7659, England, UK
Journal of Information and Computing Science
Vol. 2, No. 3, 2007, *pp*. 228-234

Towards Designing and Implementing an Expert Network To Manage the Computer Communication Networks

Ibrahiem M. M. El Emary [1 +], Salam A. Najim[1], and Musbah Aqel [2]

[1] Faculty of Engineering, Al Ahliya Amman University

[2] Faculty of Engineering, Applied Science University, Amman-Jordan

(*Received 12 October 2006, accepted 30 January 2007*)

Abstract. This paper aimed to design and implement a new hybrid intelligent technique called expert network. The main purpose of this expert network is to use it as a management tool assist network administrator in his job regarding the monitoring process of network performance. Here, the management task that uses this expert network was how to detect a fault (i.e. fault detection) that may be occurred on a certain part of the network either on hardware or software where this fault can affect on the performance of the computer network as: delay, throughput, utilization. Our proposed expert network depends on using both feed forward Neural Network and CBR. The Multilayer feed forward NN is used to predict the upcoming or the next values of the network major parameters (as load, delay, throughput…etc), then this predicted value will produces a residual signal used for comparing the actual with the desired network parameter. According to the comparison results, the Expert Network will alter the network administrator with certain alarm if the difference between the actual and desired network parameter outside a certain range (threshold value, knowing that this threshold is suggested by the network administrator). Accordingly, the network administrator begins to take some action towards repairing the fault component. Efficiency of the proposed expert-network was examined through investigating the time of detecting a fault. This time was too short relative to the running time of the computer network.

Keywords: ES, CBR, ANN, NMS, SNMP and TCP.

1. Introduction

In a computer communication network, there are several problems that may occur due to different reasons like failing of certain hardware component or some type of bugs in the various software applications. Subsequently, these problems can affect the network performance and communication between users. The occurred problems should be detected and solved quickly as possible before users scan it. Network administrator comes to know about these problems that occur in the network before its effect reaches the users. Here, the importance of this paper appears since we build an expert network shell to use it for solving a real world problem (i.e. monitoring the computer network performance to detect any deviation than a certain threshold value).

The proposed expert network gives a reasonable results relating to predict the occurred problem in the computer network in an optimum time which facilitate the job of the network administrator.

A simulation for a Gigabit Ethernet network is implemented using OPNET IT GRU. The simulated scenarios where developed to simulate a heavy web browsing network and to implement an artificial fail on the web server to check the effectiveness of the expert network in view point of detecting the fails that occur on the servers. The output results show that the implemented expert network gives an excellent indication for detecting the faults that occurred on the computer network. The expert network is trained to the ideal load, and then it is tested with a network fault to be able for detecting the actual fault when it occurs on the network. Also, the implemented expert network provides the network administrator the ability to detect an error when it lies with a certain level (i.e. threshold). This paper is organized from five sections. Section two deals with fault identification process. In section three, we describe the proposed hybrid AI system for fault diagnosis. Section four was dedicated for the simulated results. Conclusion and future works close the paper in section five.

+ *E-mail address:* doctor_ebrahim@yahoo.com&drsalamadil@yahoo.com

2. Fault Identification Process of the Computer Network

Normally, the first step of fault diagnosis of the computer network concerned with the filtering and correlation of alarms. The second step involves further analysis and identification of the exact cause of alarms, or the fault. This process is an iterative one where alarm data are analyzed and decisions are made whether more data should be gathered, a finer grained analysis should be executed, or problem solving should be performed. Gathering more data deals with sending tests to the network elements or requesting network performance data. Problem solving requires expert knowledge which consist of domain knowledge (i.e., knowledge about the network elements and topology, and physical faults), and meta-knowledge (i.e., knowledge about how to diagnose faults) [3, 5, 7]. A symbolic AI technology such as ES or CBR is ideal for achieving this functionality. CBR systems are more robust than ES since it can: handle new and changing data trough their ability of identifying the goodness of a proposed solution based on simulations and/or pervious cases from the case library. CBR can be thought of as such as an expert that applies previous experiences stored as cases in a case library. Thus, the problem solving process becomes on the recalling old experiences and interpreting the new situation in terms of those old experiences [1, 6].

CBR problem solving can be depicted as a five-step process as shown in Figure1: retrieval, interpretation and adaptation, evaluation and repair, implementation, and evaluation and learning. The first step is retrieving cases that best match the current situation or case. Thus it is crucial to use an appropriate indexing method, such as decision trees or nearest neighbor matching. Once a case is retrieved, it must be interpreted and then adapted. The interpretation process is a simple comparison between the retrieved cases and the current case. Adaptation is a complicated, domain-dependent process that uses rules to adapt the current case to the problem situation and propose an initial solution, based on the similarities and differences [2, 4].

The next step is an evaluation and repair cycle where the proposed solution is evaluated through comparisons to cases with similar solutions or through simulation, and the solution is modified accordingly. After the CBR system has found its best solution, the solution is implemented and the results are evaluated. The resulting evaluation, solution steps, and problem context are entered into a new case, which is then indexed into the case library, allowing the system to learn.

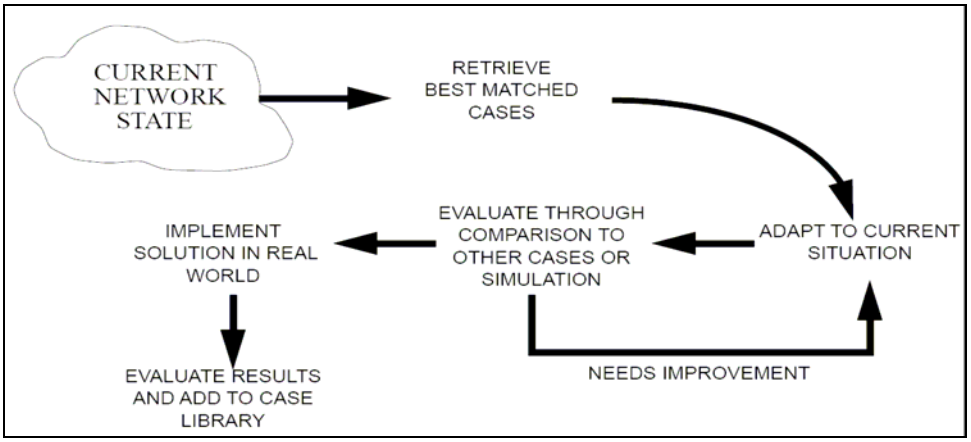


Fig. 1: The Case-Based Reasoning Process

3. The Proposed Hybrid AI system for Fault Diagnosis

In this section, we present a complete description for a hybrid AI system which is a good candidate for a fault diagnosis. This hybrid AI system uses both probabilistic and symbolic problem solving AI techniques. Figure 2 shows how NN/CBR system can be used for fault diagnoses.

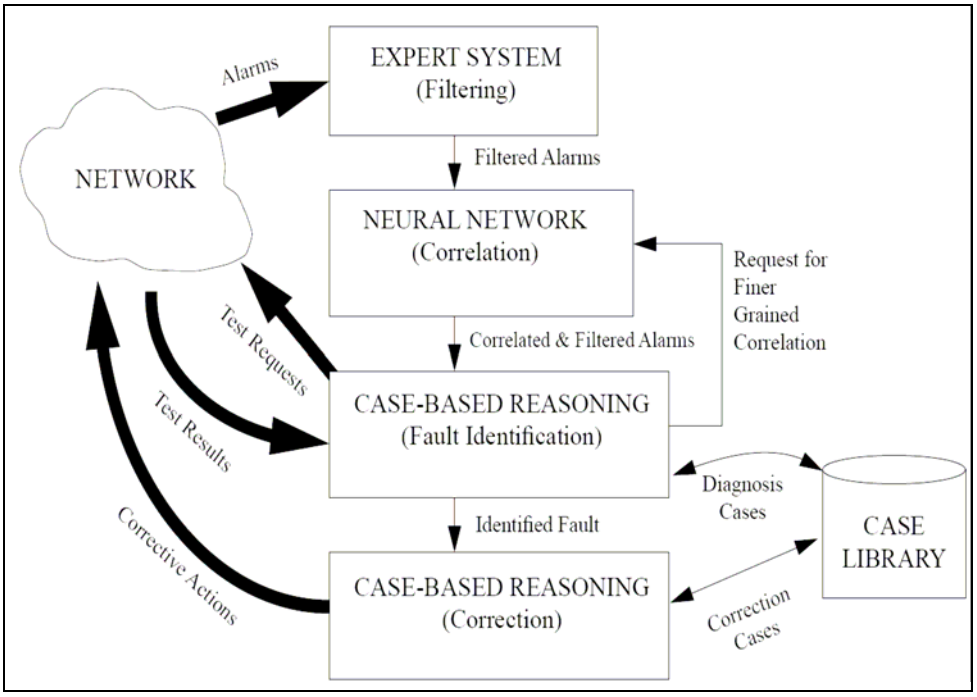


Fig. 2: A hybrid AI System for Automated Network Fault Management

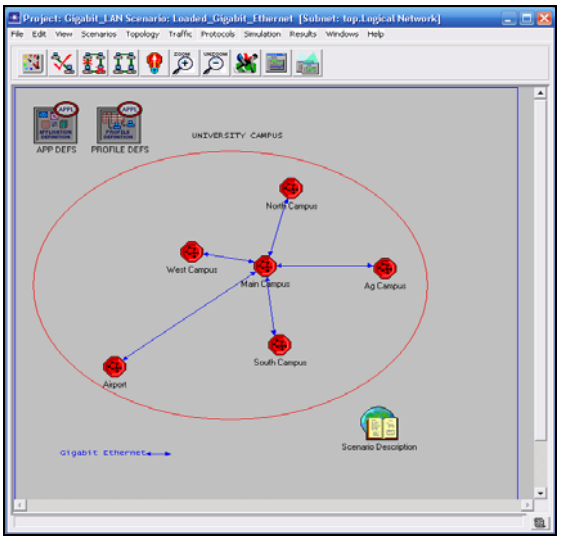
From Figure 2, we can see that the operation process of the hybrid AI system will be as follows: Network alarms are fed into an ES that filters the alarms through compression, count, suppression, and generalization. The pattern recognizing ability of neural networks can be used for correlating alarms and recognizing alarm patterns. Filtered alarms serve as input into a NN, trained to recognize common fault categories. The output consists of the most likely fault types, protocol types, and geographic area of the fault. This analysis is then fed into a case-based reasoning system where testing and further analysis is performed. The CBR system can make decisions based on past problem solving experiences, whether to gather more data, problem solve, or run the data through the same NN or a different NN that was trained on a finer grained set of alarms.

The first step in developing a NN component for the fault management system is to train the NN. Neural networks learn to recognize patterns by generalizing from many examples. Supervised training should be used where many examples of similar and different faults are fed into the network with known outcomes. Once the network is trained, it can be integrated with the CBR system after receiving the filtered and correlated alarms, the CBR system attempts to identify what information would further the diagnosis of the fault. The CBR system needs to decide on its own what additional tests need to be made on the network elements and what granularity of NN to use to further analyze the data. To achieve this end, the CBR library is searched for the cases that most resemble the current situation. Once the cases are found, similarities and differences are discovered and a new solution is proposed. This continues in an iterative manner until the fault is identified. Once a fault is identified, the whole problem solving episode should be analyzed. The value of the tests (i.e., useful, not useful), the steps taken, any circuitous paths or dead-ends taken, and the success of the analysis should be stored into a new case. This information, in addition to contextual information comprises a new case which is then indexed into the case library. In this manner, the fault management system is able to learn from its successes and failures.

4. Simulation Results of the Developed Expert Network System

This section is concerned with: illustrating the structure of the examined computer network, simulation of such examined network with OPNET IT GURU Academic Edition to calculate the expected performance of the network, and to provide training set for the developed expert network using Visual Studio (Visual Basic). First of all, Figure 3 illustrates university 1000BaseX local area network that is dispatched over multiple areas (campuses) and each campus has its own defined load and request from the main campus. The main campus of the university contains two kinds of servers one of them is web-browsing server and the second one is e-mail server and each one of these two servers has one redundant server for backup purposes

if any fail has occurred as shown in Figure 4.



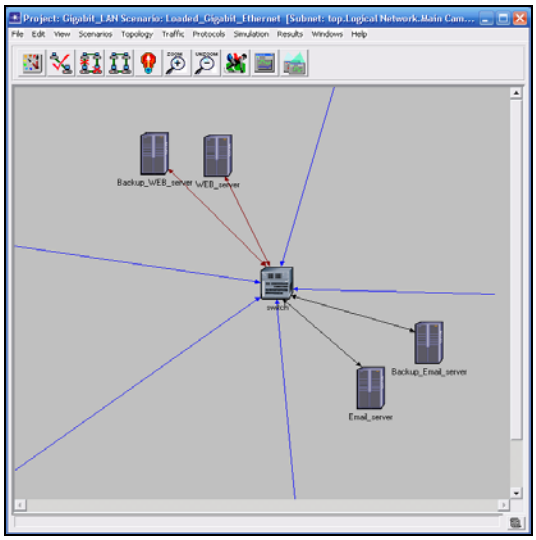


Fig. 3: Implemented Net. Using OPNET Fig. 4 Main Campus Network elements

The other university campus like north, south, and west having two network elements to simulate a heavy web browsing station and a heavy e-mail load as shown in Fig.5.

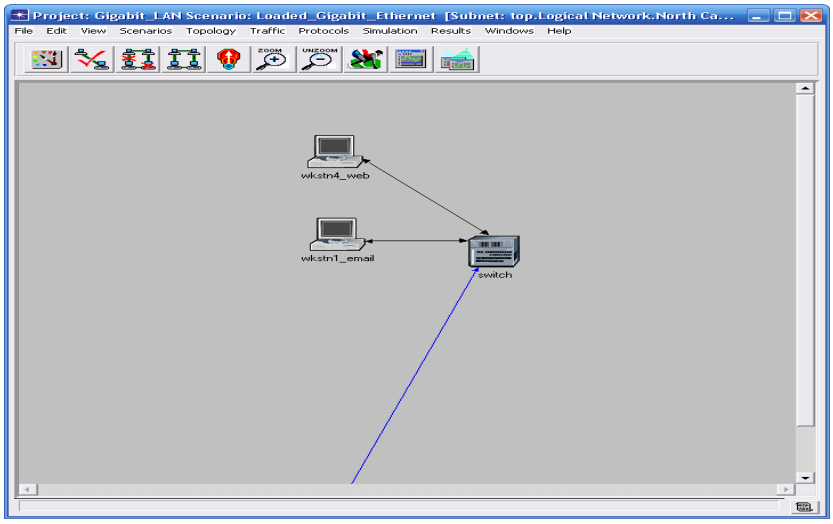


Fig. 5: North campus Network elements

Case1: Running the network in normal operation.

After building each network components by using OPNET editors and specifying each element attribute, we run the simulator for three hours in order to get or obtain the desired network parameters such as: delay, load, and request per server, as shown in Figure 6.

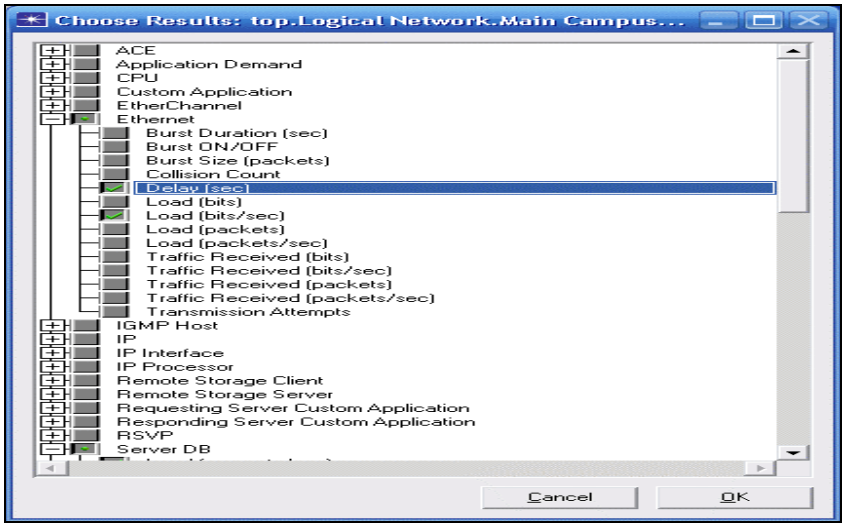


Fig.6: Choosing Individual results

After running the simulator for three hours (time of simulation), the desired parameters are stored and displayed by the OPNET as shown in Figure 7, which shows the relation between the server load (requests) against the simulation time in sec.

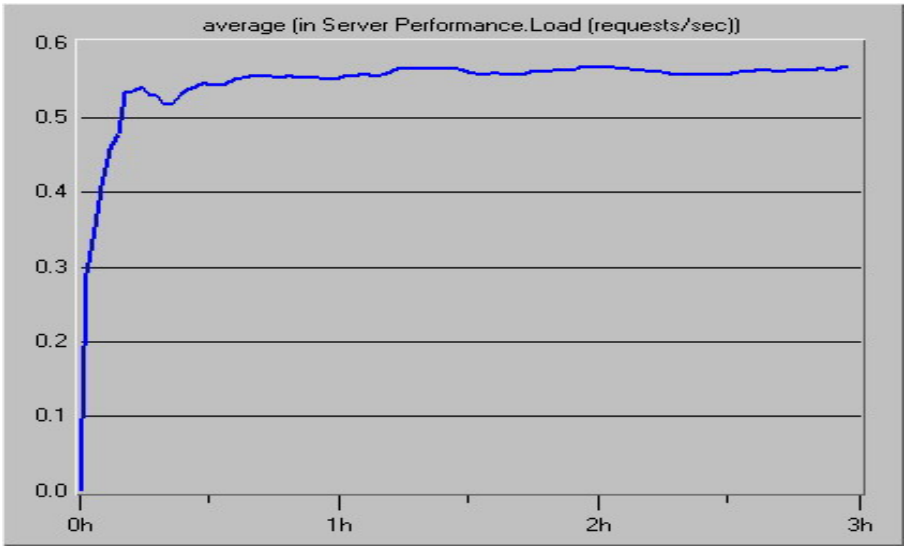


Fig.7 Web-Server Load (request/sec)

Case 2: Running the network with an artificial server fail

OPNET utilities provide the ability to simulate a fail of one or more services during the time of simulation. This utility has been applied to simulate a fail scenario of backup Web-Server during the time of simulation, to achieve statistical results of what happens to the main server when such fail occurs in the networks as shown in Figure 8.

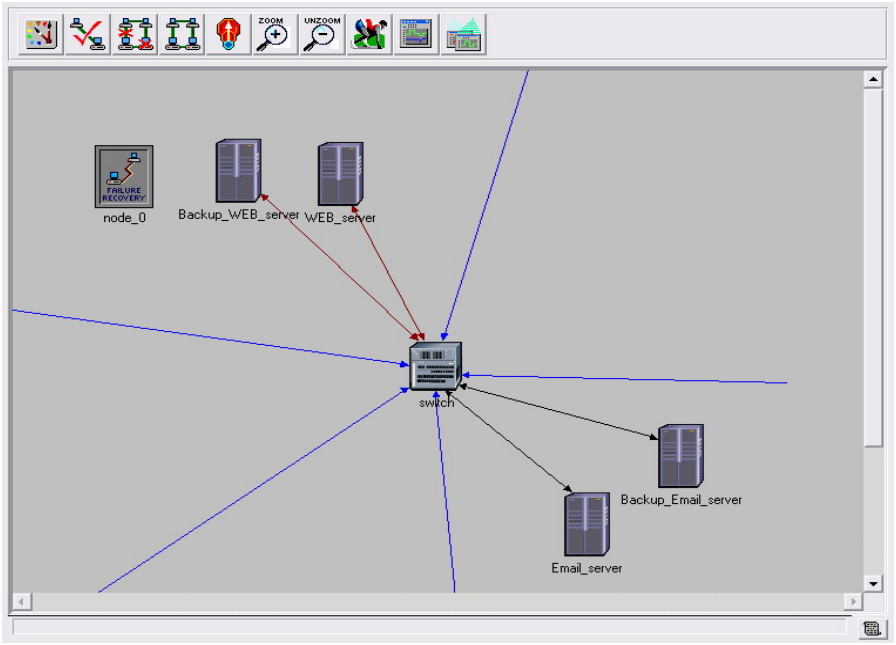


Fig.8. Backup Web-Server Fail Nodes

Figure 9. shows the performance parameter represented by load (request/sec) for the above two cases.

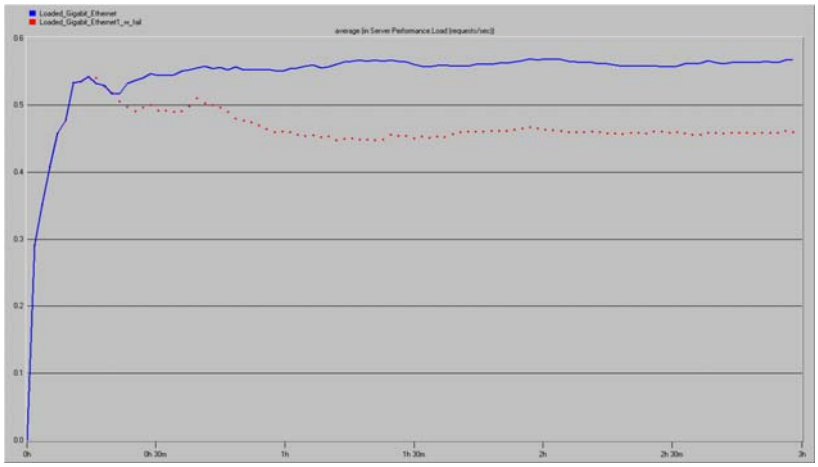


Fig.9: Web Server Load (requests/sec) for Ideal/Failed

From Figure 9, it is shown that there is a clear difference (approx 5%) in the network performance parameter (request/sec) for both cases: the normal case and the failure case. This difference appears after few seconds from the fail time (1000 seconds). After that, the role of expert network begins to appear to detect the failure of the web-server. The expert network has been trained previously using the training set obtained from case 1. Expert network tries to predict the desired performance parameters and compare it with the actual value for the network performance parameter. The difference between the actual and desired network parameter will produce a residual signal which alter the network administrator if it lies outside the threshold range. Figure 10 shows such a case of detecting a failure of browsing server on the network at time (1224sec).

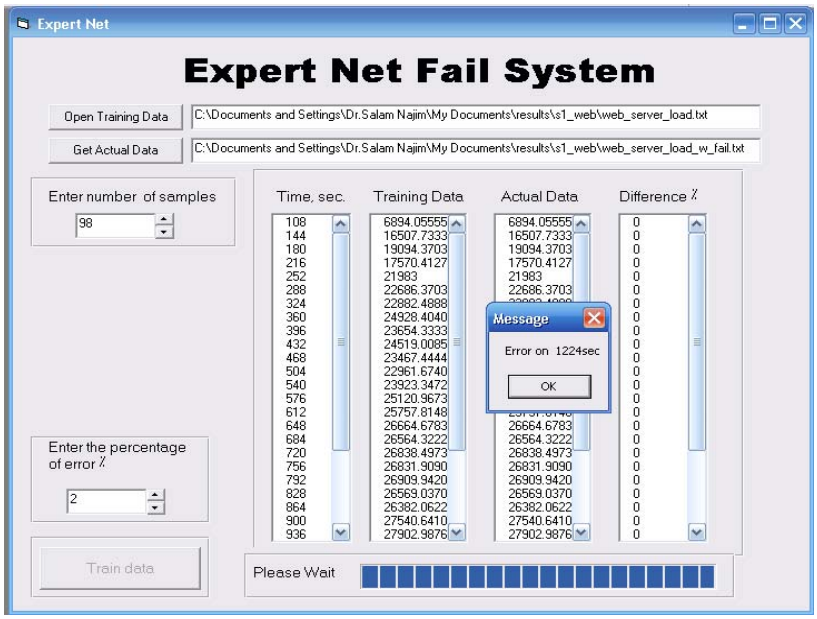


Fig.10: Expert Network System Detects a (2%) Error

5. Conclusion and Future Works

In this paper, we developed a hybrid AI system (Expert Network) in order to use it in monitoring the performance of a computer network and alter the network administrator with an alarm if there is a pre-determined deviation between the observed network parameter and the desired one. This hybrid AI system is structured from two sections, the first one is neural network and the second one is case based reasoning system. Our work is executed through using OPNET IT GURU and this simulation tools is used to analyze and evaluate the performance of the computer network. Through this simulation tool, we build an ideal computer network then we take the same of simulated results on one of its component which was a web server. After building such a network, we make two experiments, one of them describes the normal operation of the network without any failure at any component, and here we take some results related to the load in bit/sec on the web server. The second case represents taking the same results of load on the web server but in this case an artificial failure in the redundant web server. These results are taken and stored in an excel sheet, then it is considers as an input to the expert network system which is built using visual basic. This built software has a capability of learning, training, comparing between actual results and desired results. Then, finally altering the administrator with an alarm message if a difference between the actual and desiered results lies outside the range of certain pre-determines threshold value. The efficiency of this intelligent hybrid AI system was too high science it detects any occurred faults within too limited (very small) intervals. As a future work to be executed by others, we suggest to apply the proposed Expert Network in managing other types of network like one that wireless technology. Also, we would like to apply the proposed Expert Network in another network management tasks like: security, reconfiguration, and accounting.

6. References

[1] A. Tanenbaum. *Computer Networks***,** Fifth Edition. McGraw-Hill, 2003.

[2] D. Medhi. *Network Reliability and Fault Tolerance***,** Department of Computer Networking, University of Missouri-Kansas City, 1999.

[3] H. Li, J.S. Baras. *Intelligent Distributed Fault Management for Communication Networks***.** Center for Satellite and Hybrid Communication Networks, Department of Electrical and Computer Engineering, University of Maryland, 2002.

[4] R. Cullough Barzilay, O. D. Rambow. *A New Approach to Expert System Explanations*. New York: Times books. 2001.

[5] D. Medhi. Fault Management in Communication Networks, *Special Issue of J. Networks and System Management*. 1997, **5**(2).

[6] E**.** Rich. *Artificial Intelligence*, Second Edition. McGraw Hill. 1995.

[7] W. Khan Gurer, R. I. Ogier. *An Artificial Intelligence Approach to Network Fault Management*. SRI International, 333 Ravenswood Ave, Menlo Park, CA 94025 USA, 2001.