International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 5, Issue 5, May 2015)

Simulation of a Fault Tolerant University Campus Area Network

Adeyanju, I.A.[1], Adeosun, O.O.[2], Awobayo, T.I.[3], Olayeni B.R.[4]

Department of Computer Science and Engineering, Ladoke Akintola University of Technology, PMB 4000, Ogbomoso, Oyo State, Nigeria

***Abstract*— The University has peculiar network needs that must be met for effective dissemination of information. A network allow sharing of files, data and other information giving authorized users the ability to access information stored on other computer on the network. The Campus Area Network (CAN) is still faced with so many problems with the ever increasing demand of computer network due to inadequate fault tolerance. In this paper, fault tolerance was simulated using redundancy and its effect in a network investigated. In order to improve network reliability, information about a CAN was gathered and the problem analysed. The analysis was used in formulating a CAN model. Also statistics were collected for evaluation and effectiveness of the model. It was investigated that the Database server response time with fault tolerance and non-fault tolerance are 0.52sec & 0.97sec respectively which gradually reduces gradually to a constant values of 0.28sec & 0.33sec respectively. The File server response time for both fault tolerance and non-fault tolerance are 2.4 sec & 3.4sec respectively initially and then reduces gradually to a constant value of 0.75sec & 1.8sec respectively. Furthermore, Hypertext transfer protocol server response time for both fault tolerance and non-fault tolerance are 3sec & 3.5sec respectively initially which reduces gradually to a constant value of 1.7sec & 2sec respectively. We present in this paper the simulation and analysis of a fault tolerant CAN in the OPNET Modeller environment.**

***Keywords*— simulation, fault tolerance, CAN, OPNET, Network**

I. INTRODUCTION

The scope of communication has increased significantly in the past decade. This boom in communication would not have been without progressively advancing computer network. For easy sharing and access of information among users, the computer involved must be interconnected. The interconnection of a building or group of building into one enterprise network that consist of many local area networks (LANs) is known as campus area network (CAN). The physical systems that compose a network are subjected to a wide range of problems ranging from signal distortion to component failure. The CAN is having problem of traffic in network due to inadequate fault tolerance procedure employed by the user or institution.

The server fault tolerance with primary backup system provides fault tolerance capabilities by replicating service state on one or more backup servers. Clients interact with the primary server. Backup servers monitor the health of primary server and in case of a primary server failure, one of the backup server is promoted to act as the new primary server (fail over). Primary backup techniques have been used to build numerous dependable systems [7]. For simulation purpose at the design stage and prior to actual network deployment, the network simulator used was OPNET. It was used to imitate the real life scenario on the CAN.

II. THEORETICAL BACKGROUND

The information about CAN is widely reported in the literature. CANs allows easy file sharing between different departments, all the files are usually shared on the server machine of each LAN. This type of network offers a lot of simplicity in the transfer and downloading of files [13].

A. Campus Area Network (CAN)

Campus area networks are classified by scale, components and connection method. How a CAN is connected and what components it uses will determine how fast, reliable, and accessible the network is.

1) Switches: Switches operate at the data link layer. The three main functions of a bridge are also true of a switch: they learn, forward, and remove loops. However, the switches used have many more features than bridges; for instance, they make their switching decisions in hardware by using application-specific integrated circuits (ASICs). [17]

2) Routers: This is a networking layer device, commonly specialized hardware, which forwards data packets between computer networks. This creates an overlay internetwork, as a router is connected to two or more data lines from different networks. When a data packet comes in one of the lines, the router reads the address information in the packet to determine its ultimate destination. Router performs the “traffic directing” functions on the internet.

A data packet is typically forwarded from one router to another through the networks that constitute the internetwork until it reaches its destination node [15].

3) Ethernet: This has been the world's most popular wired computer network technology for over several decades now which join local devices together via Ethernet cables. The most widely LAN technology is usually specified in a standard, IEEE 802.3. It was originally developed by Xerox from an earlier specification called Aloha net (for the Palo Alto Research Center Aloha network) and then developed further by Xerox, DEC, and Intel. Ethernet LAN typically uses coaxial cable or special grades of twisted pair wires [4]. In this paper, it was also used in wireless LANs.

4) Server: A Server is a device with a particular set of programs or protocols that provide various services, which other machines or clients request, to perform certain tasks. Together, a server and its clients form a client/server network to provide routing systems and centralized access to information, resources, stored data, etc. Servers are generally not used by humans directly in campus area network, but rather run continuously to provide services to the other computers (and their human users) on the campus area network. Services provided can include printing and faxing, software hosting, file storage and sharing, messaging, data storage and retrieval, complete access control (security for the network's resources, and many others) [16].

B. Fault Tolerance

There is no way we can make a network one hundred percent reliable but the network can be survivable from intentional and unintentional attacks by employing the good fault tolerance procedure. Fault tolerance is the property that enables a computer or network to continue operating properly in the event of the failure of (one or more fault within) some of its components. A fault tolerant design enables a system to continue its intended operation possibly at a reduced level rather than failing completely, when some part of the system fails [3].

Fault tolerance requires some basic characteristics which include: no single point of failure, fault isolation to the failing component, fault containment to prevent propagation of the failure and availability of reversion modes.

Also fault-tolerant design has some disadvantages that are not obvious and they include: interference with fault detection in the same component, interference with fault detection in another component, reduction of priority of fault correction, test difficulty, increase cost and inferior components. Fault-tolerant systems are typically based on the concept of redundancy [14].

Redundancy is the provision of functional capabilities that would be unnecessary in a fault-free environment [6]. It is a fault tolerant measure to reduce failures in a network. This can consist of backup components which automatically "kick in" should one component fail. The idea of incorporating redundancy in order to improve the reliability of a system was pioneered by John von Neumann in the 1950s [12].

C. Related Work

A distributed mechanism with two approaches to routing and load balancing in communications network was worked upon by Vaidyanathan, Callele and McCrosky [11]. They provided an overview of requirements to be considered in designing a fault tolerant communication network. The underlying concepts of packets, packet routing, and fault tolerance were introduced. They defined packets as information grouped together into packages to be passed from one device to another. The novel components contributed by [11] are the design of new packet routing algorithms that controls the actions of the individual routers. The performance of these networks degrades gently as these faults occur and this is a highly attractive feature for any application where it is very difficult to perform repairs.

Atayero, Alatishe and Iruemi [1] presented in their paper a simulation of an OPNET modeller architecture algorithm for designing and implementing the Local Area Networks (LAN) with its performance under ever increasing network traffic, and how this is affected by various network metrics such as latency and end-to-end delay. It was observed that the sent packet across the network at a particular point in time was equivalent to the received packets. This implies that there was neither packet loss nor any significant collision on the network at that particular instant. Simulation results suggest that the delay on the Ethernet network is less when only switch is used compared to when hubs and switch or only hub is used.

Medard and Lumetta [8] introduced a fault detection and recovery mechanism using optical encoding schemes. They used interconnected ring topology which was able to achieve this but due to the cost and extensibility, mesh-based architecture is more promising. Covering mesh topologies with ring (i.e providing both mesh topologies and distributed, ring-based restoration) was also used in order to enable recovery over mesh topologies. A fault tolerant redundancy process was also considered.

Ganesan and Girija [2] presented in their paper how the campus area network can be established optimally using wired and wireless technology. When it comes to wired network, speed, reliability and security are its major strengths while implementation, maintenance and overall cost are its limitations. But when it comes to wireless network, implementation, mobility and overall cost are its strengths while performance and security still needs some improvement. Wireless LAN has redefined what it means to be connected. It has stretched the boundaries of the local area computer network. It makes an infrastructure as dynamic as it needs to be.

Kotz and Essien [5] conducted the largest-ever trace-based study of wireless LAN users, in an effort to understand patterns of activity in the network. They analysed the campus-wide wireless network and found that many wireless cards are extremely aggressive when associating with access points, leading to a large number of short "sessions" and a high degree of roaming within sessions. These extra-subnet roams often occurs when the user is stationary, leading to failures of IP traffic. They discovered that Cross-subnet roams were an especial problem, because they broke IP connections, thus indicating the need for solutions that avoid or accommodate such roams.

Tipper et al [10] presented in their paper various survivability strategies of wireless access network. They discussed the effects of failures and survivability issues in PCS networks with emphasis on the unique difficulties presented by user mobility and the wireless channel environment. A simulation model to study a variety of failure scenarios on a PCS network was described, and the results shows that user mobility significantly worsens network performance after failures, as disconnected users move among adjacent cells and attempt to reconnect to the network. A multilayer framework for the study of PCS network survivability was presented. Metrics for quantifying network survivability were identified at each layer. Possible survivability strategies and restoration techniques for each layer in the framework were also discussed.

III. Designing A Fault Tolerance Campus Area Network (CAN) With Opnet

In the designing of a fault-tolerant campus area network, there are many design models, such as hierarchical model, which exists and that can be followed to simplify the design process. Hierarchical model simplifies the design through the methodology of breaking the network into three main components that are: access network, distribution network (convergence/aggregation network) and core network (backbone network) which simplify, make it smaller and more manageable [9]. The components of the campus area network needed includes: routers, servers, firewall, switches, cables, access point, wireless connection, PCs, trunks and telephone. The figure 1 below shows the methodology adopted in simulating the CAN in the OPNET architecture algorithmic diagram with a slight modification [1].

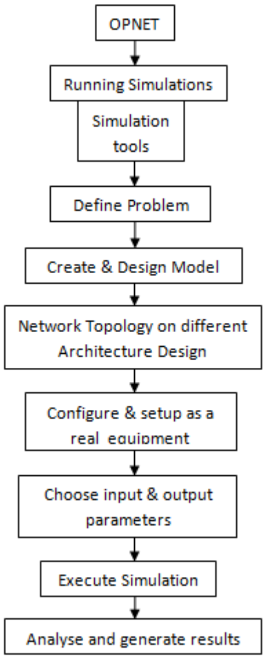

OPNET
Running Simulations
Simulation tools
Define Problem
Create & Design Model
Network Topology on different Architecture Design
Configure & setup as a real equipment
Choose input & output parameters
Execute Simulation
Analyse and generate results

Figure 1: The OPNET Architecture [1]

Figure 2 and 3 shows the design diagram of both the non-fault tolerant and fault tolerant campus area network respectively. The fault tolerance implemented is the backup server in case of network failure.

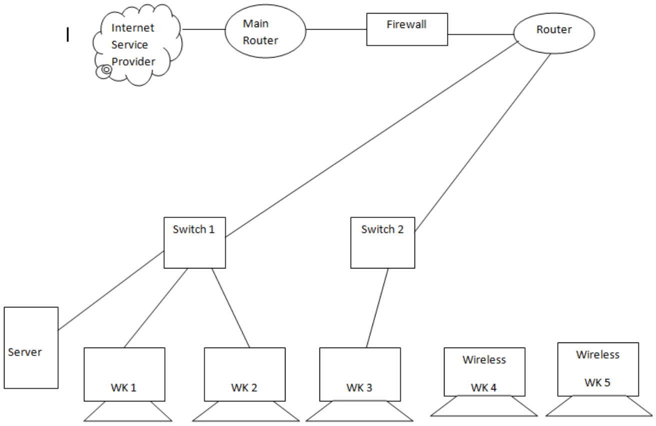


Figure 2: Non-fault tolerant campus area network

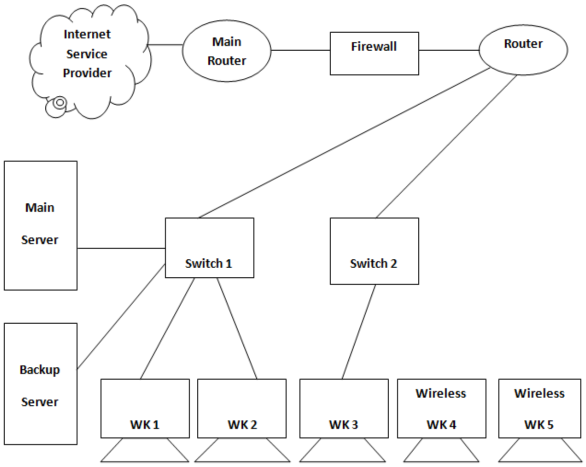


Figure 3: Fault tolerant campus area network

IV. IMPLEMENTATION AND RESULT

The figure 4 shows the simulated design of the headquarter subnet where the fault tolerant campus area network was implemented.

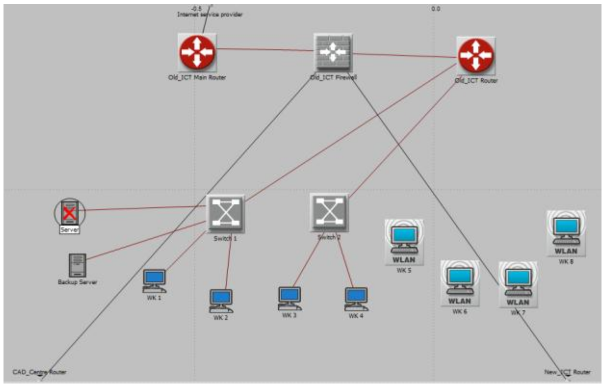


Figure 4: Headquarter (LAUTECH old ICT) subnet of the fault tolerant campus area network

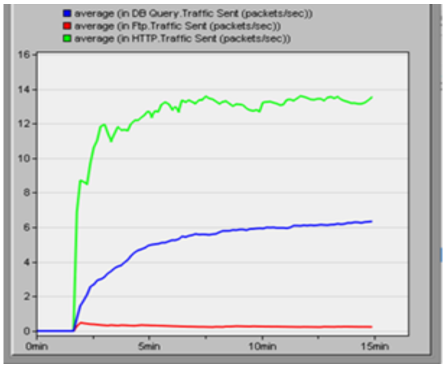
The standard measures/ metrics that is used for testing and evaluating the reliability of the campus area network are Ethernet Delay, Wireless LAN Delay, Wireless LAN Load, Throughput, Traffic Sent, Traffic Received and Response Time.

In this case, three different applications were used: FTP, Database and Http. The model is measured for its performance by running data, file and web traffic; hence the average delay, throughput, load, and received traffic are the performance metrics used in this work. All graphs show a combination of the 2 scenarios.

TABLE I:
SIMULATION EVALUATION RESULT

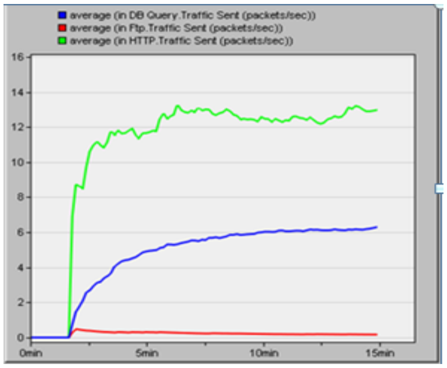
	Parameters	Non fault tolerant scenario	Fault tolerant scenario
1	Ethernet Delay	0.00055 sec	0.00055 sec
2	Wireless LAN Delay	0.00035 sec	0.00036 sec
3	Wireless LAN Load	140,000 bits/sec	180,000bits/sec
4	Wireless LAN Throughput	140,000 bits/sec	180,000bits/sec
5	Traffic Sent DB Server FTP Server HTTPServer	 13.5packets/sec 6.2 packets/sec 0.5 packets/sec	 13.5 packets/sec 6.2 packets/sec 0.5 packets/sec
6	Traffic Received DB Server FTP Server HTTP Server	 3.5 packets/sec 2.2 packets/sec 0.5 packets/sec	 3.5 packets/sec 2.2 packets/sec 0.5 packets/sec
7	DB server Response time	0.97 sec	0.52 sec
8	FTP server Response time	3.4 sec	2.4 sec
9	HTTP server Response time	3.5 sec	3 sec

In this section, both non fault tolerant and fault tolerant campus area network results are compared in order to evaluate their efficiency. Here, we have kept the same settings & scenario for recording measurements for both non-fault tolerant and fault tolerant campus area network. Our investigations reveal that additional server is useful in decreasing the campus area network downtime, while in the process of requests. Thus, it is evident that the use of single server is not advisable in most cases. The table 1 shows the simulation evaluation result.



(a)

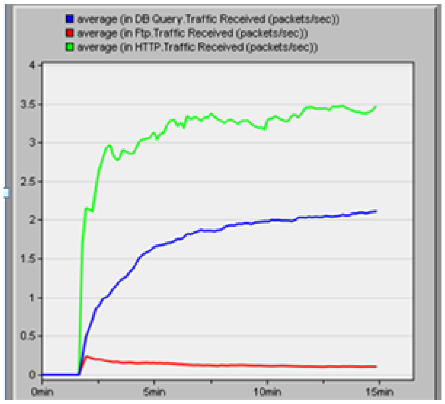
Fig 5a: Fault tolerant average server traffic sent (packets/sec)



(b)

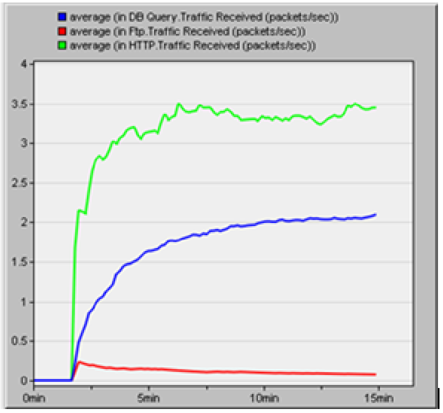
Fig 5b: Non Fault tolerant average traffic sent (packets/sec)

Figure 5 represents the average number of packets per second submitted to the transport layer by the FTP, Http and Database Application respectively for both non fault tolerance and fault tolerance scenario in the node.



(a)

Fig 6a: Fault tolerant average server traffic received (packets/sec)



(b)

Fig 6b: Non Fault tolerant average server traffic received (packets/sec)

Figure 6 represents the average packets per second forwarded to the FTP, Http and Database Application respectively for both non fault tolerance and fault tolerance scenario by the transport layer in the node. From the figure and , it can be inferred from the graph that the server traffic sent for FTP, Http and Database is equal to the traffic received but at different time interval thus there was no packet loss.

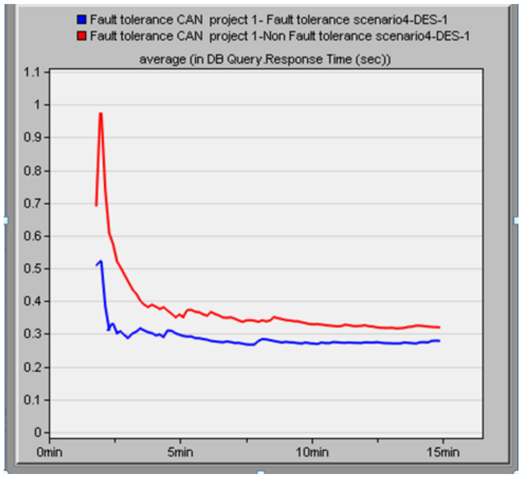


Figure 7: Comparison of both fault tolerant and non fault tolerant DB server response time (sec)

Figure 7 represents the time elapsed between sending a request and receiving the response packet. It is measured from the time a client application sends a request to the server to the time it receives a response packet. From the figure, it can be deduced that there is a considerable difference between the DB response time for fault tolerance campus area network and DB response time for non fault tolerance campus area network with respect to the backup server. The DB response time with the backup server is found to be 0.52 sec initially and then reduces gradually to a constant value of 0.28 sec and without the server it varies from 0.97 to 0.33 sec. The response time with backup server is much better than the response time without the backup server.

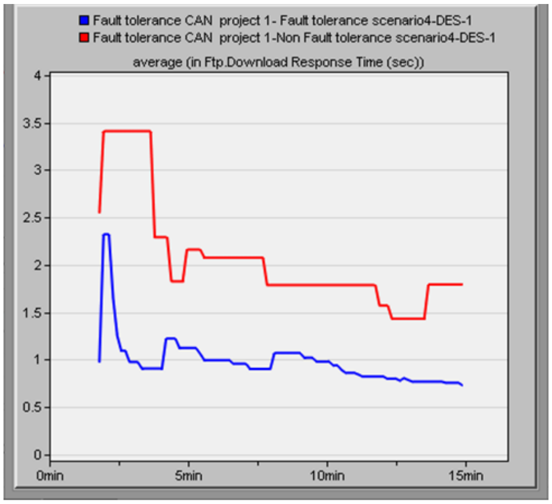


Figure 8: Comparison of both fault tolerant and non fault tolerant FTP server response time (sec)

From the figure 8, it can be deduced that there is a considerable difference between the FTP response time for fault tolerance campus area network and FTP response time for non fault tolerance campus area network with respect to the backup server. The FTP response time with the backup server is found to be 2.4 sec initially and then reduces gradually to a constant value of 0.75 sec and without the server it varies from 3.4 to 1.8 sec. The response time with backup server is much better than the response time without the backup server.

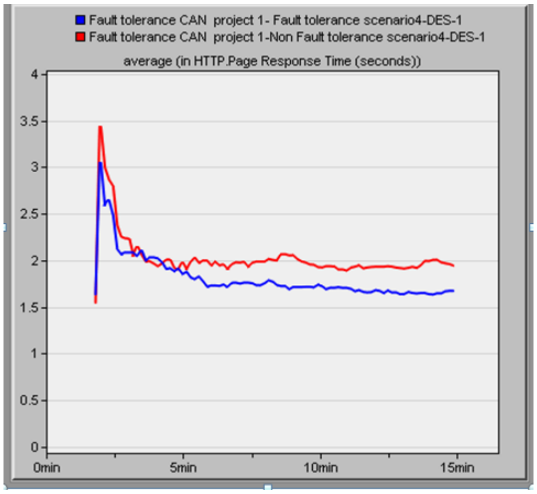


Figure 9: Comparison of both fault tolerant and non fault tolerant HTTP server response time (sec)

From the figure 9, it can be deduced that there is a considerable difference between the HTTP response time for fault tolerance campus area network and HTTP response time for non fault tolerance campus area network with respect to the backup server. The HTTP response time with the backup server is found to be 3 sec initially and then reduces gradually to a constant value of 1.7 sec and without the server it varies from 3.5 to 2 sec. The response time with backup server is much better than the response time without the backup server.

In conclusion, from the result it is advisable to implement a fault tolerance measure in case of network failure e.g. backup server. Also there is more efficiency in the performance of the fault tolerant campus area network than the non fault tolerant campus area network.

V. CONCLUSION

The OPNET Modeler tool was used as the simulation software for simulating a fault tolerant campus area network.

The usage of this software has thrown more light on the fact that real life network scenario can be imitated successfully. In the section IV above, it can be seen that there was variably no packet loss i.e. sent packet across the network at a particular point in time was equivalent to the received packets as indicated by the graphs depicting the traffic sent and received in the 3 servers. Furthermore, due to the variation in the response time of both scenario it was deduced that the network delay in the fault tolerant campus area network was lesser than the non fault tolerant campus area network because of the backup server used.

REFERENCES

[1] Atayero, A.A., Alatishe, A.S. and Iruemi, J.O. (2012). Modeling and Simulation of a University LAN in OPNET Modeller Environment. International Journal of Emerging Technology and Advanced Engineering (IJETAE) vol 2, No 2, Pg 1-4.

[2] Ganesan, R. And Girija, C.R. (2013). Establishing Campus Networking (CAN) using a combination of wireless and wired connectivity-An Optimum Solution. International Journal of Science, Environment and Technology, vol 2, No 3, pg 478-486.

[3] Johnson, B. W. (1984). Fault-Tolerant Microprocessor-Based Systems. IEEE Micro, vol. 4, no. 6, pg 6–21.

[4] Jones Stephen and Kovacs Ron (2003). Introduction to communications technologies: A guide for non-engineers. Editors ISBN 0-8493-1266-3.

[5] Kotz David, Essien Kobby (2002). Analysis of a campus-wide wireless network. In MOBICOM’02, Pg 107-118.

[6] Laprie, J. C. (1985). Dependable Computing and Fault Tolerance: Concepts and Terminology. Proceedings of 15th International Symposium on Fault-Tolerant Computing (FTSC-15), pg 2–11.

[7] Manish Marwah, Shivakant Mishra and Christof Fetzer (2003). TCP Server Fault Tolerance Using Connection Migration to a Backup Server. In Proceedings of IEEE International Conference on Dependable Systems and Networks.

[8] Médard, M., & Lumetta, S. S. (2003). Network reliability and fault tolerance. Encyclopedia of Telecommunications.

[9] Tavares Sofia Agueda (2011). Network Architecture for university campus network. Degree of Master of Engineering Programme, Pg 1-8, 96. http://net.educause.edu/ir/library/pdf/ecar_so/ers/ERS0502/ekf0502.pdf.

[10] Tipper David, Dahlberg Teresa, Shin Hyundoo and Charnsripinyo Charlermpol (2012). Providing fault tolerance in wireless access networks. IEEE Communications Magazine Pg 58-64.

[11] Vaidyanathan, C. S, Callele, D. and McCrosky, C.(1993). An overview of a fault tolerant communication network. In proceeding of: WESCANEX 93. ‘Communications, computers and power in the modern environment.’ Conference proceedings, IEEE.

[12] Von Neumann, J. (1956). “Probabilistic Logics and Synthesis of Reliable Organisms from Unreliable Components”, in Automata Studies, eds. C. Shannon and J. McCarthy, Princeton University Press, pg 43–98.

[13] Wikipedia (2015a) Computer Network. Retrieved April 18, 2015, from http://en.wikipedia.org/wiki/computer network.

[14] Wikipedia (2015e) Fault tolerance. Retrieved April 18, 2015, from http://en.wikipedia.org/wiki/fault tolerance.

[15] Wikipedia (2015d) Router (Computing). Retrieved April 18, 2015, from http://en.wikipedia.org/wiki/Router.

[16] Wikipedia (2015f) Server. Retrieved April 13, 2015, from http://en.wikipedia.org/wiki/Server.

[17] Wikipedia (2015c) Switch. Retrieved April 13, 2015, from http://en.wikipedia.org/wiki/switch.