IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 1, January 2017

Novel Approach Based on Genetic Algorithm for Adaptive Resource Management in Software-Defined Networks

Mohammed Najm Abdullah[1], Afrah Salman Dawood[1]

Department of Computer Engineering, University of Technology, Baghdad, Iraq[1]

Abstract: Resource management in Software-Defined Networking (SDN) is an important aspect in this brand new approach of networking. Implementing an efficient resource management with high Quality-of-Service (QoS) requirements like low delay or high bandwidth is a big challenge. Dynamic and adaptive solutions can be benefit for improving the performance of these networks. In this paper, we propose a novel approach of using genetic algorithm with SDN architecture and FNSS toolchain for better optimization of adaptive resource management with testbeds on datacenter topologies with video streams run on VLC media player to evaluate the performance of our solution. We explained the testbeds on the test video (delay and bandwidth) by implementing the adaptive solution using Genetic Algorithm (GA) for adaptively selecting the capacity according to link delay which then reduces the overall timeof packet delivery and increases the bandwidth. The final results shows an improvementin reducing the average delay values of 94.4% when using OVS controller and 90.1% when using Floodlight controller, and an improvement in bandwidth of 950 Mbps- 1.82 Mbps to 37.075 Mbps- 42.9 Mbps in OVS controller and 952 Mbps- 1.7 Mbps to 22.97 Mbps- 27.45 Mbps in Floodlight controller.

Keywords: SDN, Adaptive Resource Management, Fast Network Simulation Setup (FNSS), OVS Controller, Floodlight Controller, Mininet, Genetic Algorithm, Datacenter topology.

I. INTRODUCTION

Software-Defined Networking is the new concept of networking that separates the control plane from the data plane [1] to simplify network management and improve data transmission and overcome difficulties of traditional networks. The environment of these networks, known as Software Defined Environment (SDE), is responsible for optimizing Resource Management (RM) in network computing infrastructure [2]. SDN provides a great management, scalability, performance and other features [3]. Currently, resource management is based on parameters like link capacity, delay, bandwidth, etc.
Algorithms can be analyzed in different methods such as theaverage-case analysis for measuring the average performance on problems of size n. Adaptive analysis basically contrasts the performance of the algorithm to the difficulty of the related issue [9]. Genetic algorithm is one of the best adaptive algorithms which is a heuristic random search mechanism (i.e. evolutionary process) [10].These adaptive algorithms can be used in traditional networks and Software-Defined Networks for better resource utilization. The possibilities of using such adaptive algorithms in such modular networks has already been researched and discussed in [11].

In this paper, we propose a novel approach of optimizing resources in these networks based on adaptive algorithms (i.e. genetic algorithm) to enhance QoS requirements. The rest of the paper is arranged as follows: related work is described in Section II, Section III explains basic concepts and tools used in this work. In Section IV, we presents our proposal, and finally, Sections V and VI describes results and performance evaluation and conclusions, respectively.

II. RELATED WORK

Since resource management is an important concept in any system, SDN researchers are moving towards optimizing and improving these resources. The new orientation is to adaptively optimize RM using adaptive algorithms in order to increase performance and maintenance of SND. Dissertation in reference [3] focused on resources in the SND structure and proposed an EPM application which is a novel SDN-based structure for the purpose of assigning privacy preserving multicast in cloud data center.

This dissertation then optimized limited resources to guarantee the security in SDN and presented the design of an investigative approach induced by rule placement in the facet of traffic changes and mobile devices movement. Reference [4], managed Virtual Network's (VN) resources efficiently by using SDN's control plane and extending SDN controller to monitor resource utilization in VNs. Researcher in this reference showed an improvement in VN's acceptance ratio by around 40% and a reduction of VN's resource costs by around 10%. The paper in reference [5], presented a new arrangement of SDN-based management and control for fixed basis network that tools up support for both static and dynamic resource

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 1, January 2017

management applications. The authors in this reference developed an employment algorithm for determining the allocation of managers and controllers in the proposed distributed management and control layer, after that, they selected two specific applications with adaptive load-balancing and energy management objective and explained how the proposed layer be content with the selected applications. Reference [6], explained a summarization on techniques used for resource management with available SDN tactics based on OpenFlow protocol. Authors in [7] presented a theoretical suggestion to distribute and balance the load on controllers based on genetic algorithm when imbalance is detected.The doctoral thesis in [8]proposed and implemented an architecture of cloud platform provided with a mechanism that impacts the dynamic nature of SDN and a use of a virtualization-based technique to protect virtual appliances.

In summary, we can see that our proposal is a novel in this field by using adaptive algorithms (i.e. genetic algorithm) for optimizing resources in SDN using a combination of network software like Mininet, FNSS, OVS controller, Floodlight controller, and VLC media player and different packages like PULP for solving ILP problems.

III. BASIC CONCEPTS

This section describes basic terms and concepts used in this work such as Mininet, OVS and Floodlight Controller and FNSS toolchain.Mininetis a software emulator used to run SDN topologies with a suitable controller [12]. Emulator is hardware or software that enables one computer system (i.e. host) to behave like another computer system (i.e. guest).

Two types of controllers have been used in this paper which are OVS Controller [13] (i.e. Open Virtual Switch Controller) that is a multilayer virtual switch authorized under the open source Apache 2.0 permit. It can operate both as a software-based network switch running within the virtual machine (VM) hypervisors, and as the control stack for dedicated switching hardware; as a result, it has been ported to multiple virtualization platforms, switching chipsets, and networking hardware accelerators. The other controller is the Floodlight Project Controller [14] that is introduced by Big Switch Networks and is based on OpenFlow protocol. This controller can be invaluable or precious to SDN researchers because of its sensible REST API and the use of Java programming language; besides to the satisfying results it offers.

A helpful toolchain named FNSS [1] (Fast Network Simulation Setup) has also been used, which is a set of programming tools used for implementing convoluted software development function or to create a software product. FNSS is based on a set of scenarios in its work as shown in Figure 1. The completed scenario has to be released to FNSS to be achieved in the prioritized goaled software simulator or emulator.

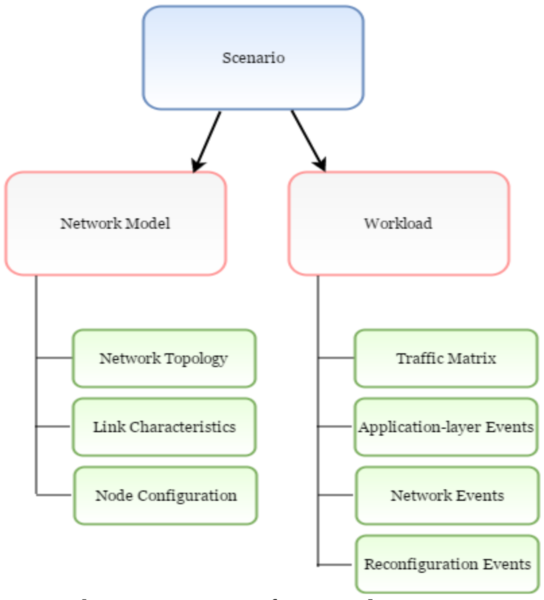


Figure 1: The Components of a Complete FNSS Scenario

IV. PROPOSED ADAPTIVE SOLUTION USING GENETIC ALGORITHM

Resource management optimization has been implemented in this paper based on using adaptive solution by genetic algorithm to find the best capacity for each link in the network according to its current delay, and then assigns a suitable capacity to increase the bandwidth and reduce the delay according to new link capacity. Genetic algorithm has been implemented using regular phases as shown in algorithm 1. The proposed solution has been implemented in a python programming language according to algorithm2.

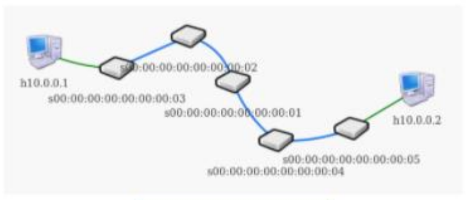
```
Algorithm1: Basic steps of genetic algorithm
Step 1. Generate the required initial population
Step 2. Find the grade of the population
Step 3. For specific range; repeat
    a. Prepare parents
    b. Randomly add other individuals to promote genetic diversity
    c. Mutate some individuals
    d. Crossover parents to create children
    e. Make children the new population
Step 4. Return the best capacity
```

```
Algorithm2: The steps of the proposed solution
Step 1. Import required libraries
Step 2. Set the desired datacenter topology with links and cores
Step 3. Set the delay to 'ms'
Step 4. Assign constant weights for each link in topology
Step 5. Set adaptive capacities to each link according to algorithm1 with 'Mbps' as a capacity units and 'bytes' as a buffer units
Step 6. Set buffer sizes according to bandwidth and delay product
Step 7. Draw the topology with the help of 'nx' library
```

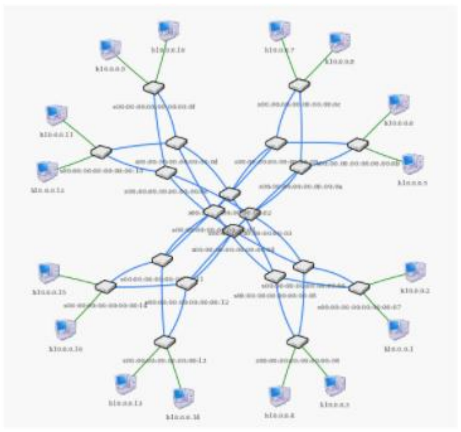
Step 8. Set video transferring properties
Step 9. Convert FNSS topology to Mininet
Step 10. Launch the new Mininet topology with the required controller and start the network
Step 11. Perform video transferring with VLC media player and see the results
Step 12. Stop the network and implement clean command to clear all Ethernet links to be able to run new networks.

V. RESULTS

The proposed solution has been implemented and tested using Mininet emulator, OVS controller, python package, networkx library, PULP library, and FNSS toolchain. All these software tools have been implemented on Ubuntu 16.04 operating system. We implement our proposed solution on three types of network topologies and record the results according; these topologies are Two Tier Datacenter Topology, Three Tier Datacenter Topology, and Fat Tree Datacenter Topology for different number to cores and levels (see figures 2 through 5). We record the delay and bandwidth values according to three types of packets (1 Kbyte, 24 Kbyte, and 64.4 Kbyte) through the ping command between different hosts and results are shown in tables 1 through 6.



Fat Tree Level Two Topology



Fat Tree Level Four Topology

Figure2: Fat Tree Datacenter Topology

Table 1: Floodlight Simulation Results of Fat Tree Datacenter Topology

Packet Size	Average Delay in (ms)	
	Before Using The Proposed Solution	After Using The Adaptive Solution with Genetic Algorithm
1 Kbyte	17.544	20.466
24 Kbyte	389.022	34.927
64.4 Kbyte	1047.974	167.248

Table 2: OVS Simulation Results of Fat Tree Datacenter Topology

Packet Size	Average Delay in (ms)	
	Before Using The Proposed Solution	After Using The Adaptive Solution with Genetic Algorithm
1 Kbyte	12.962	13.339
24 Kbyte	375.67	22.191
64.4 Kbyte	1029.848	39.210

Table 3: OVS Simulation Results of Two Tier Datacenter Topology

Packet Size	Average Delay in (ms)	
	Before Using The Proposed Solution	After Using The Adaptive Solution with Genetic Algorithm
1 Kbyte	8.831	16.139
24 Kbyte	371.503	33.729
64.4 Kbyte	1025.229	50.644

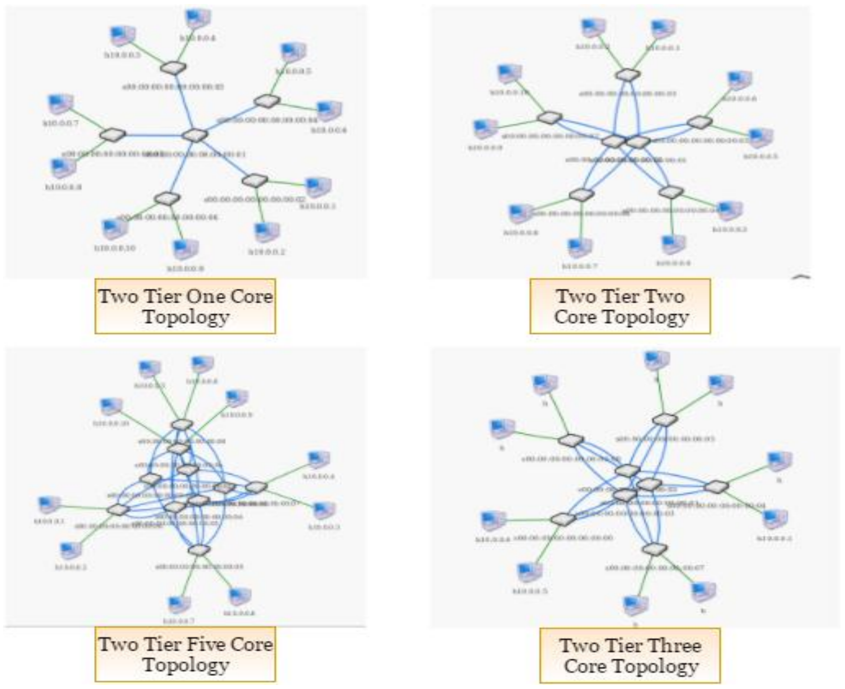
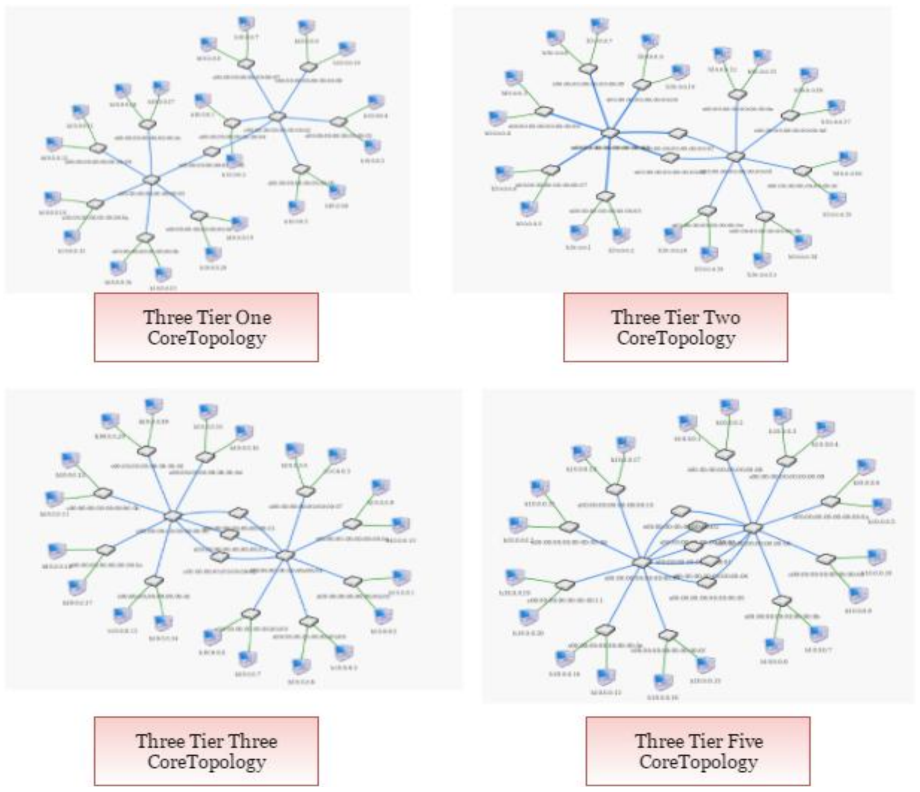

Two Tier One Core Topology
Two Tier Two Core Topology
Two Tier Five Core Topology
Two Tier Three Core Topology
Figure3: Two Tier Datacenter Topology

Table 4: Floodlight Simulation Results of Two Tier Datacenter Topology

Packet Size	Average Delay in (ms)	
	Before Using The Proposed Solution	After Using The Adaptive Solution with Genetic Algorithm
1 Kbyte	19.721	14.7644
24 Kbyte	370.901	43.1662
64.4 Kbyte	1038.980	45.1587


Three Tier One CoreTopology
Three Tier Two CoreTopology
Three Tier Three CoreTopology
Three Tier Five CoreTopology
Figure4: Three Tier Datacenter Topology

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 1, January 2017

Table 5: OVS Simulation Results of Three Tier Datacenter Topology

Packet Size	**Average Delay in (ms)**	
	Before Using The Proposed Solution	After Using The Adaptive Solution with Genetic Algorithm
1 Kbyte	17.712	8.859
24 Kbyte	385.549	18.1145
64.4 Kbyte	1037.191	34.8235

Table 6: Floodlight Simulation Results of Three Tier Datacenter Topology

Packet Size	**Average Delay in (ms)**	
	Before Using The Proposed Solution	After Using The Adaptive Solution with Genetic Algorithm
1 Kbyte	23.689	18.0775
24 Kbyte	388.254	27.95375
64.4 Kbyte	1044.579	57.91575

VI. PERFORMANCE EVALUATION AND CONCLUSION

By comparing the results from the previous section, we can see the efficiency of our algorithm. The performance of OVS controller can be shown in figures 5, 6, and 7; while the performance of Floodlight controller can be shown in figures 8, 9, and 10. The OVS controller reduced the delay by 94.4% while the bandwidth increased from 950 Mbps- 1.82 Mbps to 37.075 Mbps- 42.9 Mbps as an average. The Floodlight controller reduced the delay by 90.1% while the bandwidth increased from 952 Mbps- 1.7 Mbps to22.97 Mbps- 27.45 Mbps as an average. We can see the performance of our proposed solution by running videos of 2.00 MByte and 5.00 Mbyte in VLC media player between two hosts.The performance of the proposed solution can be seen when running the video before and after implementing the genetic algorithm as shown in figures 11 and 12.

As a comparison in performance between OVS and Floodlight controllers, we can see that OVS controller performs better than Floodlight controller in fat tree topologies whereas Floodlight controller performs better than OVS in other types of datacenter topologies.It is worth mentioning that although OVS performs better than Floodlight controller in the percentage but OVS controller cannot maintain high number of cores in datacenter topology while Floodlight controller can work properly with more than one core in the same topology.

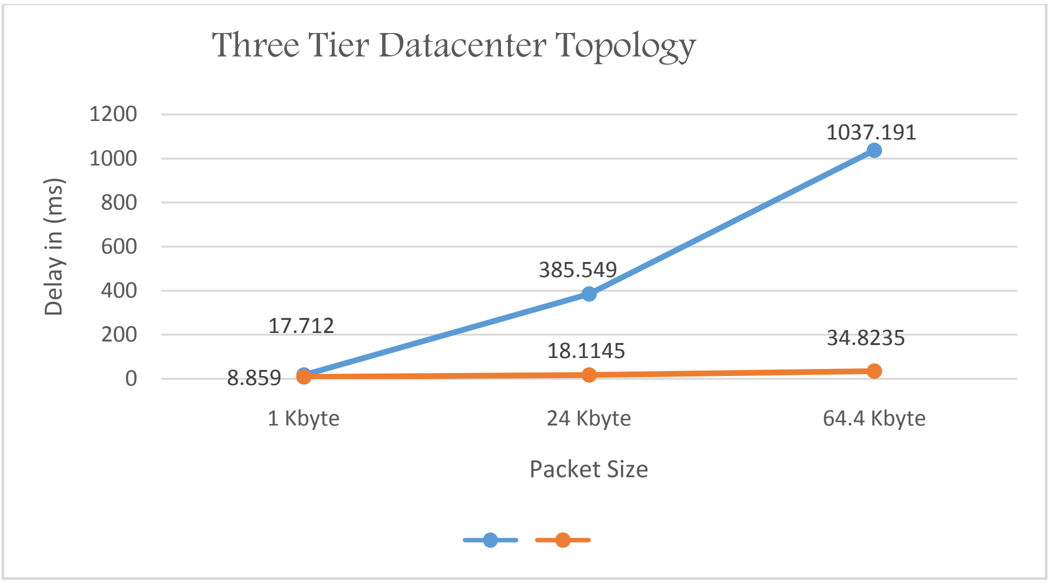


Figure5: Performance Evaluation for Three TierDatacenter Topology using OVS Controller

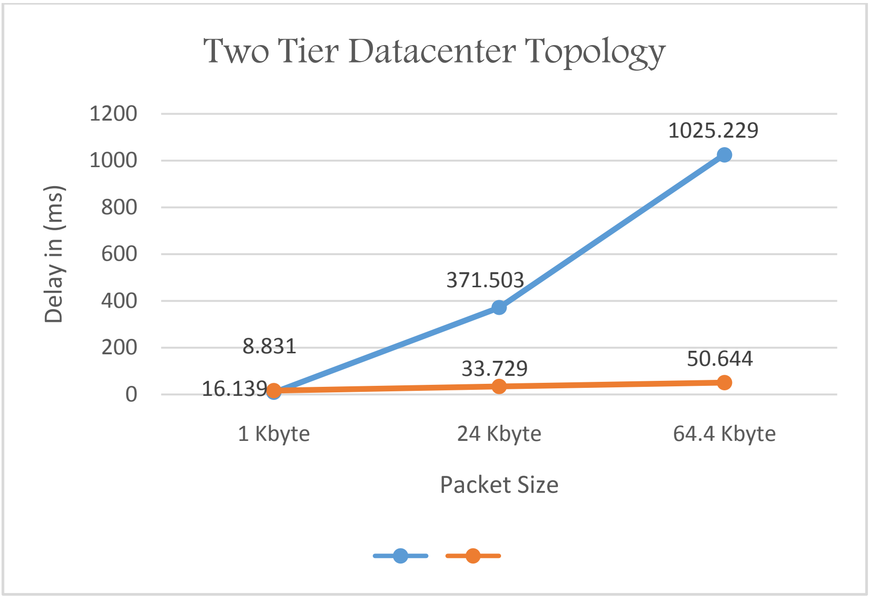


Figure6: Performance Evaluation for Two Tier Datacenter Topology using OVS Controller

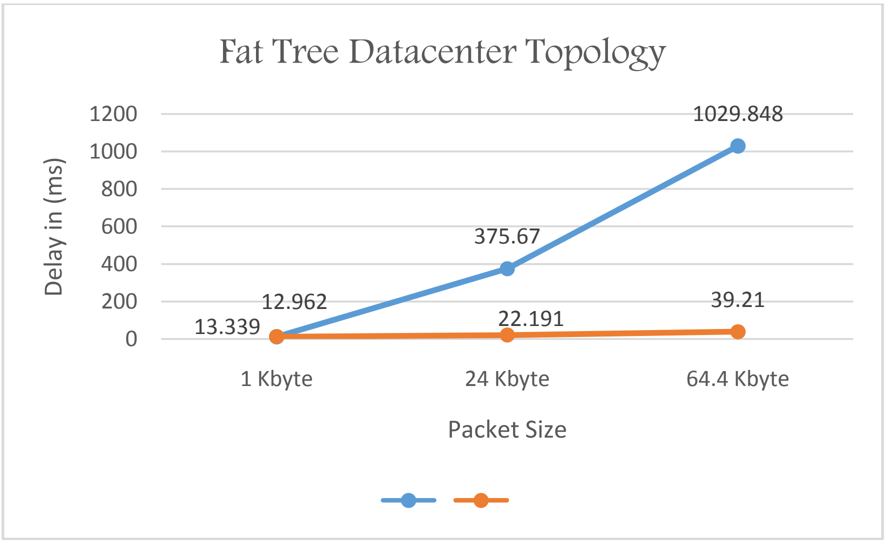


Figure7: Performance Evaluation for Fat Tree Datacenter Topology using OVS Controller

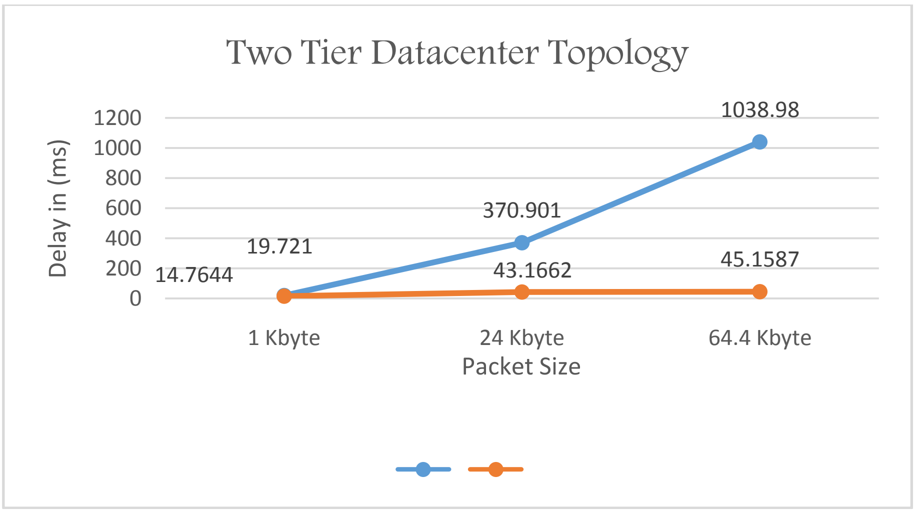


Figure8: Performance Evaluation for Two Tier Datacenter Topology using Floodlight Controller

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 1, January 2017

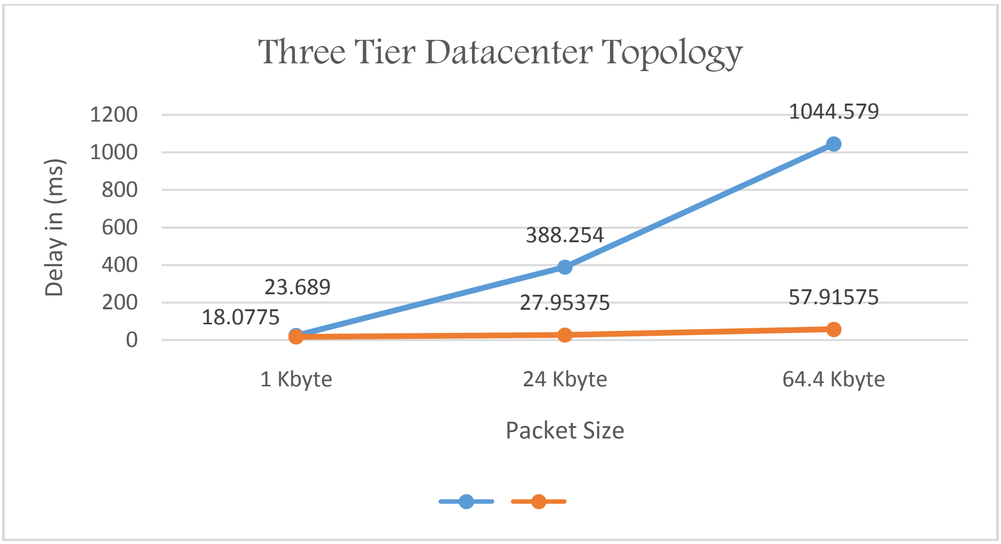


Figure9: Performance Evaluation for Three Tier Datacenter Topology using Floodlight Controller

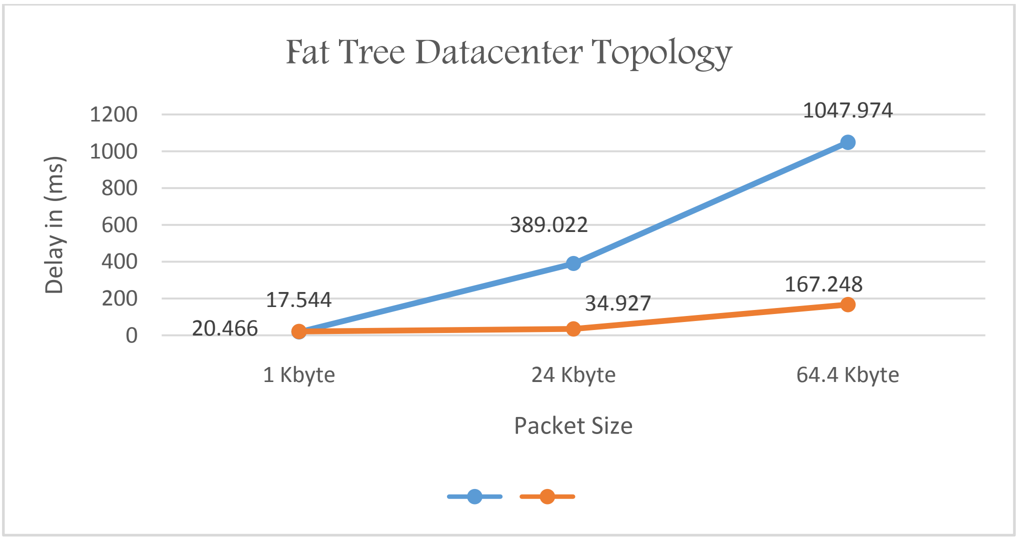
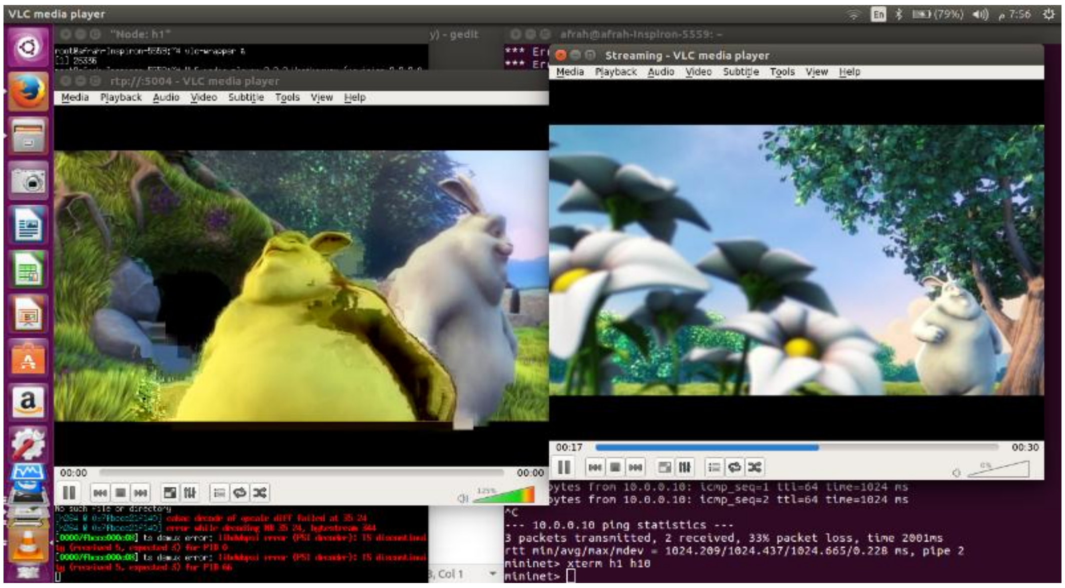


Figure10: Performance Evaluation for Fat Tree Datacenter Topology using Floodlight Controller



```
No such file or directory
[h264 @ 0x7fbccc21f140] [illegible]
[h264 @ 0x7fbccc21f140] [illegible]
[0000/fbccc000c08] ts demux error: [illegible]
[illegible]
[0000/fbccc000c08] ts demux error: [illegible]
[illegible]
```

```
bytes from 10.0.0.10: icmp_seq=1 ttl=64 time=1024 ms
bytes from 10.0.0.10: icmp_seq=2 ttl=64 time=1024 ms
^C
--- 10.0.0.10 ping statistics ---
3 packets transmitted, 2 received, 33% packet loss, time 2001ms
rtt min/avg/max/mdev = 1024.209/1024.437/1024.665/0.228 ms, pipe 2
mininet> xterm h1 h10
mininet>
```

Figure11: Video Performance before Implementing Genetic Algorithm

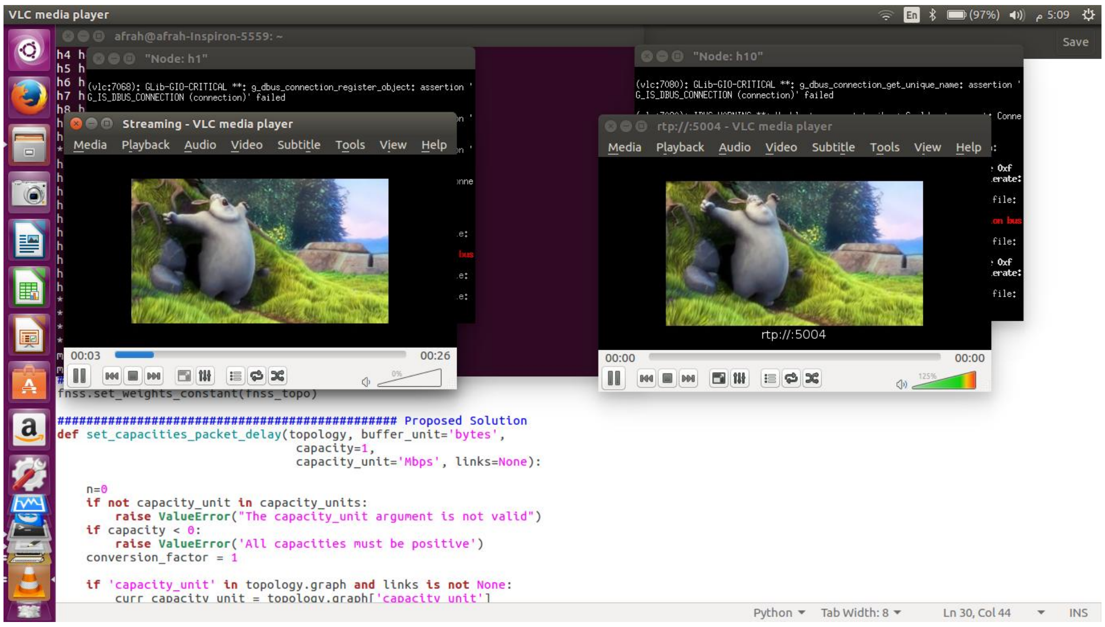
IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified

Vol. 6, Issue 1, January 2017



```
(vlc:7068): GLib-GIO-CRITICAL **: g_dbus_connection_register_object: assertion '
G_IS_DBUS_CONNECTION (connection)' failed
```

```
(vlc:7080): GLib-GIO-CRITICAL **: g_dbus_connection_get_unique_name: assertion '
G_IS_DBUS_CONNECTION (connection)' failed
```

```
fnss.set_weights_constant(fnss_topo)

############################################### Proposed Solution
def set_capacities_packet_delay(topology, buffer_unit='bytes',
                                capacity=1,
                                capacity_unit='Mbps', links=None):

    n=0
    if not capacity_unit in capacity_units:
        raise ValueError("The capacity_unit argument is not valid")
    if capacity < 0:
        raise ValueError('All capacities must be positive')
    conversion_factor = 1

    if 'capacity_unit' in topology.graph and links is not None:
        curr_capacity_unit = topology.graph['capacity_unit']
```

VLC media player
afrah@afrah-Inspiron-5559: ~
"Node: h1"
"Node: h10"
Streaming - VLC media player
rtp://:5004 - VLC media player
Media Playback Audio Video Subtitle Tools View Help
rtp://:5004
00:03
00:26
00:00
00:00
Save
Python
Tab Width: 8
Ln 30, Col 44
INS

Figure12: Video Performance after Implementing Genetic Algorithm

REFERENCES

[1] L. Saino, C. Cocora, and G. Pavlou, A Toolchain for Simplifying Network Simulation Setup, in Proceedings of the 6th International ICST Conference on Simulation Tools and Techniques (SIMUTOOLS '13), Cannes, France, March 2013.

[2] T. Ahmed, Resources Management in Software Defined Environment.

[3] H. Li, Resource Optimizations in SoftwareDefined Networking, dissertation submitted in partial fulfillmentof the requirements for degree of doctor of philosophy in computer science and engineering, Ph.D, 2015.

[4] R. Mijumbi, J. Serrat, J. Rubio-Loyolay, N.Boutenz, F. D.Turckz and S.Latr´ex. Dynamic Resource Management in SDN-basedVirtualized Networks, ISBN 978-3-901882-67-8, 10th CNSM and Workshop ©2014 IFIP.

[5] D.Tuncer, M. Charalambides, S.Clayman, and G.Pavlou, Adaptive Resource Management and Controlin Software Defined Networks.

[6] T.Zinner, M. Jarschel, A. Blenk, F. Wamser, W. Kellerer, Dynamic Application-Aware Resource ManagementUsing Software-Defined Networking: Implementation Prospects and Challenges, IEEE International Workshop on Quality of Experience Centric Management (QCMan), 2014, 10.1109n/noms.2014.6838404.

[7] S. Kang, G. Kwon, Load Balancing Strategy of SDN Controller Based on Genetic Algorithm, Advanced Science and Technology LettersVol.129 (Mechanical Engineering 2016), pp.219-222.

[8] A. Marotta, Architectures and Algorithms forResource Management in Virtualized Cloud Data Centers, a doctoral thesis in the Department of Electric Engineering and Information TechnologiesMarch 2015.

[9] I.Baran, Adaptive Algorithms for Problems InvolvingBlack-Box Lipschitz Functions, Master thesis at the Massachusetts institute of technology, June 2004.

[10] D. Jakobovic and M. Golub, Adaptive Genetic Algorithm, Journal of Computing and Information Technology – CIT, 1999, 3, 229-235.

[11] R. Maniu and L. A. Dumitru, Exploring the Possibilities of a Self-Regulating SDN Controller, “MirceacelBatran” Naval Academy Scientific Bulletin, Volume XVIII – 2015 – Issue 1.

[12] A.S. Dawood, M.N. Abdullah, A Survey and a Comparative Study on Software-Defined Networking, International Research Journal of Computer Science (IRJCS) ISSN: 2393-9842, issue 08, Volume 3 (August 2016).

[13] http://openvswitch.org/

[14] https://www.sdxcentral.com/sdn/definitions/sdn-controllers/open-source-sdn-controllers/what-is-floodlight-controller/