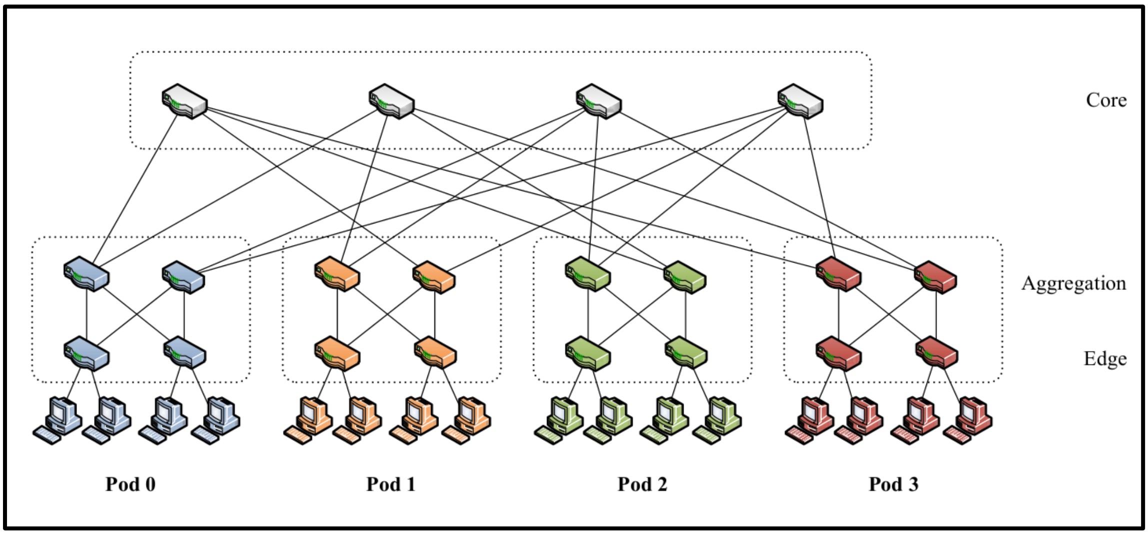
Figure (4.8) described tested network throughput. After forwarding two flows in the network by using iperf tools it is noted that when first flow is forwarded the amount of the throughput will be higher and after forwarding the second flow the throughput will decrease because the two flows transfer at the same path



Figure(4. 8): Throughput of the Network

4.2.3 Fat tree 4 Pod Datacenter Topology

This type of topology consist of three layer (edge, aggregation and core) where each pod connected to $(n/2)^2$ of servers and each aggregation switch connect to (k/2)core switch, and (k/2)edge switch. Each switch in edge level connected to (n/2) server and (n/2)switches in aggregation level as shown in the figure (4.9)



Figure(4. 9) : Fat tree 4 Pod

Python interpreter with special modules such as Fast Network Simulation Setup(FNSS) is used to create the tested topology, table (4.9) explain the parameters to this topology with 4 pods.

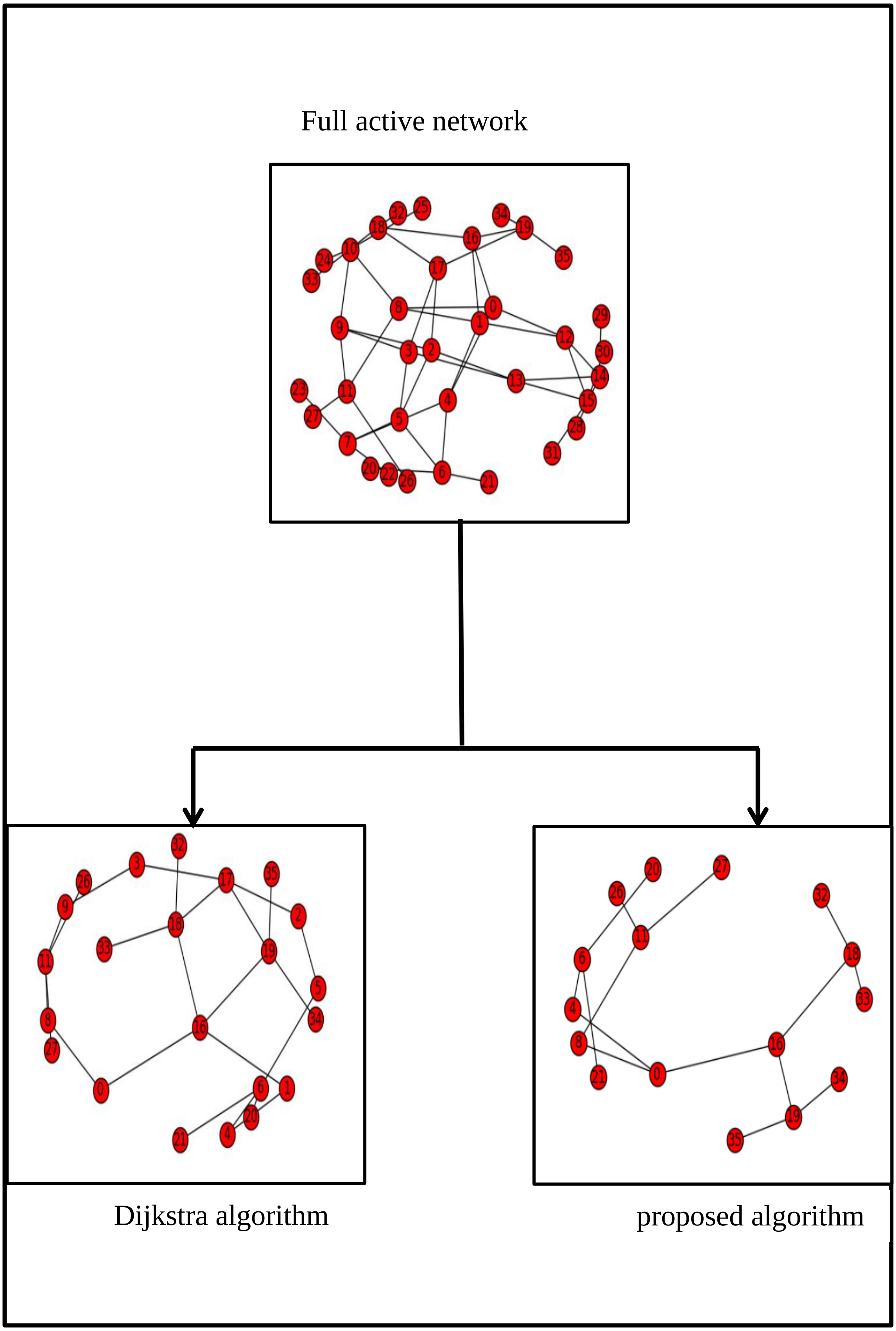
Table(4. 9): Fat tree Topology Parameters

Parameter	Value
no. nodes	36
no. switches	20
no. hosts	16
links capacity	10 Mbps
delay time	2 ms

Figure (4.10) demonstrates the number of nodes in three cases: full active network, after using proposed algorithm and after using Dijkstra algorithm. It is noted that the number of nodes in case of using proposed algorithm less than other cases because of exploiting the residual capacity of the links as shown in the table (4.10).

Table(4. 10): Number of Nodes

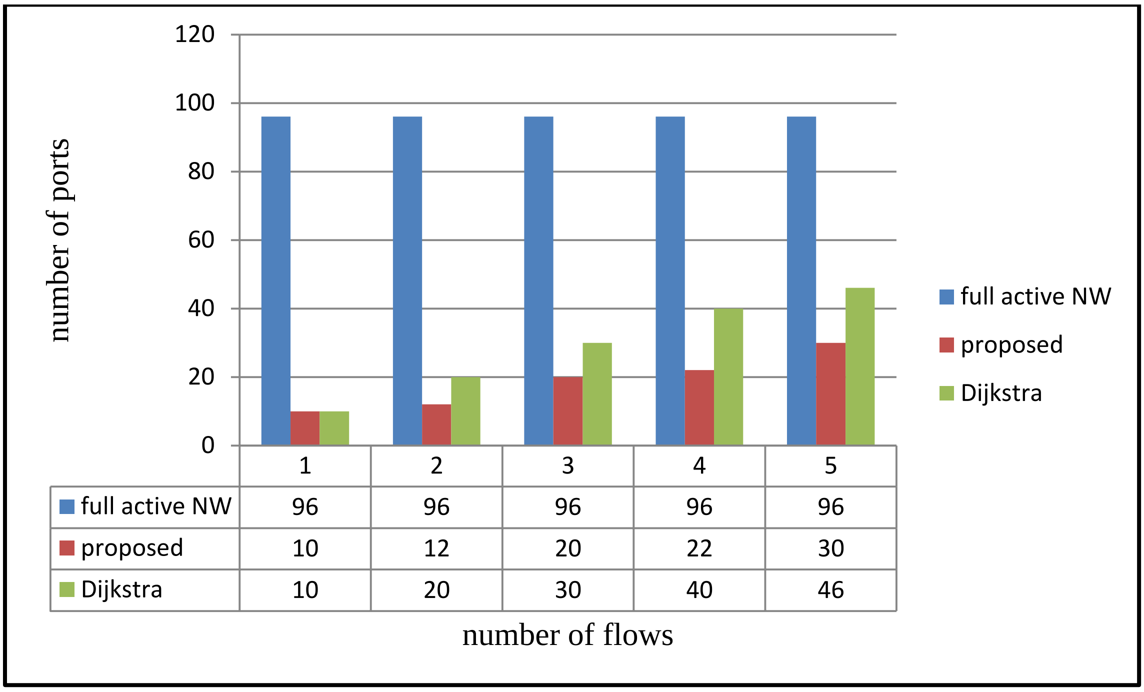
case	number of nodes	hosts	switches
full network	36	16	20
Dijkstra algorithm	27	10	17
proposed algorithm	21	10	11



Figure(4. 10):Number of Nodes

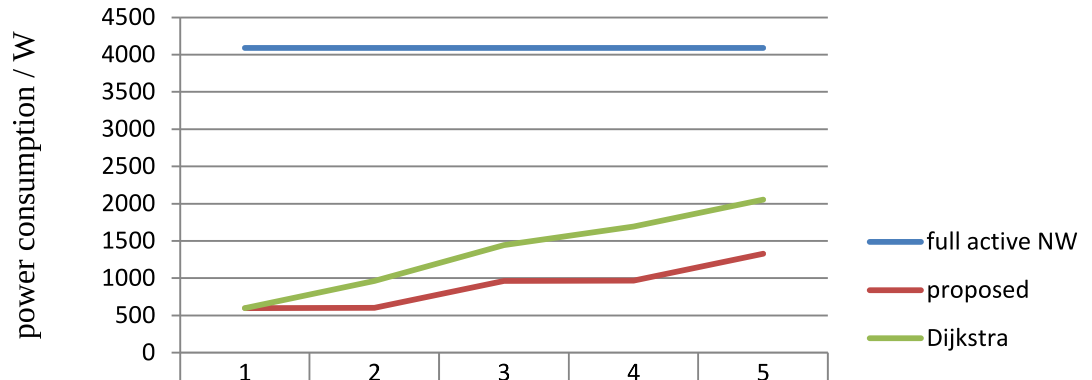
Below is set of figures show different comparison cases of energy consumption, using a number of activation ports and switches prior to the number of flows using proposed algorithm and Dijkstra algorithms .

Figure(4.11) explain the number of active ports in three cases and because proposed algorithm based on sharing the same link with more than one flow so the number of active links will be less than other cases. This resulting in saving the amount of energy as show in figure (4.12). Figure (4.13) demonstrate number of active switches in three cases .



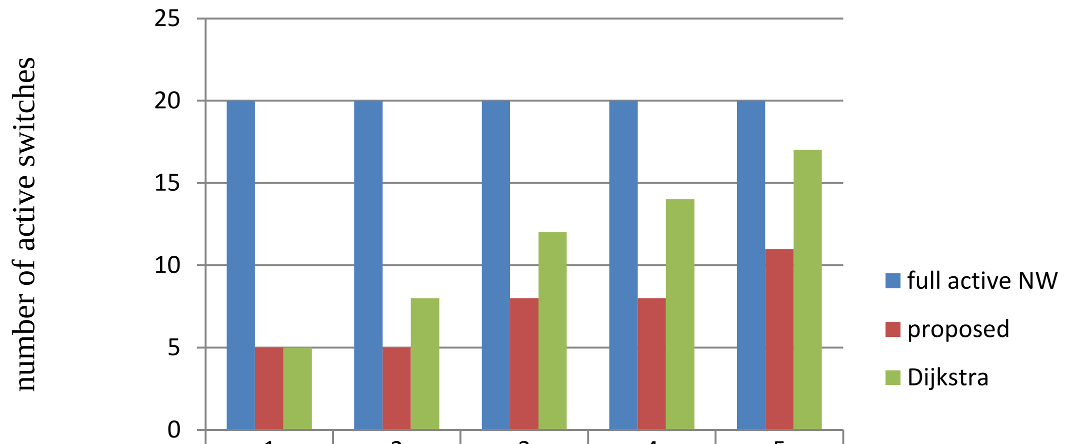
	1	2	3	4	5
full active NW	96	96	96	96	96
proposed	10	12	20	22	30
Dijkstra	10	20	30	40	46

Figure(4. 11): Number of Active Ports Prior to the Number of Flows



	1	2	3	4	5
full active NW	4092	4092	4092	4092	4092
proposed	600	602	964	966	1328
Dijkstra	600	964	1446	1692	2052

Figure(4. 12): Energy Consumption Prior to the Number of Flows



	1	2	3	4	5
full active NW	20	20	20	20	20
proposed	5	5	8	8	11
Dijkstra	5	8	12	14	17

Figure(4. 13): Number of Active Switches Prior to the Number of Flows

4.2.3.1 Network Performance

To clarifying the impact of proposed algorithm on the network performance that already mentioned in chapter two a set of tools for testing the delay time and throughput of the network are used to see the amount of network utilized before and after implementing the algorithm.

First of all, random hosts are selected with using ping tool for testing network delay time and as shown in the figure(4.14). The end to end delay is nearly equal to 8 ms because we use (2 ms) as delay time for each link and because the ICMP request takes 4 links between host1 to host2 .

```
mininet> h1 ping -c10 h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=114 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=8.99 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=8.89 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=8.90 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=9.47 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=9.44 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=9.81 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=8.97 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=8.87 ms
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=8.92 ms

--- 10.0.0.2 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9014ms
rtt min/avg/max/mdev = 8.875/19.701/114.712/31.671 ms
```

Figure(4. 14): Network Delay Time After Implementing Algorithm

The delay times for two hosts in the network before and after implementing the algorithm are nearly equal as shown in figure(4.15). This attributed to network topology nature(fat tree) which provides Multiple paths between any two hosts with the same number of links.

The minimum , maximum and average delay time according to the selected interval by using specific number of packets in the command could obtain.

```
mininet> h1 ping -c10 h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=58.7 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=9.86 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=9.81 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=9.42 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=9.83 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=9.34 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=9.32 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=9.39 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=9.91 ms
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=9.45 ms

--- 10.0.0.2 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9018ms
rtt min/avg/max/mdev = 9.324/14.512/58.762/14.751 ms
```

Figure(4. 15): Delay Time Before Implementing Algorithm

For testing the network throughput, iperf tool was used that already supported with mininet simulator, by choosing any two hosts let be host1 (h1) as a server and host2 (h2) as a client.

Figure (4.16)and figure (4.17) demonstrate the amount of difference of network throughput before and after using the proposed algorithm, It is noticed the amount of delivered traffic before implementing the algorithm (94.4 – 103 Mbits /sec) is greater than after applying algorithm(18.1- 19.8 Mbits/sec) because more traffic will transfer on the same links instead of transferring them on separate links.

```
mininet> iperf h1 h2
*** Iperf: testing TCP bandwidth between h1 and h2
*** Results: ['94.7 Mbits/sec', '103 Mbits/sec']
```

Figure(4. 16): Throughput Before Implementing Algorithm

```
mininet> iperf h1 h2
*** Iperf: testing TCP bandwidth between h1 and h2
*** Results: ['18.1 Mbits/sec', '19.8 Mbits/sec']
```

Figure(4. 17): Throughput After Implementing Algorithm

4.2.4 Two Tier Datacenter Topology

With two tier datacentre topology, the network consists of two level of switches (core and edge). Each core switch connect to all edge switches and the edge switches connect to the hosts. Table (4.11) contain the parameters to the 2 tier datacentre topology with 4 core switches, 8 edge switches and 4 hosts for each edge.

Table(4. 11): Two Tier Topology Parameters

Parameter	Value
no. nodes	44
no. switches	12
no. hosts	32
links capacity	10 Mbps
delay time	2 ms

After implementing the proposed algorithm the average energy consumption based on the values of table (4.12) is 64% comparison to the full active network values of table(4.13).

Table(4. 12): Energy Consumption Based on Proposed Algorithm

Proposed Algorithm		
Switches/ W	Ports / W	Total / W
354	6	360
354	8	362
590	14	604
590	16	606
826	22	848

Table(4. 13): Energy Consumption in Case of Full Active NW

full active NW		
switches	ports	total
1416	96	1512
1416	96	1512
1416	96	1512
1416	96	1512
1416	96	1512

The proposed algorithm has achieved the rate of energy reduction which is up to 7%, compared to the Dijkstra algorithm as shown in the table (4.14). Table(4.14) contains values indicating the amount of energy consumption in case of Dijkstra algorithm which based on number of hops for saving energy while proposed algorithm work on allowing more than on traffic to share the same link and switch off unused one.

Table(4. 14): Energy consumption based on Dijkstra algorithm

Dijkstra Algorithm		
Switches/ W	Ports/ W	Total/ W
354	6	360
472	12	484
708	18	726
708	24	732
944	30	974

4.2.4.1 Network Performance

For checking the delay time of the network two hosts are selected and test the end to end delay with minimum and maximum time by using ping command after and before implement the algorithm. 10 second the time that is used for computing the ping statistics, figure(4.18) shown that statistics before using the proposed algorithm and figure (4.19) shown that statistics after using it.

```
mininet> h1 ping -c 10 h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=82.7 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.266 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.065 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.064 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.060 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.069 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=0.065 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=0.053 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=0.075 ms
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=0.070 ms

--- 10.0.0.2 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 8999ms
rtt min/avg/max/mdev = 0.053/8.356/82.779/24.807 ms
```

Figure(4. 18): Ping Statistics Before Using Algorithm

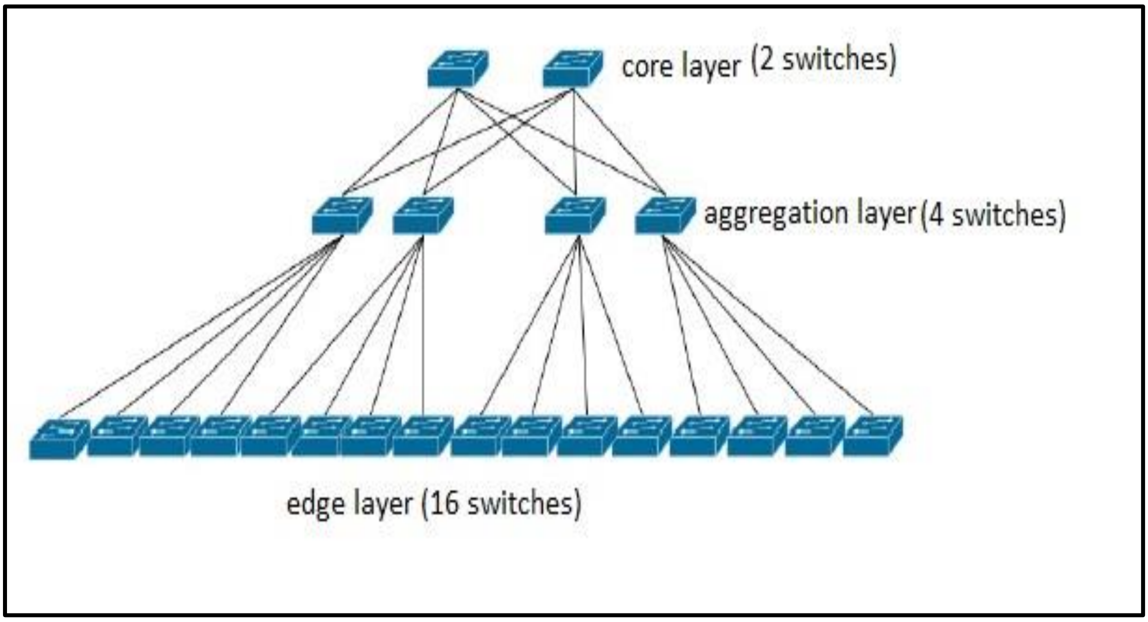
```
mininet>  h1 ping -c 10 h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=85.2 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.414 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.084 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.084 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.075 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.183 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=0.078 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=0.086 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=0.077 ms
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=0.081 ms

--- 10.0.0.2 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9004ms
rtt min/avg/max/mdev = 0.075/8.637/85.215/25.526 ms
```

Figure(4. 19): Ping Statistics After Using Algorithm

4.2.5 Three Tier Datacenter Topology

Three tier datacentre topology consist of three levels (core, aggregation, edge) where each switch in core and aggregation level will connect to gather. Each aggregation switch will connect to fixed number of switches in edge level as show in the figure(4.20) where each one of two switches in the core layer connect to four switches in aggregation layer. Each switch of them connect to four switches in the next level (edge level) .

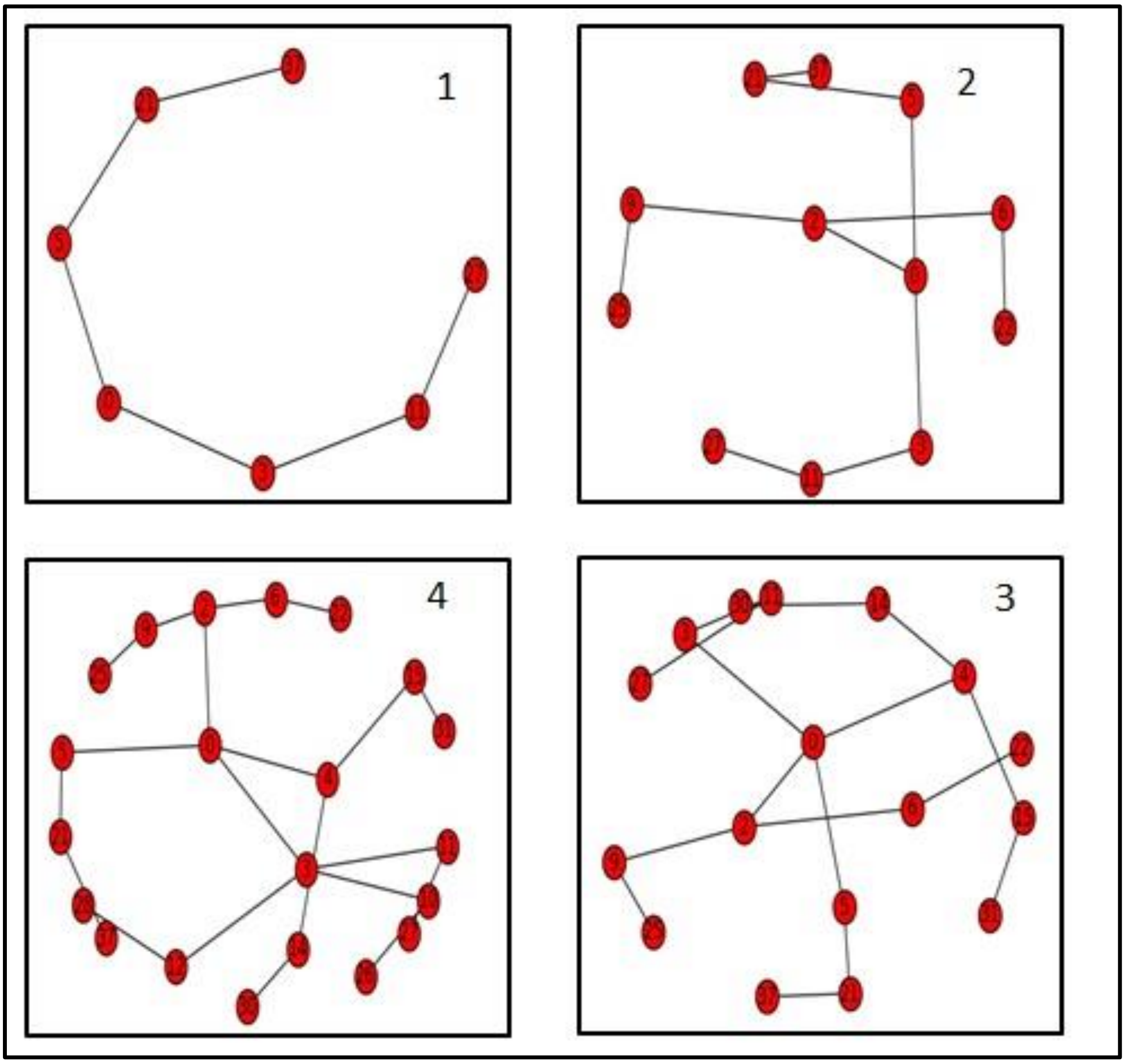


Figure(4. 20) : Three tier Datacenter Topology

Figure(2.21) represents building virtual topology phases which is , already mentioned in chapter three, step by step after each flow forwarding of the four flows in the table(4.15) and shows how to gradually build the topology.

Table(4. 15): Traffic Matrix

source node	destination node	traffic volume
37	27	100 k
25	22	100 k
31	30	100 k
28	26	100 k



Figure(4. 21): Building Virtual Topology Stages

Traffic matrix with five flows used to calculate the amount of energy saving which may reach 45% depending on the amount of difference in energy consumption when the proposed algorithm is used as shown in Table (4.16). Table(4.17) shows energy consumption when the network is fully operated. The number of active switches was calculated adding to the number of used ports in the case of the proposed algorithm only. The energy consumption of the virtual topology is obtained from the total consumption of the network devices (switches)

In the case of the full active network, the energy consuming of the devices are calculated for the network completely regardless of whether the network devices participate in the transport process or not.

Table(4. 16): Energy Consumption Based Proposed Algorithm

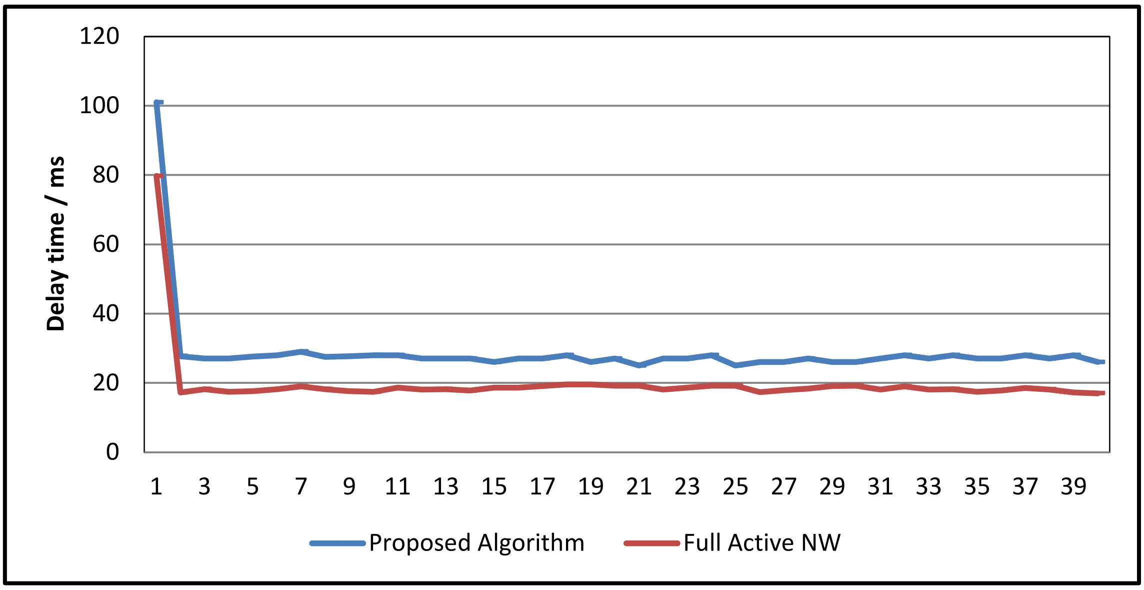
Flow number	Energy consumption in watt		
	Switches/ W	Ports/ W	Total/ W
1	590	10	600
2	944	16	960
3	1298	24	1322
4	1534	32	1566
5	1770	36	1806

Table(4. 17):Energy Consuming for Full Active NW

Flow number	Energy consumption in watt		
	Switches/ W	Ports/ W	Total/ W
1	2596	64	2660
2	2596	64	2660
3	2596	64	2660
4	2596	64	2660
5	2596	64	2660

4.2.5.1 Delay Time

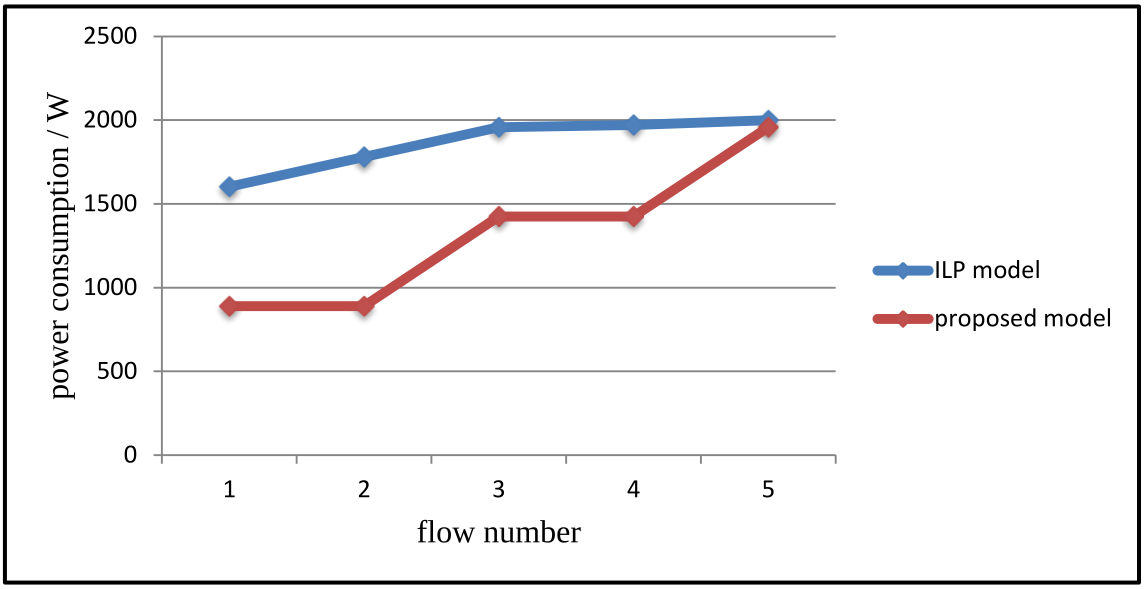
Figure(4.22) demonstrates the delay time between two hosts, h1 and h2 in two cases full active network and after forwarding the traffic using proposed algorithm. The delay time in the second case will be greater from the first one because of the proposed algorithm used off -line technique that mentioned in chapter three so it will select the links that already visited although it has a longer path instead of selected short path. This will effect the delay time, while increasing the links utilization of the network since the link will carry more traffics.



Figure(4. 22): Delay Time

4.3 Results Comparison With (ILP) Model[12]

In addition to comparing the energy consumption in obtained results which are relative to the Dijkstra algorithm, the results of the proposed model are compared with the results of model that already mentioned in the literature survey in chapter one. Where both model exanimate on the fat tree data centre topology with four pod and one by one traffic matrix (static traffic matrix). In addition to the power model in consideration as (ILP) model. Both models differ in used power management methodology to reduce energy consumption that already mentioned in chapter two, where the proposed model based on low power idle technique instead off sleeping technique that is used in (LIP) model. The result of comparison demonstrates in figure (4.23).



Figure(4. 23):Power Consumption ILP vs. Proposed Model

4.4 Results Discussion

This section discuss the analyzed results obtained from evaluation performance test.

4.3.1 Energy Saving

Proposed algorithm reduces the energy consumption by depending on the dynamic modification approaches in general. In particular the archived reduction in energy consumption is attributed low power ideal (LPI), minimizing number of active network element (switches) and number of active ports per switch. Redirecting the traffic to the path that already used and allowing more than one traffic to share the same link while the constraint of the capacity not violate represent the objective function of this model.

Figure (4.4) represents simple network model for the first flow of the two algorithms (proposed and Dijkstra) shows consumption of the same amount of energy because both of them use the same path but with the second flow the number of active switches based on Dijkstra algorithm is 6 by adding one additional switch as shown in the table(4.4).

Also in the Abilene network in the table(4.7) notes that the second flow doesn't consume any additional energy because the path already activated from previous flow by using proposed algorithm. On the other hand, if we used Dijkstra algorithm we need to active 3 new nodes for forwarding the second flow.

Figure(4.14) declares the result of energy consuming based on proposed algorithm For the fat tree datacentre topology. Two tier datacentre topology with energy consuming is explained in the table (4.12). Three-tier datacentre topology with energy consumption described in the table(4.16). All of them achieved amount of energy saving more than other cases (full active network and using Dijkstra algorithm)that demonstrate in the table(4.13), table (4.14) and table(4.17) prior to the same reason which is sharing the same link with more than one flow so the number of active links will be less.

4.3.2 Performance Evaluation Testing

Finding another path by applying proposed algorithm from the path founded by shortest path algorithm which is based on a number of hops may impact on the network performance in general . In particular the delay time and throughput.

Figure (4.8) which demonstrate the throughput of the network after forwarding two flows and because they share the same path, shows throughput decreasing from the iperf server perspective after forwarding the second traffic.

For fat tree datacenter topology the delay time test result shown in the figure (4.14) for proposed algorithm and figure(4.15) for Dijkstra algorithm. Nearly both results are equal because the topology nature which provides multiple paths to the same traffic.

The same results obtained from testing of two-tier datacenter topology delay time in the figure(4.18) and figure(4.19). The delay time of the three tier datacenter topology shown in the figure(4.22), which is greater in case of using proposed algorithm. This is because the algorithm work on the principle of off-line technique instead of on-line one that mentioned in chapter three.

Chapter 5

Conclusion and Suggestion for Future Works

Chapter Five
Conclusion and Suggestion for Future Works

5.1 Conclusion

The main purpose of this thesis is produce a model for reducing the energy consumption in computer networks which depends on its architecture and design using software defined network (SDN). Depending on this model the following conclusions can be drawn.

1- The proposed model was found to be efficient in reducing energy consumption due to the used method (dynamic modification) .

2- The SDN architecture and design had the greatest role to accomplish the thesis goal because of the general view of the controller and the ease of programming the network.

3- Re-forwarding, the traffic to the already used links instead of using a new one helps in increasing links utilizations of the network in general.

4- Proposed model most appropriate for a datacenter topology rather than other backbone topology due to the delay time of delivery the traffic because the number of links between each two host in the network is almost equal.

5- Reducing energy consumption is affected by the detail level of the network devices (switches). More accurate details lead to better results.

5.2 Suggestions for Future Works

Some of the suggestions listed below For future works guidance:

1. Testing the model with different controller such as floodlight or open daylight which are java based .
2. Developing this model by using different methods for energy saving in Software Wireless Network (SWN).
3. Develop the model to operate on dynamic traffic matrix instead of static one that is used in testing this model.
4. Expand the tested network topology to include more switches and hosts and gain more results .

Refrences

[1] B. E. Smith, *" Green computing Tools and Techniques for Saving Energy, Money, and Resources"*. 6000 Broken Sound Parkway NW, Suite 300 Boca Raton: Taylor & Francis Group, 2014.

[2] S. L. Ward Van Heddeghem, Bart Lannoo, Didier Colle, Mario Pickavet, Piet Demeester, ""Trends in worldwide ICT electricity consumption from 2007 to 2012"," *Computer Communications,* 2014.

[3] G. K. Brian Underdahl *Software Defined Networking For Dummies*, 2015.

[4] D. B. Rawat and S. R. Reddy, "Software Defined Networking Architecture, Security and Energy Efficiency: A Survey," *IEEE Communications Surveys & Tutorials,* vol. 19, pp. 325-346, 2017.

[5] W. M. Wenjuan Li, Lam For Kwok, ""A Survey on OpenFlow-based Software Defined Networks: Security Challenges and Countermeasures"," 2016.

[6] S. B. Yadi Ma, ""Smart Pre-Classifier to Reduce Power Consumption of TCAMs for Multi-dimensional Packet Classification"," *SIGCOMM'12, August 13–17, 2012, Helsinki, Finland.,* 2012.

[7] K. Kannan and S. Banerjee, ""Compact TCAM: Flow Entry Compaction in TCAM for Power Aware SDN"," 2014.

[8] J. M. Fr´ed´eric Giroire, T. Khoa Phan, ""Optimizing Rule Placement in Software-Defined Networks for Energy-aware Routing"," 2014.

[9] R. CÂRPA, O. GLÜCK, and L. LEFEVRE, "Segment routing based traffic engineering for energy efficient backbone networks," in *2014 IEEE International Conference on Advanced Networks and Telecommuncations Systems (ANTS)*, 2014, pp. 1-6.

[10] N. Tran Manh, T. Nguyen Huu, T. Ngo Quynh, H. Hoang Trung, and S. Covaci, "Energy-aware routing based on power profile of devices in data center networks using SDN," in *2015 12th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, 2015, pp. 1-6.

[11] G. Y. Deze Zeng, Lin Gu, Song Guo, and Hong Yao, ""Joint Optimization on Switch Activation and Flow Routing towards Energy Efficient Software Defined Data Center Networks"," *IEEE ICC*, 2016.

[12] F. A. Moghaddam and P. Grosso, "Linear programming approaches for power savings in software-defined networks," in *2016 IEEE NetSoft Conference and Workshops (NetSoft)*, 2016, pp. 83-87.

[13] M. Rahnamay-Naeini, S. S. Baidya, E. Siavashi, and N. Ghani, "A traffic and resource-aware energy-saving mechanism in software defined networks," in *2016 International Conference on Computing, Networking and Communications (ICNC)*, 2016, pp. 1-5.

[14] Y. Hu, T. Luo, N. Beaulieu, and C. Deng, "The Energy-Aware Controller Placement Problem in Software Defined Networks," *IEEE Communications Letters*, vol. PP, pp. 1-1, 2016.

[15] M. J. F. Martín Casado, Justin Pettit,Jianying Luo, Natasha Gude, Nick McKeown,, ""Rethinking Enterprise Network Control"," 2009.

[16] T. D. Nadeau and K. Gray, *SDN: Software Defined Networks*, 2013.

[17] T. A. Nick McKeown , Guru Parulkar, ""OpenFlow: Enabling Innovation in Campus Networks"," "March 14, 2008".

[18] Y. W. Wenfeng Xia, Chuan Heng Foh,Dusit Niyato,Haiyong Xie, "A Survey on Software-Defined Networking," *IEEE COMMUNICATION SURVEYS & TUTORIALS,* vol. 17, 2015.

[19] Q. H. Fei Hu, and Ke Bao, "A Survey on Software-Defined Network and OpenFlow: From Concept to Implementation," *IEEE COMMUNICATION SURVEYS & TUTORIALS,* vol. 6, 2014.

[20] M. D. D. F. Celio Trois, Luis C. E. de Bona,Magnos Martinello, ""A Survey on SDN Programming Languages:Towards a Taxonomy"," *IEEE Communications Surveys & Tutorials,* 2015.

[21] S. Azodolmolky, ""Software Defined Networking with OpenFlow"," 2013.

[22] F. M. V. R. By Diego Kreutz, Paulo Esteves Verı´ssimo, Christian Esteve Rothenberg,Siamak Azodolmolky, and Steve Uhlig, "Software-Defined Networking:A Comprehensive Survey," *IEEE,* vol. 103, 2015.

[23] X. Sun, T. S. E. Ng, and G. Wang, "Software-Defined Flow Table Pipeline," in *2015 IEEE International Conference on Cloud Engineering*, 2015, pp. 335-340.

[24] K. L. Heng Qi *"Software Defined Networking Applications in Distributed Datacenters"*, 2016.

[25] C. B. Paul Göransson *"Software Defined Networks A Comprehensive Approach"*, 2014.

[26] T. S. Manar Jammala, Abdallah Shamia, RasoolAsalb, and Yiming Lic, "Software-Defined Networking: State of the Art and Research Challenges," *Elsevier's Journal of Computer Networks*.

[27] F. M. V. R. Diego Kreutz , Paulo Verissimo, "Towards Secure and Dependable Software-Defined Networks," *ACM 978-1-60558-595,* 2013.

[28] S. S.-H. Sakir Sezer, Pushpinder Kaur Chouhan, "Are We Ready for SDN? Implementation Challenges for Software-Defined Networks," *IEEE Communications Magazine*, 2013.

[29] S. M. M. Matthew N.O. Sadiku *"Performance Analysis of Computer Networks"*, 2013.

[30] K. Kant, *"Introduction to Computer System Performance Evaluation"*.

[31] J. W. ZHAOGANG SHU, JIAXIANG LIN, SHIYONG WANG, DI LI, SEUNGMIN RHO, CHANGCAI YANG1, "Traffic Engineering in Software-Defined Networking: Measurement and Management," 2016.

[32] S. Khan, A. Gani, A. Abdul Wahab, M. Guizani, and M. K. Khan, "Topology Discovery in Software Defined Networks: Threats, Taxonomy, and State-of-the-art," *IEEE Communications Surveys & Tutorials,* pp. 1-1, 2016.

[33] B. S. D. Larry L. Peterson *Computer Networks a Systems Approach*, 3 ed., 2003.

[34] J. D. Sloan, *"Network Troubleshooting Tools"*, 2001.

[35] J. W. S. R. H. Zhang, *"Green Communications Theoretical Fundamentals, Algorithms and Applications"*: Taylor & Francis Group, 2013.

[36] S. M. Fabian Kaup, David Hausheer, ""Measuring and Modeling the Power Consumption of OpenFlow Switches"," 2014.

[37] B. G. A. a. O. Ozkasap, ""State-of-the-art Energy Efficiency Approaches in Software Defined Networking"," *The Fourteenth International Conference on Networks,* 2015.

[38] K. R. Deepankar Medhi, ""Network Routing Algorithms, Protocols, and Architectures"," 2007.

[39] M. D. Maurizio Palesi *"Routing Algorithms in Networks-on-Chip"*, 2014.

[40] C. E. L. Thomas H. Cormen , Ronald L. Rivest,Clifford Stein, *"Introduction to Algorithms"*, 3rd ed., 2009.

الخلاصة

بسبب أن الطاقة المستهلكة لتكنولوجيا المعلومات و الاتصالات قد تزايدت بسرعة في السنوات الاخيرة نسبة إلى التطورات الحاصلة في هذا المجال، اصبحت مشكلة تقليل استهلاك الطاقة موضوعا مهما بالنسبة لمعظم الباحثين في مجال الشبكات. في الشبكة التقليدية، من الصعب تحديد طريقة تعاونية للحد من استهلاك الطاقة لجميع أجهزة الشبكة لأن وظائف التحكم لاجهزة الشبكة تكون موزعة عليها، من جهة أخرى، معمارية شبكات جديدة وجدة سميت ب (software defined network) مع ميزات جديدة مثل وحدة تحكم مركزية.

وتهدف هذه الأطروحة إلى إنتاج نموذج لتقليل استهلاك الطاقة باستخدام طريقة التعديل الديناميكي .إعادة توجيه حركة البيانات من المسار الافتراض كأن يكون أقصر مسار إلى مسار آخر يخدم هدفنا وهو تقليل الاجهزة المتحكمة في سير البيانات وقد تم اختبار النموذج المقترح باستخدام أنواع مختلفة من الشبكات مثل شبكة العمود الفقري و شبكات datacenter topologies ومقارنتها بخوارزمية ديجكسترا.

ومن خلال النتائج التي تم الحصول عليها من عملية المحاكاة نجد أن النموذج يحقق توفير الطاقة بنسبة 64٪ تقريبا مقارنة ب اشتغال الشبكة بصورة كاملة و 19٪ في حالة استخدام وخوارزمية ديكسترا لشبكة fat tree datacenter topology و 64٪ و 7٪ لشبكات two tier datacenter، و 54٪ لشبكات three tier datacenter

جمهورية العراق

وزارة التعليم العالي والبحث العلمي

الهيئة العراقية للحاسبات والمعلوماتية

معهد المعلوماتية للدراسات العليا

معمارية و تصميم الشبكة المعرفة بالبرامجيات اساس نموذج لحفظ الطاقة

رسالة مقدمة الى

معهد المعلوماتية للدراسات العليا في الهيئة العراقية للحاسبات

وهي جزء من متطلبات نيل شهادة ماجستير علوم

في علوم الحاسبات

اعداد الطالب

بسام نوري شاكر

بأشراف

أ.م.د. محمد نجم عبد الله

1438 2017