VISVESVARAYA TECHNOLOGICAL UNIVERSITY
BELGAVI-590018

Internship Report
On

"Software defined network Framework for data broadcasting with machine learning"

Carried out
At
LMT software solutions Pvt. Ltd.

Submitted by:
VINAY KUMAR P (1JS16CS425)

Under the Guidance of:
Mr. PRASAD M R
(Asst. Prof. Dept. of CSE, JSSATEB)
and
MR. SHASHI
(H R, LMT, Bangalore)

Department of Computer Science and Engineering
JSS ACADEMY OF TECHNICAL EDUCATION
Uttarahalli-Kengeri Main Road, Bengaluru – 560060

JSS ACADEMY OF TECHNICAL EDUCATION

Uttrhalli-Kengeri Main Road, Bengaluru – 560060

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the Internship work entitled **"*Software defined network Framework for data broadcasting with machine learning*"** has been successfully carried out by **VINAY KUMAR P (1JS16CS425)** a bonafide students of **JSS Academy of Technical Education** in partial fulfilment of the requirements for the award of degree in **Bachelor of Engineering in Computer Science and Engineering** of **Visvesvaraya Technological University, Belgavi** for the duration of one month from **7th January, 2019** to **6th February, 2019** during academic year 2018-2019. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The internship report has been approved as it satisfies the academic requirements in respect of internship work for the said degree.

Dr. PRASAD M R	**Dr. NAVEEN N C**
(Asst. Prof., Dept. of CSE)	**(H.O.D, Dept. of CSE)**

Examiners: **Signature with Date**

1: _____

2: _____

ACKNOWLEDGEMENT

It gives us immense pleasure to present before you our internship work titled **"Software defined network Framework for data broadcasting with machine learning".** The joy and satisfaction that accompany the successful completion of any task would be incomplete without the mention of those who made it possible. We are glad to express our gratitude towards our prestigious institution **JSS Academy of Technical Education, Bengaluru** for providing us with utmost knowledge, encouragement and the maximum facilities in undertaking this internship work.

First and foremost we would like to thank his Holiness **Jagadguru Sri Shivarathri Deshikendra Maha Swamiji** and **Dr. Mrityunjaya V Latte, principal, JSSATE, Bengaluru,** for proving an opportunity to present this Internship as a part of my curriculum in the partial fulfilment of the degree course.

We express our sincere gratitude for **Dr. Naveen N C, Professor & Head, Department of Computer Science and Engineering,** for his co-operation and encouragement at all moments of my approach.

We are sincerely grateful to our Internship Guide **Mr. Prasad M R, Assistant Professor, Dept. of Computer Science and Engineering, JSSATEB**, for his spirited guideline and advice in carrying our Internship work and his valuable suggestions and constant supervision has been very helpful.

I express our deepest gratitude and special thanks to **Mr. SHASHI KUMAR, Managing Director, LMT software solution Pvt. Ltd**, **Bengaluru,** for his constant support and guidance during the internship.

VINAY KUMAR P [1JS16CS425]

LMT SOFTWARE SOLUTIONS PVT.LTD.

#17/46,First floor,service road,Remco Layout,Vijayanagar,Bangaluru-560040

COMPANY PROFILE

With the active participation of its multi-disciplinary Assignment Execution Team, LMT (Logic Mind Technology) Software Solutions Pvt Ltd. has emerged as a leader in the ITES in India and has established itself in the field of software development, data processing, data conversion, digital printing, Digitization, System integration, smart card personalization, IT facility management and other IT enabled services. LMT Software Solutions Pvt Ltd., incorporated in 2011, is a professionally managed, rapidly growing, multifaceted Information technology company. The company is actively involved in developing automation and e-Governance solutions for Transport, Social Security, Citizen Identity, Education, Public Distribution System, Retail Management and a host of other application areas.

LMT Software Solutions Pvt Ltd. is a leading System integrator in India providing complete turnkey solutions on BOO & BOOT basis including facility management services, Smart Cards applications, Document Management System (DMS), Work Flow Management and Manpower Deployment.

LMT Software Solutions Pvt Ltd. has successfully completed many e-governance projects for the various departments of Govt of AP and has won accolades for its superior service delivery, timely execution of projects and the quality of the deliverables. LMT is being trusted by many clients who are looking for reliable and quality services for their business. LMT is currently operating and managing in Bangalore and giving services to e-commerce business services.

LMT Software Solutions Pvt Ltd. adopted project team and dedicated organization structure. In project based organization, the project manager's directors have a high level of power to oversee and control the project assets. The project manager in this structure has downright power over the project and can secure assets expected to fulfil project targets from inside then again outside the parent organization, subject just to the extension, quality, furthermore, budget constraints are identified in the project.

In the project based structure, staff is particularly relegated to the project and report specifically to the project manager. The project manager is in charge of the execution evaluation and vocation movement of all undertaking colleagues while on the project. This prompts expanded project faithfulness. Complete line power over undertaking endeavors bears the project manager solid undertaking controls and brought together lines of correspondence. This prompts quick response time and enhanced responsiveness. In addition, project work forces are held on a restrictive instead of shared or low maintenance premise. Project teams create an in number feeling of task recognizable proof and possession, with profound faithfulness efforts to the project and a decent comprehension of the way of project's exercises, mission, or objectives.

Contact Information

Company Name	:	LMT Software Solutions Pvt Ltd.
Development & Data Processing Centre	:	#17/46, 1st Floor, Service Road Remco Layout, Bangalore-40.
General Phone No	:	080-41155670
Company Email	:	info@lmtsoftware.com
Website	:	www.lmtsoftware.com
Contact Number	:	9036159759

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

Software denned networking empowers network operators with more edibility to program their Networks. With SDN, network management moves from codifying functionality in terms of low level device conjurations to building software that facilitates network management and debugging. By separating the complexity of state distribution from network specification, SDN provides new Ways to solve long-standing problems in networking routing, for instance while simultaneously allowing the use of security and dependability techniques, such as access control or multi-path. However, the security and dependability of the SDN itself is still an open issue. In this position Paper we argue for the need to build secure and dependable SDNs by design. As a rest step in this Direction we describe several threat vectors that may enable the exploit of SDN vulnerabilities. We then sketch the design of a secure and dependable SDN control platform as a materialization of the concept here advocated. We hope that this paper will trigger discussions in the SDN community around these issues and serve as a catalyzer to join ports from the networking and security & Dependability communities in the ultimate goal of building resilient control planes.

INTRODUCTION

Software-defined networking (SDN) is an architecture that aims to make networks agile and flexible. The goal of SDN is to improve network control by enabling enterprises and service providers to respond quickly to changing business requirements.

In a software-defined network, a network engineer or administrator can shape traffic from a centralized control console without having to touch individual switches in the network. The centralized SDN controllers directs the switches to deliver network services wherever they're needed, regardless of the specific connections between a server and devices.

This process is a move away from traditional network architecture, in which individual network devices make traffic decisions based on their configured routing tables.

ARCHIETECTURE

A typical representation of SDN architecture comprises three layers: the application layer, the control layer and the infrastructure layer.

The control layer represents the centralized SDN controller software that acts as the brain of the software-defined network. This controller resides on a server and manages policies and the flow of traffic throughout the network.

The infrastructure layer is made up of the physical switches in the network. The application layer, not surprisingly, contains the typical network applications or functions organizations use, which can include intrusion detection systems, load balancing or firewalls. Where a traditional network would use a specialized appliance, such as a firewall or load balancer, a software-defined network replaces the appliance with an application that uses the controller to manage data plane behavior.

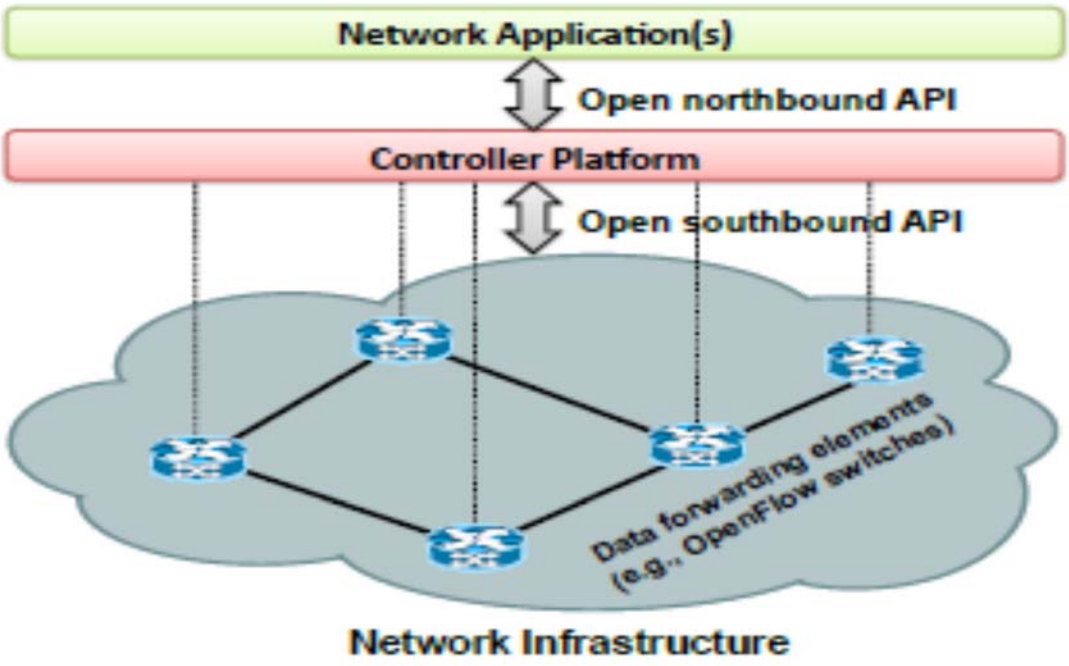


FIG 3.1: SDN architecture

SDN architecture separates the network into three distinguishable layers, connected through northbound and southbound APIs.

These three layers communicate using respective northbound and southbound application programming interfaces (APIs). For example, applications talk to the controller through its northbound interface, while the controller and switches communicate using southbound interfaces, such as OpenFlow -- although other protocols exist. There is currently no formal standard for the controller's northbound API to match OpenFlow as a general southbound interface. It is likely the OpenDaylight controller's northbound API may emerge as a de facto standard over time, given its broad vendor support.

WORKINGS

How SDN works?

SDN encompasses several types of technologies, including functional separation, network virtualization and automation through programmability. Originally, SDN technology focused solely on separation of the network control plane from the data plane. While the control plane makes decisions about how packets should flow through the network, the data plane actually moves packets from place to place.

In a classic SDN scenario, a packet arrives at a network switch, and rules built into the switch's proprietary firmware tell the switch where to forward the packet. These packet-handling rules are sent to the switch from the centralized controller.

The switch also known as a data plane device queries the controller for guidance as needed, and it provides the controller with information about traffic it handles. The switch sends every packet going to the same destination along the same path and treats all the packets the exact same way.

Software-defined networking uses an operation mode that is sometimes called adaptive or dynamic, in which a switch issues a route request to a controller for a packet that does not have a specific route. This process is separate from adaptive routing, which issues route requests through routers and algorithms based on the network topology, not through a controller.

The virtualization aspect of SDN comes into play through a virtual overlay, which is a logically separate network on top of the physical network. Users can implement end- to-end overlays to abstract the underlying network and segment network traffic. This micro segmentation is especially useful for service providers and operators with multitenant cloud environments and cloud services, as they can provision a separate virtual network with specific policies for each tenant.

BENEFITS OF SDN

With SDN, an administrator can change any network switch's rules when necessary prioritizing, deprioritizing or even blocking specific types of packets with a granular level of control and security. This is especially helpful in a cloud computing multi- tenant architecture, because it enables the administrator to manage traffic loads in a flexible and more efficient manner. Essentially, this enables the administrator to use less expensive commodity switches and have more control over network traffic flow than ever before.

Other benefits of SDN are network management and end-to-end visibility. A network administrator need only deal with one centralized controller to distribute policies to the connected switches, instead of configuring multiple individual devices. This capability is also a security advantage because the controller can monitor traffic and deploy security policies. If the controller deems traffic suspicious, for example, it can reroute or drop the packets.

SDN also virtualizes hardware and services that were previously carried out by dedicated hardware, resulting in the touted benefits of a reduced hardware footprint and lower operational costs.

Additionally, software-defined networking contributed to the emergence of software- defined wide area network (SD-WAN) technology. SD-WAN employs the virtual overlay aspect of SDN technology, abstracting an organization's connectivity links throughout its WAN and creating a virtual network that can use whichever connection the controller deems fit to send traffic.

CHALLENGES

Security is both a benefit and a concern with SDN technology. The centralized SDN controller Presents a single point of failure and, if targeted by an attacker, can prove detrimental to the network.

Ironically, another challenge with SDN is there's really no established definition of software-defined networking in the networking industry. Different vendors offer various approaches to SDN, ranging from hardware-centric models and virtualization platforms to hyper-converged networking designs and controller less methods.

Some networking initiatives are often mistaken for SDN, including white box networking, network disaggregation, network automation and programmable networking. While SDN can benefit and work with these technologies and processes, it remains a separate technology.

SDN technology emerged with a lot of hype around 2011, when it was introduced alongside the Open Flow protocol. Since then, adoption has been relatively slow, especially among enterprises that have smaller networks and fewer resources. Also, many enterprises cite the cost of SDN deployment to be a deterring factor.

Main adopters of SDN include service providers, network operators, telecoms and carriers, along with large companies, like Facebook and Google, all of which have the resources to tackle and contribute to an emerging technology.

USE CASE DIAGRAM

USE CASE 1:

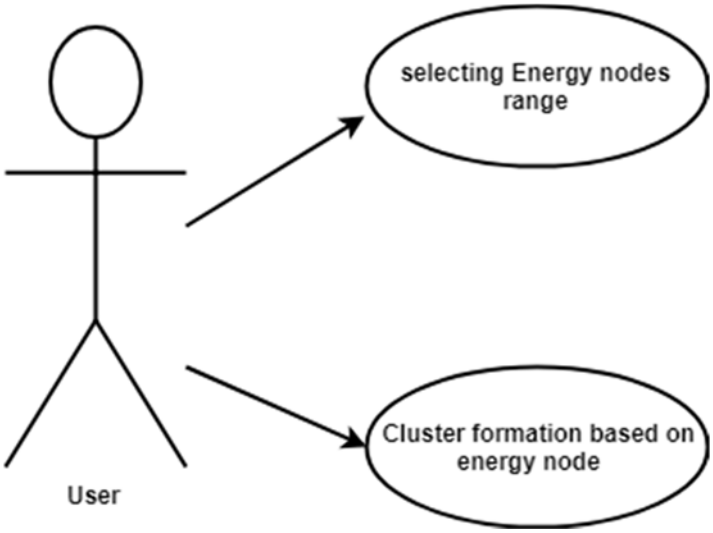


Fig: 8.1 use case 1

USE CASE 2:

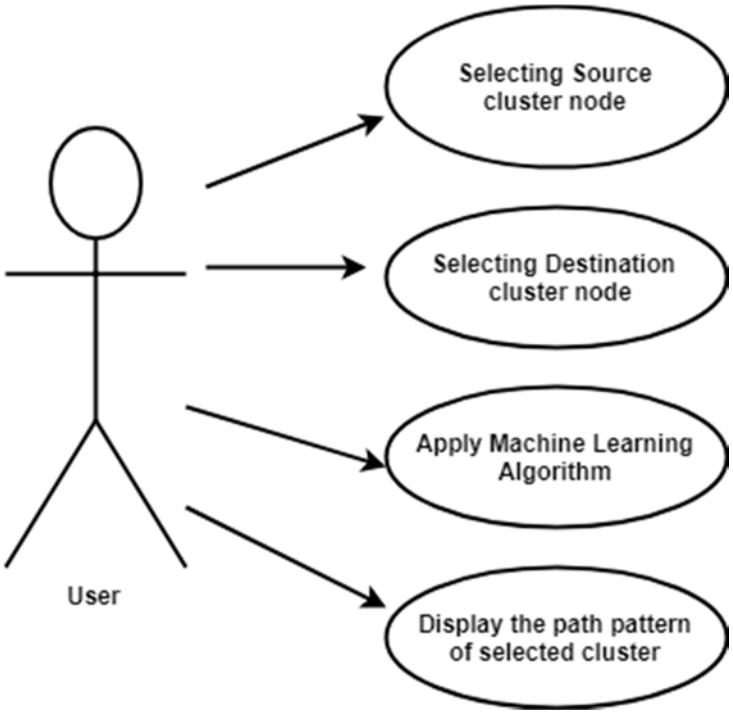


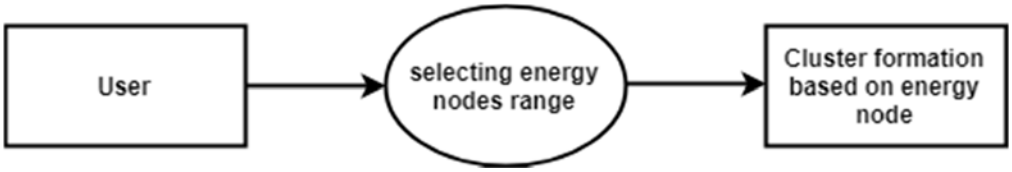
Fig: 8.2 use case 2

Data Flow Diagram

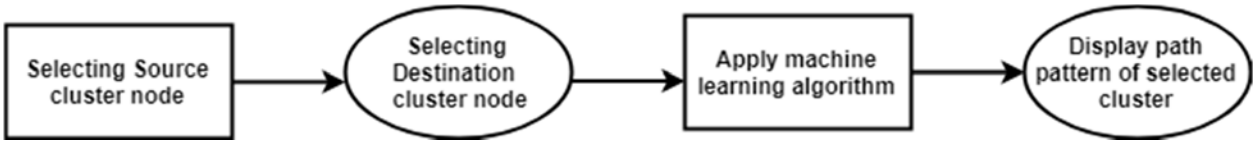
A data flow diagram is a graphical representation of the "flow" of data through an information system, modeling its *process* aspects. Often they are a preliminary step used to create an overview of the system which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design).

The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of the input data to the system, various processing carried out on these data, and the output data is generated by the system.

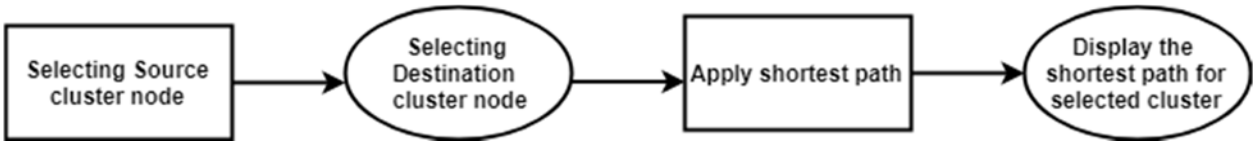
DFD 0



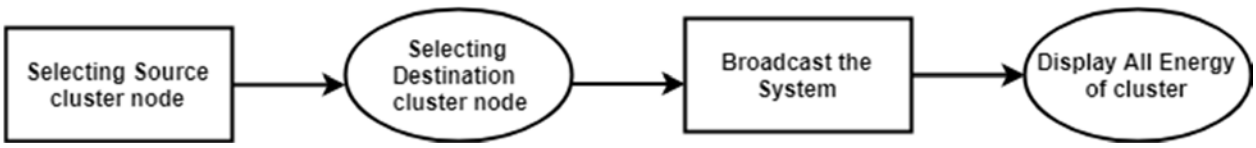
DFD 1



DFD 2

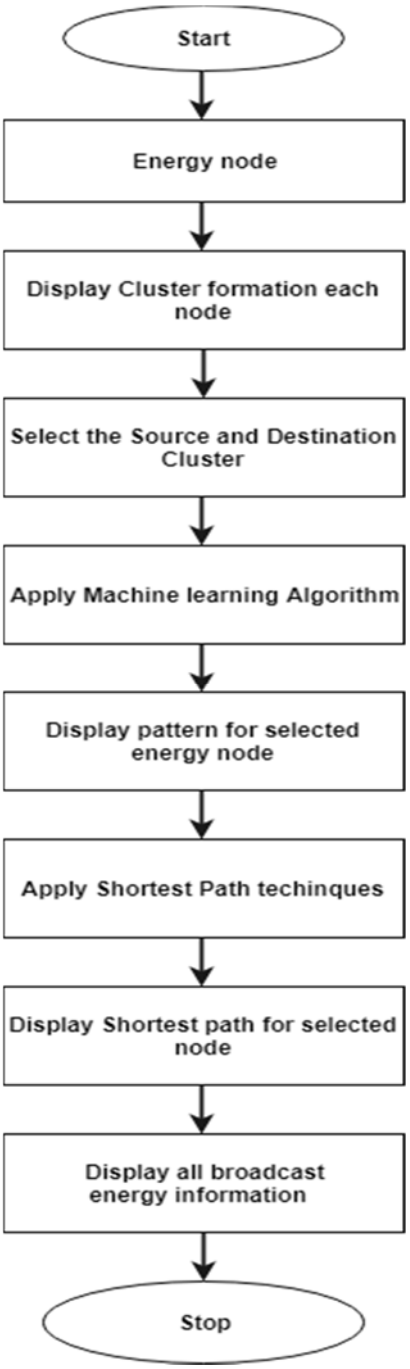


DFD 3



FLOWCHARTS

A flow chart is a graphical or symbolic representation of a process. Each step in the process is represented by a different symbol and contains a short description of the process step. The flow chart symbols are linked together with arrows showing the process flow direction.



SEQUENCE DIAGRAM:

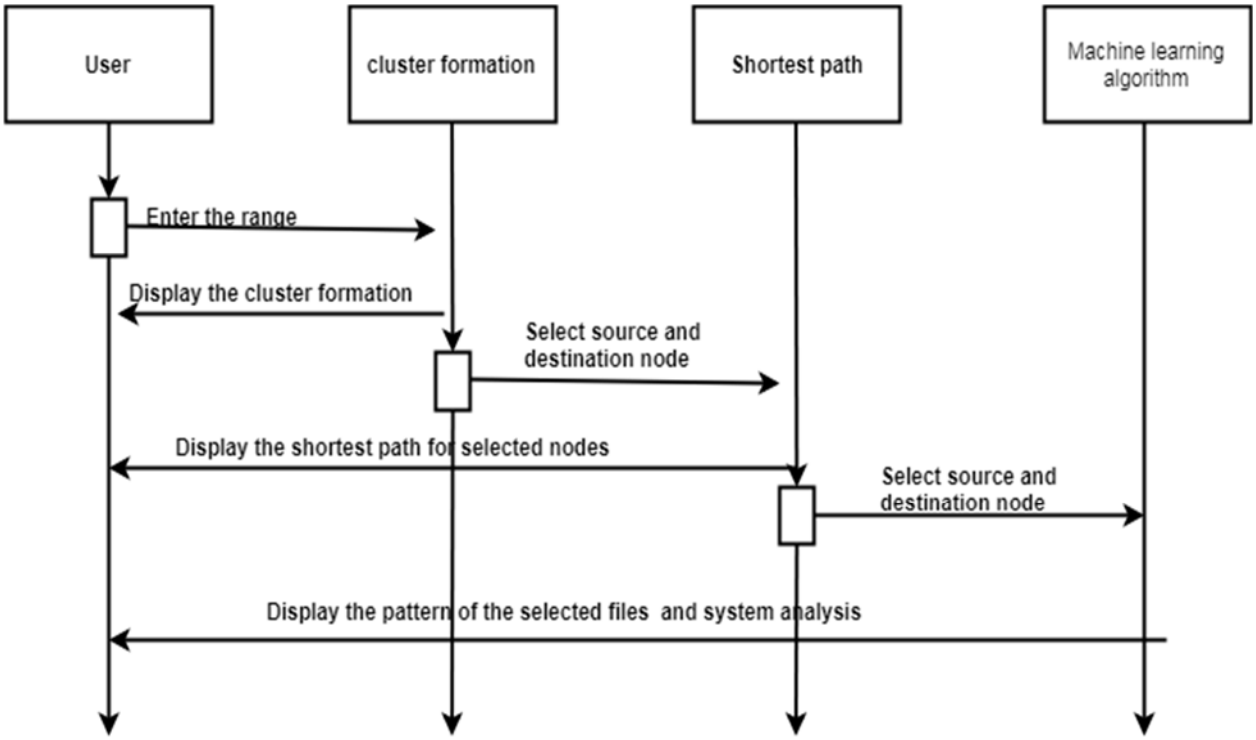


Fig: sequence

PSEUDO CODE

Random Uniform Matrix Generation

t1=MAKE(C1)

t2=MAKE(C2)

t3=MAKE(C3)

t4=MAKE(C4)

n1←RANDOM(1 - 150)

n2←RANDOM(1 - 150)

n3←RANDOM(1 - 150)

n4←RANDOM(1 - 150)

C1=ASSIGN ENERGIES(t1,n1)

C2=ASSIGN ENERGIES(t2,n2)

C3=ASSIGN ENERGIES(t3,n3)

C4=ASSIGN ENERGIES(t4,n4)

Ch1←MAXENERGY(C1)

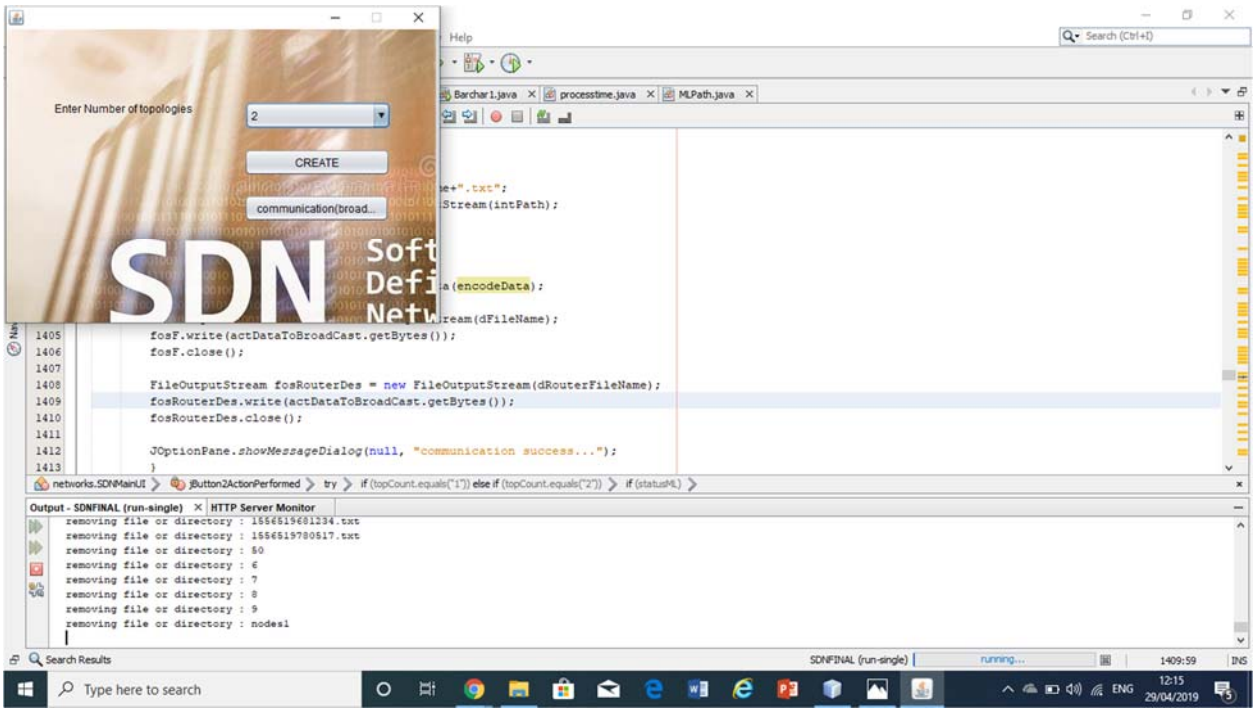
Ch2←MAXENERGY(C2)

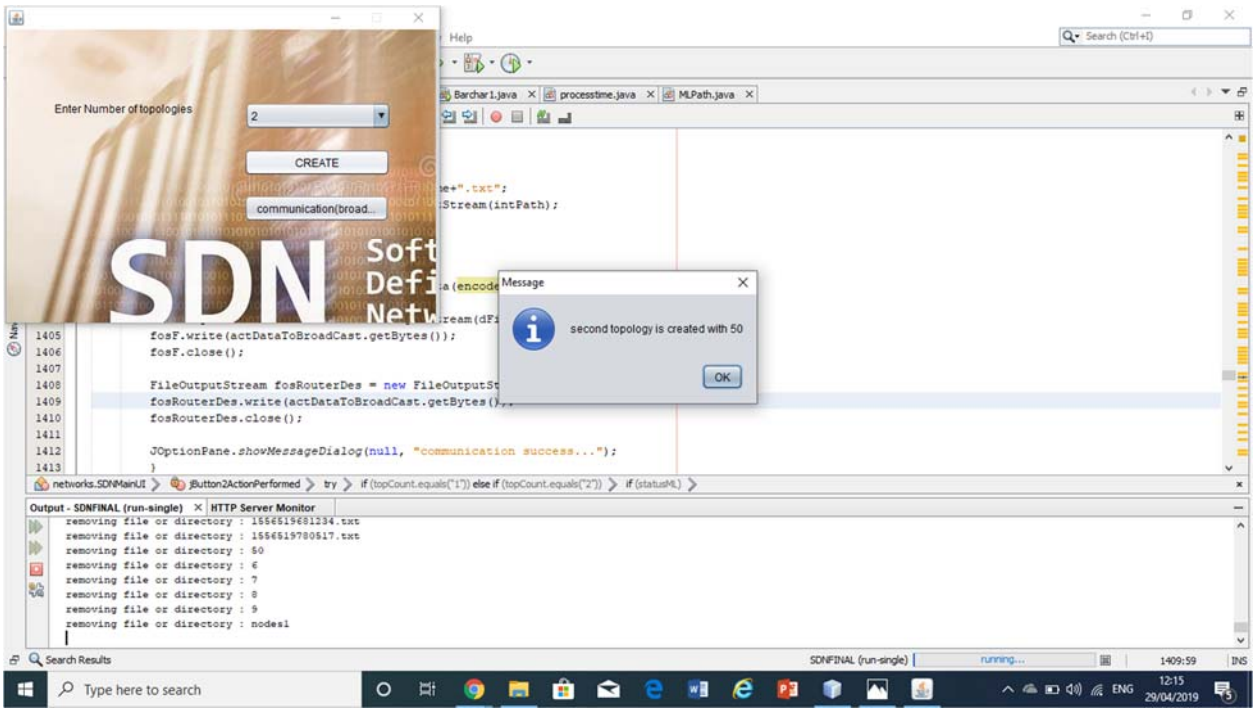
Ch3←MAXENERGY(C3)

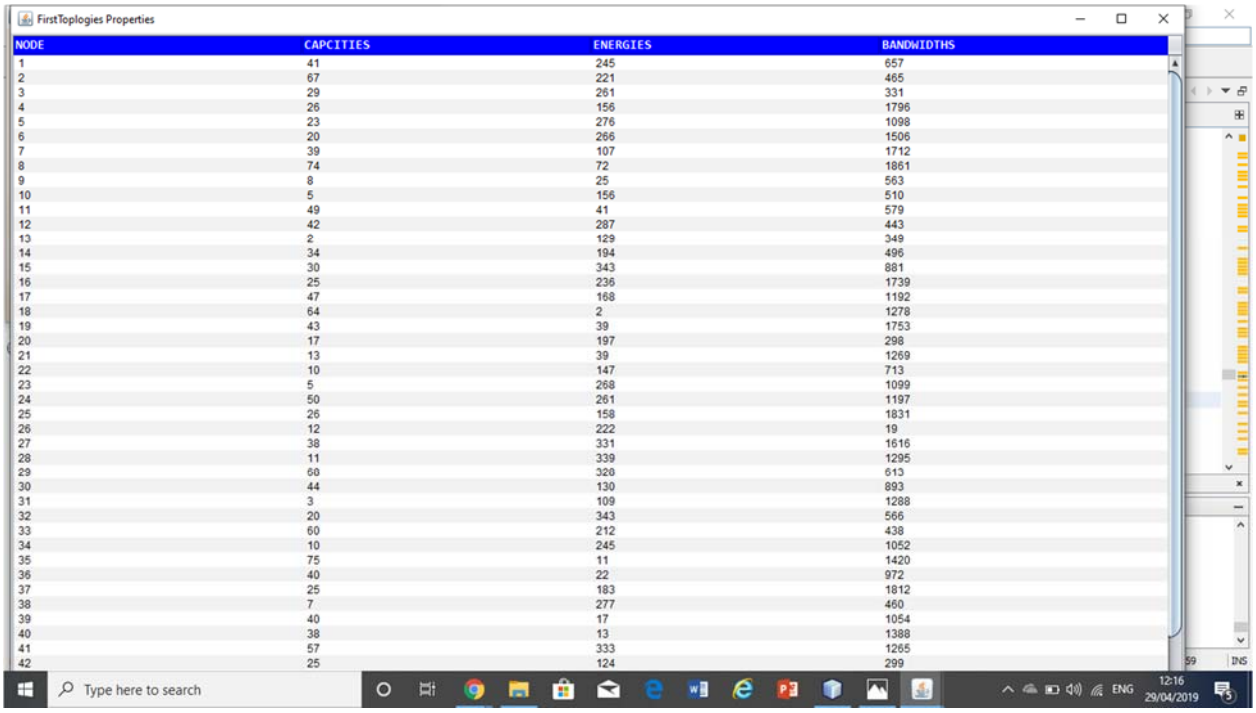
Ch4←MAXENERGY(C4)

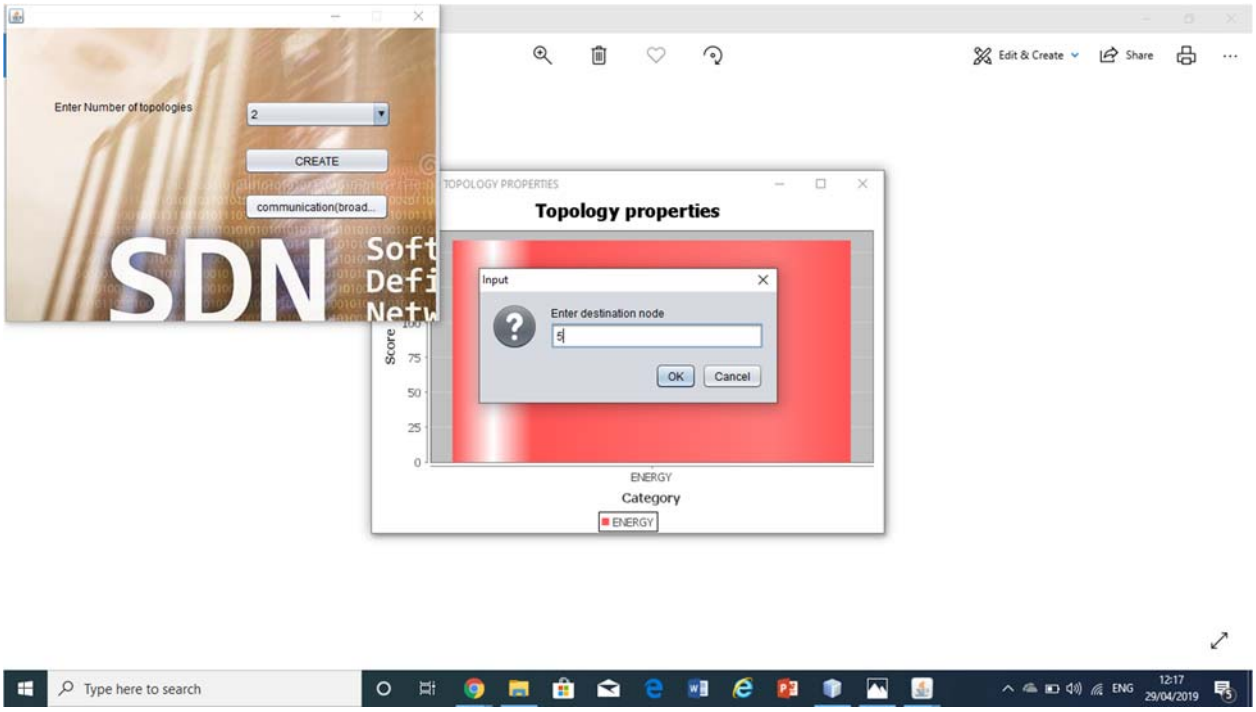
DISPLAYACK(C1,C2,C3,C4)

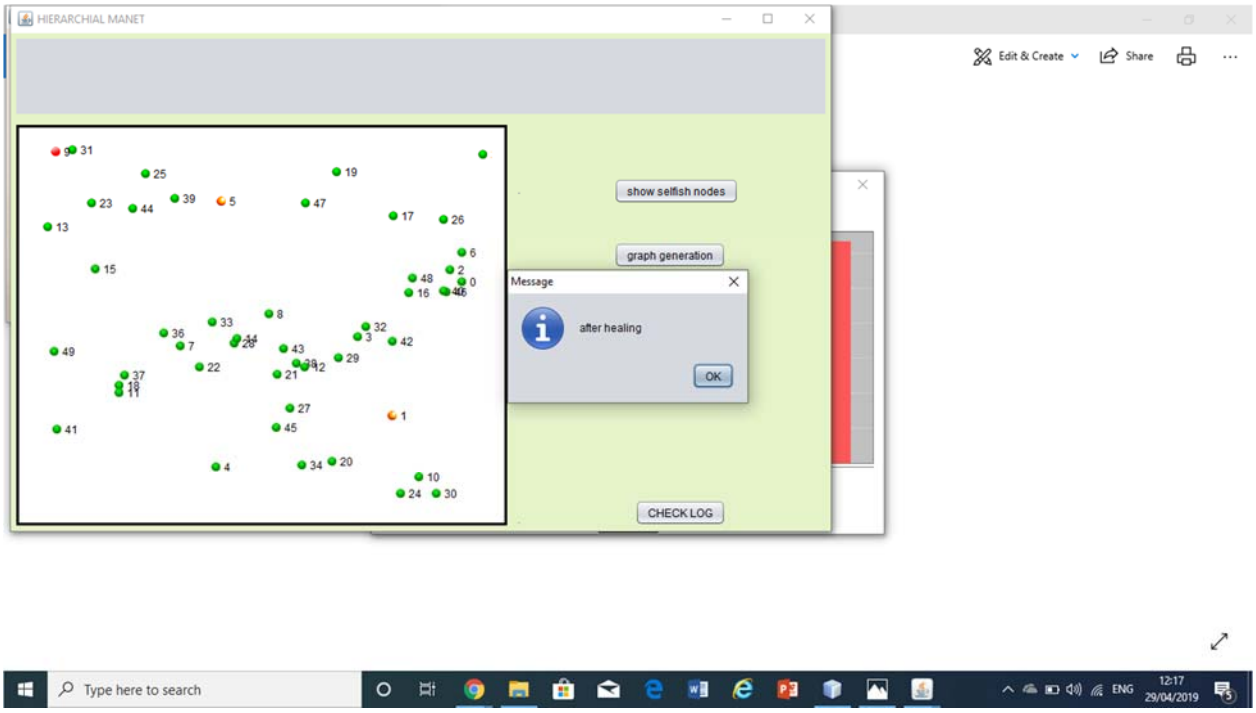
SCREENSHOTS

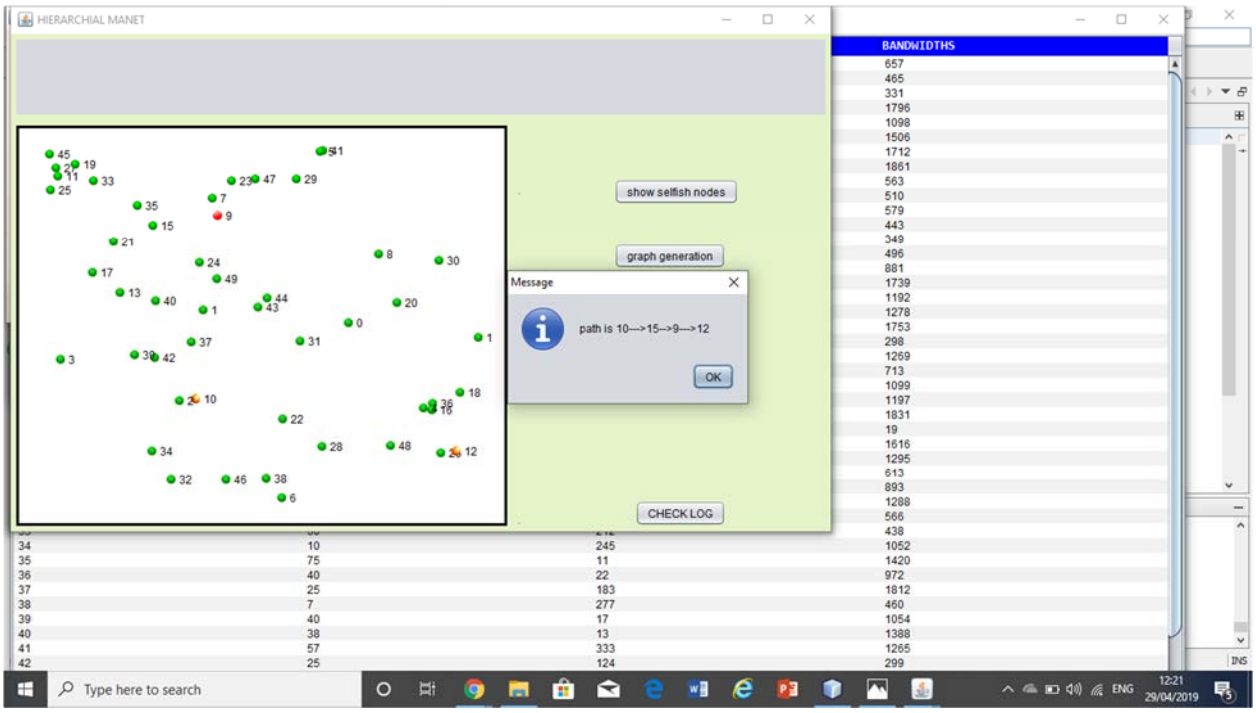


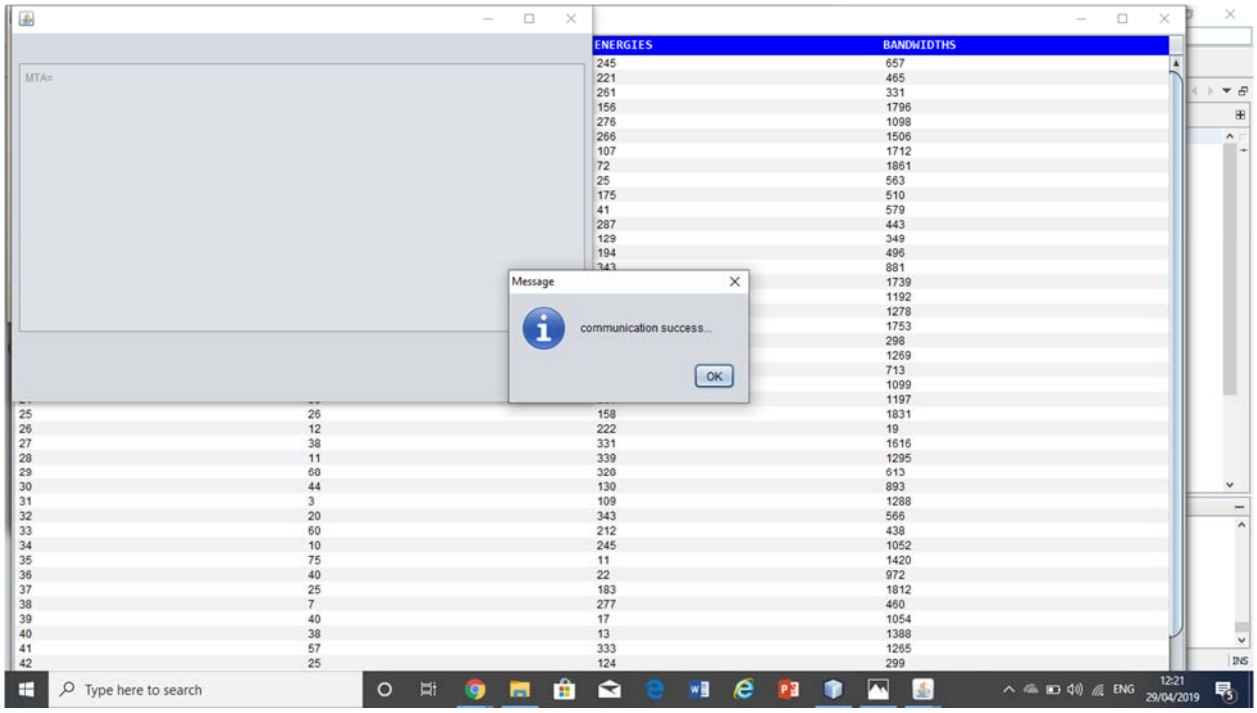


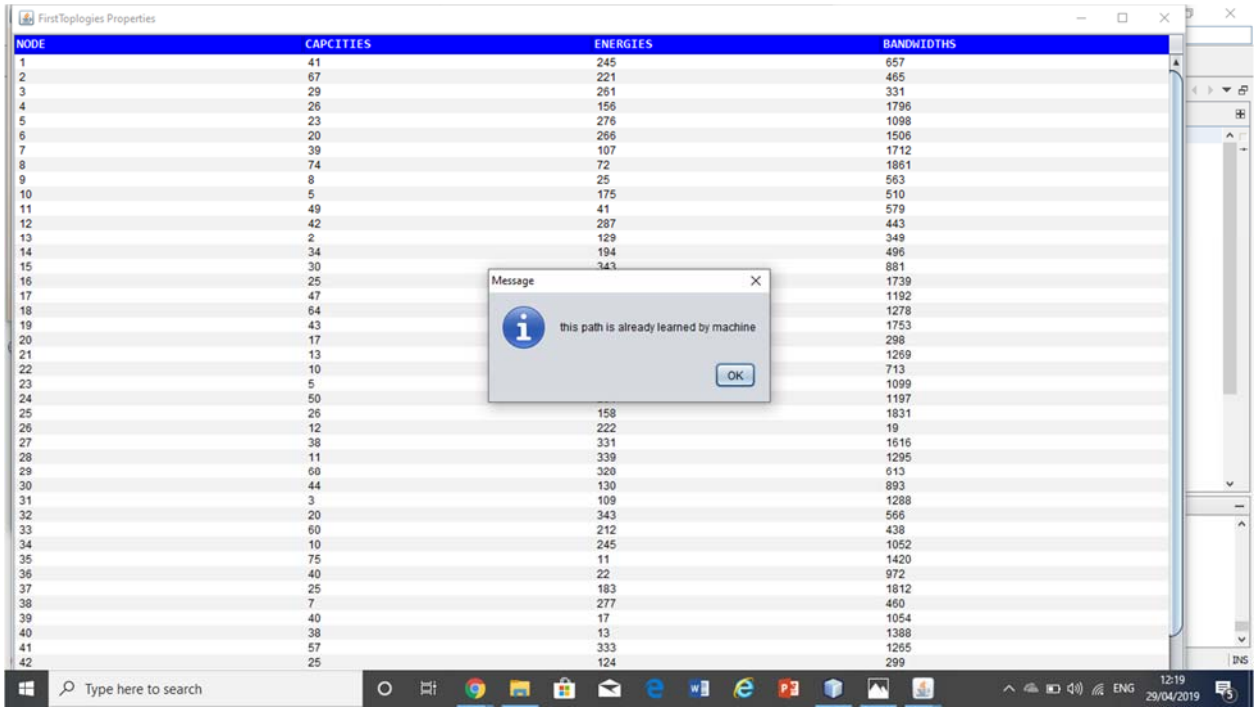

FirstToplogies Properties
NODE CAPCITIES ENERGIES BANDWIDTHS
1 41 245 657
2 67 221 465
3 29 261 331
4 26 156 1796
5 23 276 1098
6 20 266 1506
7 39 107 1712
8 74 72 1861
9 8 25 563
10 5 156 510
11 49 41 579
12 42 287 443
13 2 129 349
14 34 194 496
15 30 343 881
16 25 236 1739
17 47 168 1192
18 64 2 1278
19 43 39 1753
20 17 197 298
21 13 39 1269
22 10 147 713
23 5 268 1099
24 50 261 1197
25 26 158 1831
26 12 222 19
27 38 331 1616
28 11 339 1295
29 60 320 613
30 44 130 893
31 3 109 1288
32 20 343 566
33 60 212 438
34 10 245 1052
35 75 11 1420
36 40 22 972
37 25 183 1812
38 7 277 460
39 40 17 1054
40 38 13 1388
41 57 333 1265
42 25 124 299
Type here to search
ENG
12:16
29/04/2019


Enter Number of topologies
2
CREATE
communication(broad...
SDN
Soft
Defi
Netw
Edit & Create
Share
TOPOLOGY PROPERTIES
Topology properties
Input
Enter destination node
5
OK
Cancel
Score
100
75
50
25
0
ENERGY
Category
ENERGY
Type here to search
ENG
12:17
29/04/2019


HIERARCHIAL MANET
show selfish nodes
graph generation
Message
after healing
OK
CHECK LOG
Edit & Create
Share
Type here to search
ENG
12:17
29/04/2019


HIERARCHIAL MANET
show selfish nodes
graph generation
Message
path is 10--->15--->9--->12
OK
CHECK LOG
BANDWIDTHS
657
465
331
1796
1098
1506
1712
1861
563
510
579
443
349
496
881
1739
1192
1278
1753
298
1269
713
1099
1197
1831
19
1616
1295
613
893
1288
566
438
34 10 245 1052
35 75 11 1420
36 40 22 972
37 25 183 1812
38 7 277 460
39 40 17 1054
40 38 13 1388
41 57 333 1265
42 25 124 299
Type here to search
ENG
12:21
29/04/2019


MTA=
ENERGIES
BANDWIDTHS
Message
communication success...
OK
Type here to search
ENG
12:21
29/04/2019


FirstToplogies Properties
NODE
CAPCITIES
ENERGIES
BANDWIDTHS
Message
this path is already learned by machine
OK
Type here to search
ENG
12:19
29/04/2019

CONCLUSION

Based on the current existing scenario with the current work the conclusion is to make the edge node as cluster head for sure for broad casting. The proper broadcasting is getting done only with shortest path with which nodes are having more than aggregated threshold value. So to reduce duplicate broadcast this work is updated with machine learning model for updating of past paths.

REFERENCES

1. T. Koponen et al. Onix: a distributed control platform for large-scale production networks". In: OSDI. 2010.

2. N. Gude et al. NOX: towards an operating system for networks". In: Comp. Comm. Rev. (2008). M. Caesar et al. Design and implementation of a routing control platform". In: NSDI. 2005.

3. M. Casado et al. Rethinking Enterprise Network Control". In: IEEE/ACM Trans. on Networking 17.4 (2009).

4. P. Porras et al. A security enforcement kernel for OpenFlow networks". In: HotSDN. ACM, 2012. S. Shin et al. FRESCO: Modular Composable Security Services for Software-Dened Networks". In: Internet Society NDSS. 2013

5. N. McKeown et al. OpenFlow: enabling innovation in campus networks". In: Comput. Commun. Rev. (2008).

6. S. Sorensen. Security implications of software-dened networks. 2012. S. M. Kerner. Is SDN Secure? 2013.