PAPER • OPEN ACCESS

Optimal computer network based on graph topology model

Optimal computer network based on graph topology model

Q A A Ruhimat, G W Fajariyanto, D M Firmansyah, Slamin
Network and Security Research Group, Faculty of Computer Science, University of Jember, Indonesia

E-mail: qurrotaaar@unej.ac.id.

Abstract. In large computer network design, it is desirable to have connections which achieve the most efficient and reliable communication. In such design, there are several factors which should be considered. Two of the factors which seem to appear most frequently are, (a) the number of connection which can be attached to a computer is limited, and (b) a short communication route between any two computers is required. Thus, it will end up with a large network subject to these constraints. The computer network can be modelled as a graph, where each computer is represented by a vertex, and the connection between the two computers is represented by an edge. In this paper, we compare the efficiency and reliability of computer networks based on their graph topology model. The method used in this study is literature study, designing the new topology, and making simulation and testing using a software simulator called *Graphical Network Simulator 3* (GNS3). The results of this study are the comparison of the application of prism and petersen graph topologies in computer networks topology.

1. Introduction

Graph theory is a branch of science that has received much attention recently. The concept of graph theory can be found in various fields. In general, the concept of graph theory is used in modeling a problem and finding the solution. Graph theory was first introduced by Leonhard Euler, a Swiss mathematician in 1736. Euler used the concept of graph theory to solve the famous problem of Konigsberg bridge. [9] defined graph G as a set of pairs $(V(G), E(G))$, where $V(G)$ is a non-empty finite set of elements called vertex, and $E(G)$ is a set (maybe empty) of non-pairs sorted $\{u, v\}$ from vertices $u, v \in V(G)$ called edge that connects these vertices.

There are some terms in graph theory, such as: order, size, degree, distance, and diameter. Order is the number of vertices in the graph, size means the number of edge in the graph, degree is the number of connection to vertex, distance is the length of the shortest path between two vertices and diameter is the maximum distance between any pair of vertices. In the graph theory, if each vertex on graph has the same number of neighbors or same degree, then the graph is considered as regular graph [1].

As time goes by, graph theory also develops rapidly in various fields, in both theory and applications. One example of the application of graph theory is designing computer network. Computer network is needed in various fields of work as a means of exchanging and updating information. Computer network is a set of computers and other hardware components that are connected by a communication path; so that, between the computer and other hardwares can share information and resources. The computer network can be represented as a graph where each vertex can be considered as a computer or other hardwares (router, server, or other), and each edge describes the interconnection media between these devices [9]. In this case, each device must have interconnection media if it wants to communicate with other devices, it is like a vertex (node) that must have a path to be able to connect with another node.

Computer network is a set of computers and other hardware components that are connected by a communication path, so that between the computer and other hardware can share information and resources. [6] and [7] has classified the computer network based on their topologies, where each type has a different graph representation as described below. Besides, the illustration of each graph topology below is represented in the figure 1.

1.1. Ring topology (cycle)
Ring topology is a type of network topology that uses applications from graph theory, which is cycle graph. A cycle graph is a simple graph which has two degrees of vertices. Cycle graph with n vertices denoted by C_n. Ring topology is a series of vertex-formed network topology. Each vertex (node) is connected to the other two nodes, so that they formed a network which looks like a circle or ring. In this topology, all of the data is sent between vertices in the network and the data will move from one vertex to another circularly. In general, this type of network topology distributes the data in one direction. The advantages of this type are the easy installation and the inability to monopolize the data from one computer. The flow of data is in the direction of these edges. So, if there is a broken edge, the data distribution path will be completely cut off. If it happens, the identification of the problem will be difficult because it must be searched one by one.

1.2. Bus topology (path)
This topology is in the form of a track. In this topology, a single cable is used to connect all devices on the net. This topology is commonly used in temporary network. The advantages of this type of physical topology include ease of installation and requires fewer cable than the star topology. In addition, the cost is relatively cheaper than the others. However, this topology is not recommended for long-term use, one of the reasons is because this topology has a low security system, and similar to the ring topology, its distribution path is one-way. Therefore, if there is a distribution failure in one node, it will interferes the other node connections.

1.3. Star topology
Star topology is a network topology in the form of convergence from the central vertex (center) to each vertex or user. This topology adopts a star graph. In the star topology, the data distribution path sent from one vertex does not need to go through another vertex, but directly go to the central vertex and then the data will be directed towards the target node. In this topology, it is easy to add new vertex without worrying about disrupting the overall data distribution. In general, each vertex is not interconnected so that the interference from one vertex will not disturb the data distribution at the edge with the other vertex. However, if the central vertex is severely damaged, the effect of the entire system will stop completely and all the vertices connected to the central node cannot be used again. Judging from the economic field, this topology requires expensive costs because it does not only require long cables, but also additional devices as the central node.

1.4. Mesh topology
Mesh topology uses a complete graph form. A complete graph is a simple graph where each vertex is interconnected. A complete graph with n vertices is denoted by K_n. The number of edges in a complete graph can be found with the formula $n(n\text{-}1)/2$. In mesh topology, each device is connected to each other directly so that it can communicate directly with the intended device (dedicated links). This type of topology has a stronger resistance because it provides a variety of paths for each vertex. However, in terms of economic, it needs high cost and too complicated for a normal network

operation. This type of topology has a stronger resistance because it provides a variety of paths for each vertex. However, in terms of economic, it needs high cost and too complicated for a normal network operation. Nevertheless, the topology is usually used when there is only a few vertices are available to be connected.

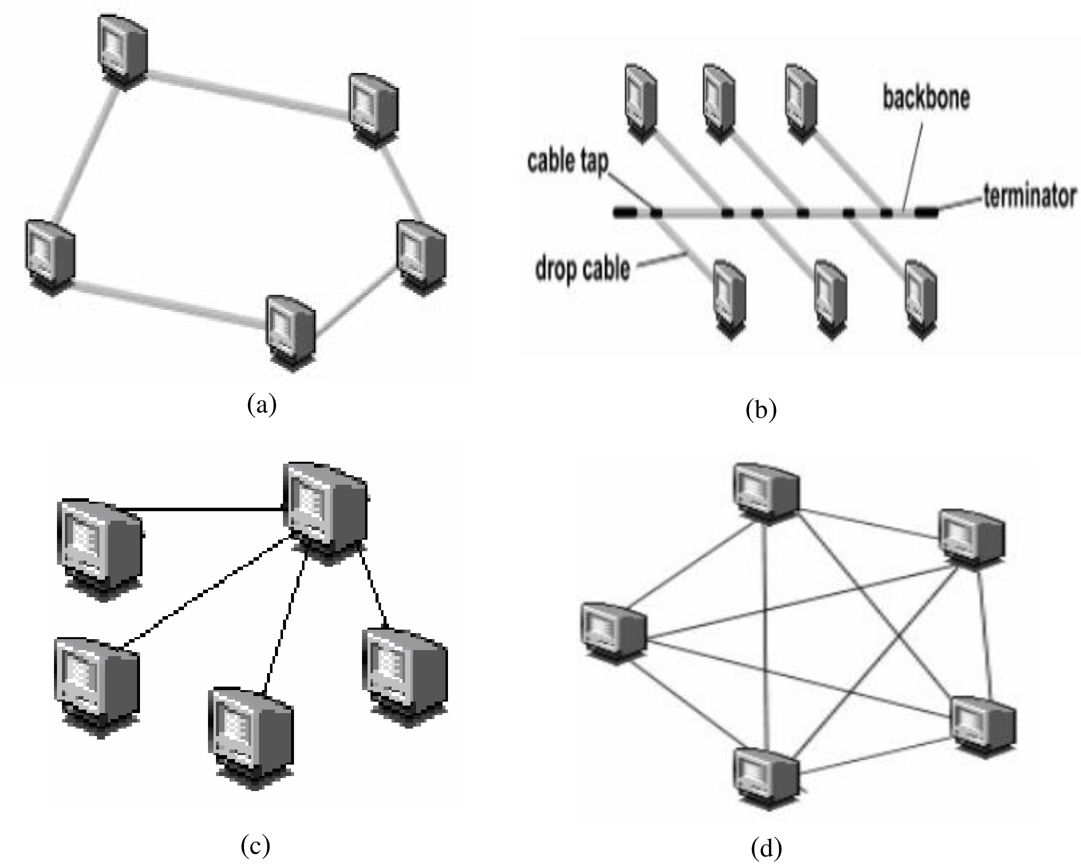

Figure 1. The illustration of network topology: (a) ring topology; (b) bus topology; (c) star topology; (d) mesh topology.

1.5. Prisma and petersen topology

In graph theory, the petersen graph is an undirected graph with 10 vertices and 15 edges, while prism graph is a graph that has one of the prisms as its skeleton. The prism and petersen graph with 10 vertices are the regular graph with 3-degrees on each vertex. The difference from both paragraphs is its diameter. The diameter of prism graph with 10 vertices is three and the diameter of the petersen graph with 10 vertices is two [11][12].

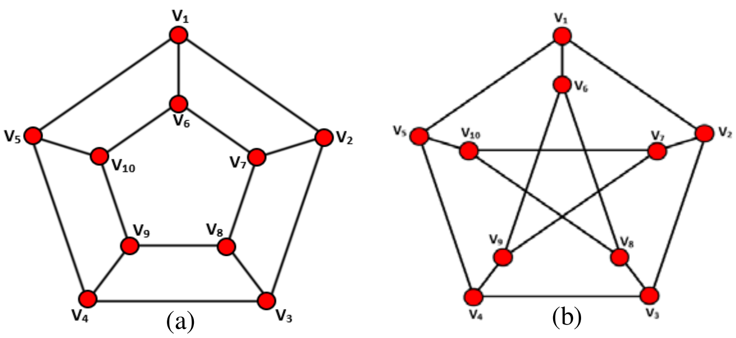

Figure 2. (a) prism graph with 10 vertices; (b) petersen graph with 10 vertices.

In large computer network design, it is desirable to have connections which achieve the most efficient and reliable communication. In such design, there are several factors which should be considered. Two of the factors which seem to appear most frequently are (a) the number of connection which can be attached to a computer is limited, and (b) a short communication route between any two computers is required [7][10]. This study discusses the application of graph theory, especially the prism graph and petersen graph in computer network topologies.

2. Methodology

The methods used in this study were literature study, designing the new topology, and making simulation. The study began with reading the literatures about the application of graph theory that has been used in computer network topologies and then proceeded design a new computer network topology model using a prism graph and petersen graph.

Based on several reasons from our literature study, such as analysis of their size and diameter, we chose the prism and petersen topology. Although it needs an enough large number of cables (size), the data distribution process here is stable enough as well. In addition, if a failure happens in one path, it will not much disturb other distribution data.

In this case, we used 10 vertices for the simulation process. The simulation and testing were done using a software simulator called *Graphical Network Simulator 3* (GNS3). Then we compared the performance of computer networks between prism topology and petersen graph topology. The scheme design of prism and petersen topologies is shown in figure 3 and 4.

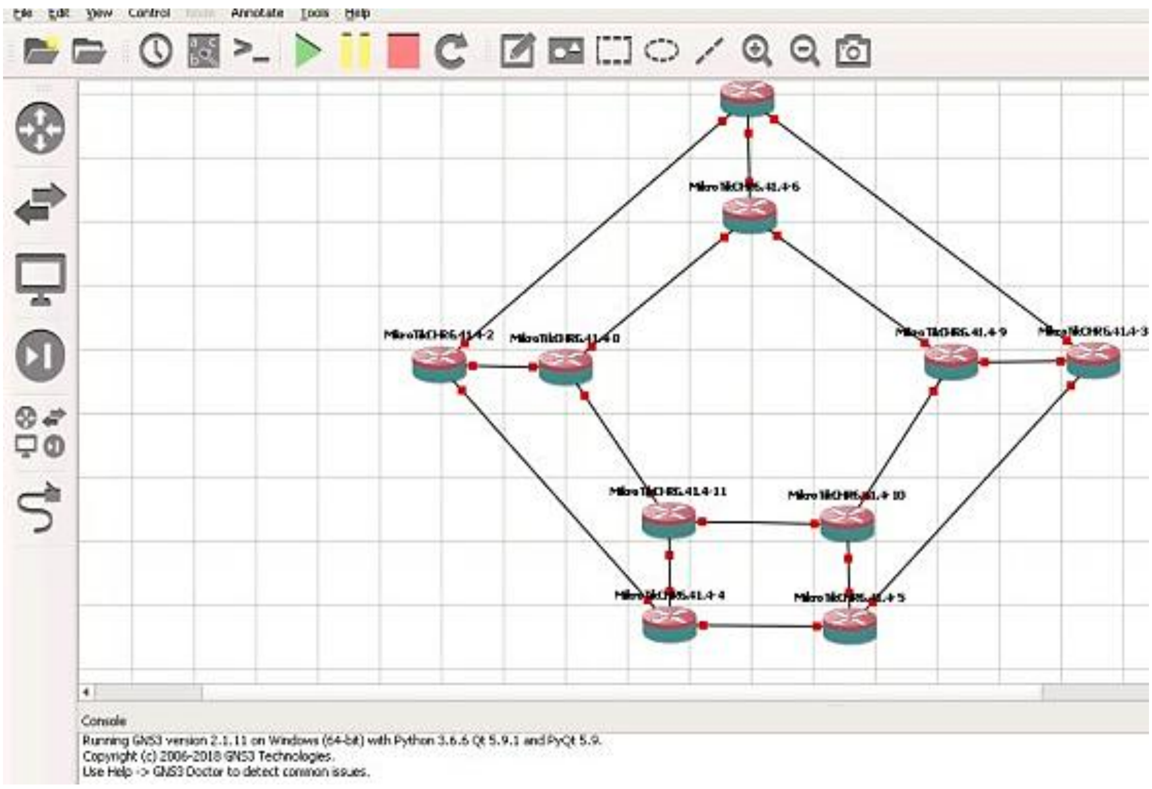

Figure 3. Scheme design of prism topology using GNS3.

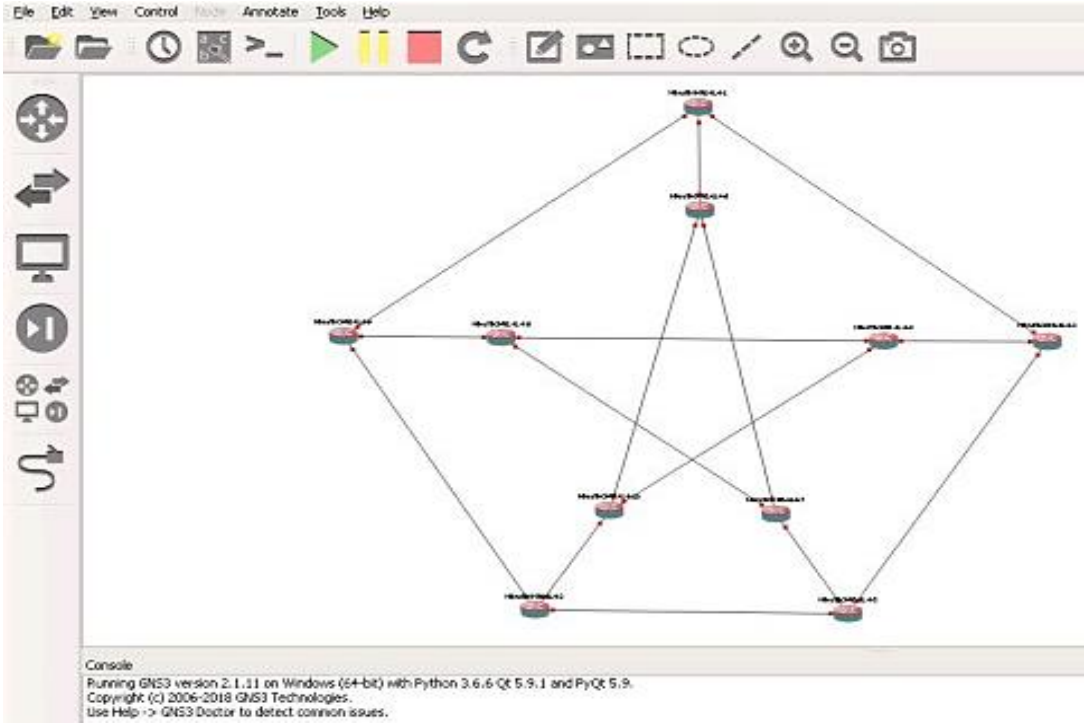

Figure 4. Scheme design of petersen topology using GNS3.

3. Main results

The data simulation processed with GNS3 simulator was applied to determine the difference in *Quality of Service* (*QoS*) from each topology in terms of its time per request and the successful requested data package. GNS3 is an open source graphical network simulator program that can simulate a complex network topologies than the other simulators. Here is a screen display in GNS3 software.

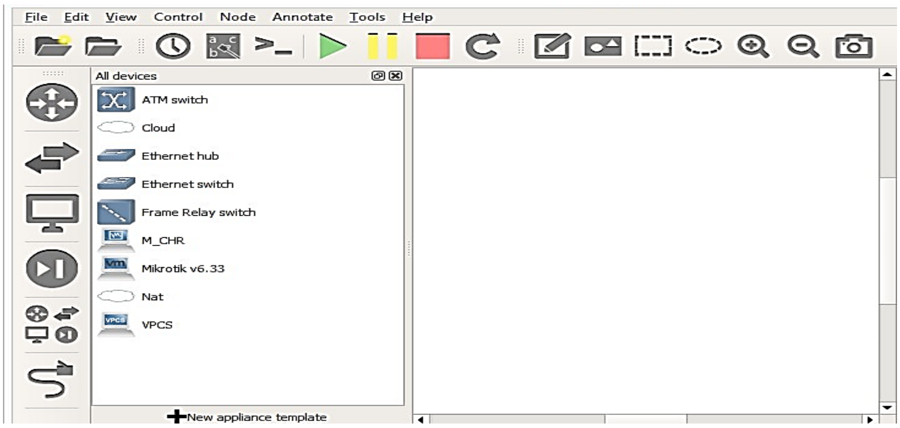

Figure 5. Initial view of *Graphical Network Simulator 3* (GNS3).

GNS3 is a modular network design simulator that can be connected to a real networks. Simulation is one of the important steps in network research. Network simulators allow one to model computer networks arbitrary by determining the behavior of network nodes and communication channels. It provides a virtual environment for various desired features such as network modeling based on certain criteria and analyzing its performance under different scenarios. The work principle of GNS3 is emulating Cisco IOS on a computer; so that, the PC can function like a router or even a switch, by activating the function of ethernet switch card [2][4].

Quality of Service (QoS) refers to the ability of a network to provide improved service to selected network traffic over various underlying technologies including Frame Relay, ATM, Ethernet, SONET, and IP-routed networks [3]. *QoS* is an *end-to-end* architecture system and it is not a feature owned by network. *Quality of* Service from a network refers to the level of speed and reliability of the delivery of various types of data loaded in a communication. Different applications need certain *QoS* requirements so that there will not be too many packets lost during the transmission process, good real-time services, low delay, and good bandwidth allocation. Speed performance refers to the delivery level reliability from various types of data loads in a communication which includes: throughput, delay and packet loss [1][8]. While, Seshan et al. in [5] developed a system for discovering network performance called SPAND (Shared Passive Network Performance Discovery) and they conceptualized the network performance in terms of characterisctics used to identify performance such as latency, loss rate, available bandwidth, and routing metrics.

In this research, the network design (as shown in figure 3 and 4) used 10 computers in each topology. Each of them has a computer as a tester, a computer as a load balancer, and another one as the recipient of the data distribution. The function of a computer tester is to receive the initial data package and the load balancer is the data distribution divider for the remaining 8 computers. The simulation begins by sending 100 requested data packages to the computer tester of each topology and then the load balancer divides the requested data to the remaining computer. By doing testing through sending this requested data package, we knew which topology model work better from its time per data request and the successful requested data package. The analysis was performed from the running request data package using GNS3 software. The results are shown by figure 6 and 7.

```
==== PRISM RESULT ====
root@UbuntuDockerGuest-1:~# ab -n 100 -c 10 -s 60 http://192.168.0.62:80/
This is ApacheBench, Version 2.3 <$Revision: 1706008 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking 192.168.0.62 (be patient).....done

Server Software:        nginx/1.10.3
Server Hostname:        192.168.0.62
Server Port:            80

Document Path:          /
Document Length:        11321 bytes

Concurrency Level:      10
Time taken for tests:   65.683 seconds
Complete requests:      100
Failed requests:        14
   (Connect: 0, Receive: 0, Length: 14, Exceptions: 0)
Total transferred:      1098346 bytes
HTML transferred:       1072558 bytes
Requests per second:    1.52 [#/sec] (mean)
Time per request:       6568.300 [ms] (mean)
Time per request:       656.830 [ms] (mean, across all concurrent requests)
Transfer rate:          16.33 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:        1  104 136.4     46     570
Processing:     9 6176 18046.9    198   60238
Waiting:        6 6114 18064.5     99   60167
Total:         12 6280 18065.8    281   60749

Percentage of the requests served within a certain time (ms)
  50%    281
  66%    460
  75%    572
  80%    670
  90%  60016
  95%  60112
  98%  60463
  99%  60749
 100%  60749 (longest request)
```

Figure 6. The result of prism topology.

```
==== PETERSEN RESULT ====
root@UbuntuDockerGuest-1:~# ab -n 100 -c 10 http://192.168.0.62:80/
This is ApacheBench, Version 2.3 <$Revision: 1706008 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking 192.168.0.62 (be patient).....done

Server Software:        nginx/1.10.3
Server Hostname:        192.168.0.62
Server Port:            80

Document Path:          /
Document Length:        11321 bytes

Concurrency Level:      10
Time taken for tests:   9.216 seconds
Complete requests:      100
Failed requests:        12
   (Connect: 0, Receive: 0, Length: 12, Exceptions: 0)
Total transferred:      1107068 bytes
HTML transferred:       1081064 bytes
Requests per second:    10.85 [#/sec] (mean)
Time per request:       921.592 [ms] (mean)
Time per request:       92.159 [ms] (mean, across all concurrent requests)
Transfer rate:          117.31 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:        4  212 199.5    174    1201
Processing:    20  663 324.4    722    2301
Waiting:       19  581 267.8    647     940
Total:         27  875 381.4   1000    2310

Percentage of the requests served within a certain time (ms)
  50%   1000
  66%   1030
  75%   1040
  80%   1041
  90%   1111
  95%   1200
  98%   1957
  99%   2310
 100%   2310 (longest request)
```

Figure 7. The result of petersen topology.

From the data above, the petersen topology model has a better result than the prism one. The parameters seen are from their time per request, the successful requested data package. The failed requests of the prism topology model is greater than the petersen one. It means that the number of successful requested data package of petersen topology model is better than the prism one. Hence, we can say that the petersen one is more reliable for sending the requested data package to be successful. Besides, the number of time per request (in mean) for prism topology model is 6568,300 ms and the petersen is just 921,592 ms. It means that the time needed for each request (in mean) by petersen to process data from the beginning (since the data sent to the computer tester) until the data is received by each remaining computer (whether it's successful or broken), is more efficient than the prism one. In addition, the total time needed for petersen topology model to process all of the data requested (100%) is just 2130 ms. It is smaller than the Prism one which is need 60749 ms.

4. Conclusion

From the simulation and data result as shown in figure 6 and 7, we have proved that the petersen topology model is more reliable and more efficient than the prism one, in sending the requested data package in terms of its time per request and the successful requested data package.

Because this research was generally made by making design and doing simulation and testing, so our suggestion for the future research is to implement this model into a real condition.

Acknowledgment
The authors would like to thank the Network and Security Research Group Faculty of Computer Science University of Jember for funding this research through “Hibah Keris” program in the academic year of 2018.

References
[1] Dekker A H and Colbert B D 2004 *Network Robustness and Graph Topology* (Canberra: Defence Science and Technology Organisation)
[2] Chaudhary R Sethi S Keshari R and Goel Sakshi 2012 A Study of Comparison of Network Simulator-3 and Network Simulator-*2 International Journal of Computer Science and Information Technologies Vol 3(1)* p 3085-3092
[3] Cisco 2009 *Cisco IOS Quality of Service Solutions Configuration Guide: Quality of Service Overview* (USA: Cisco Systems, Inc) p 1
[4] Jha R K and Kharga P 2015 Advance Open Source Simulator: NS-3 *International Journal of Computer Science and Engineering Vol 3 Issue 12*
[5] Kigodi O J, Kisangiri M and Machuve D 2013 Review on Network Performance: Meaning, Quantification and Measurement journal of Internet and Information Systems Vol 3(2) pp 16-19
[6] Meador B 2008 *A Survey of Computer Network Topology and Analysis Examples (*Washington: Computer Science & Engineering-Washington University in St Louis*)* p 2-3
[7] Pandya K 2013 Network Structure or Topology *International Journal of Advance Research in Computer Science and Management Studies Vol 1 Issue 2 July 2013*
[8] Rogier B 2016 Measuring Network Performance: Links Between Latency, Throughput and Packet Loss *Preprint* accedian.com/enterprises/blog/measuring-network-performance-latency-throughput-packet-loss/
[9] Slamin 2009 *Desain Jaringan Pendekatan Teori Graf* (Jember: Jember University Press) p 11
[10] Webb J, Docemmilli F and Bonin M 2015 *Graph Theory Applications in Network Security* (Wayville SA: Central Queensland University)
[11] Wikipedia – Petersen Graph *Preprint* en.wikipedia.org/wiki/Petersen_graph
[12] Wikipedia – Prism Graph *Preprint* en.wikipedia.org/wiki/Prism_graph