ExNet: An Intelligent Network Management System

Yoonhee Kim*, and Salim Hariri**

* Department of Electrical Engineering and Computer Science
Syracuse University, Syracuse, New York 13244-4100
yhkim@cat.syr.edu
**Department of Electrical and Computer Engineering
The University of Arizona, Tucson, Arizona 85721-0104
hariri@ece.arizona.edu

Abstract: In this paper, we investigate the use of artificial Intelligence techniques in the management of large-scale high-speed networks. We present a design of an intelligent network management system (ExNet) that efficiently and effectively monitors and controls the resources of a large scale computer network. The ExNet provides World Wide Web graphical user interface. We have implemented a rule-based prototype of ExNet and have incorporated ExNet modules with IBM NetView network management system.

1. Introduction

The current network connectivity has increased over the last decade. An explosive growth in the number of computers and the need to share information has triggered the development of computer networks. Moreover today's application demand a wide variety of services from the network. There are many different classes of traffic on the network and each traffic type has different quality of service requirements. All these applications assume that the network will provide them with the best possible service it has to offer. To provide efficient service to each application, the network must be operating efficiently at all times. If there are any network failures, they must be rectified immediately. Network failures cause various losses such as efficiency and productivity, and eventually manifest as monetary losses to both the clients and network access providers.

Current network management systems help a network operator to detect and diagnose the problems in a network. However as the complexity of today's networks increases, there is a greater demand for the network to be self managing. A self-managing system should be capable of automatically detecting and diagnosing the network problems. A capability to repair ordinary faults would also be desirable in such systems. In this paper, we present an architecture for an intelligent network management system, called ExNet. We use expert system technology to add intelligence capability to commercially available network management systems.

The organization of the paper is as follows. In Section 2, we present the ExNet architecture. The component modules of the system are described here. Section 3 covers the implementation aspects of ExNet prototype. The various subsystems we have developed are described. We conclude the paper in Section 4.

2. Architecture of an Intelligent Network Management System

Current network management systems are in general complex. They provide a lot of information and many different services. This has complicated the development of network management applications that utilize the services of these network management systems. The main goal of our effort is to apply expert system technology to reduce the severity of this problem and allow less experienced network operators to efficiently manage a large and complex computer network. Therefore we need a management system which will analyze the network and its services, and properly manage its performance. It must be intelligent enough to diagnose most of the faults as they occur and have the capability to suggest corrective actions. Despite this complexity, the system should be easy enough to use. For this we use a web-based technology to provide the required user interface [BEGG 94]. The use of web-based technology enables us to provide a better graphical user interface to ExNet. Moreover this allows users to have a global access to ExNet resources and information any time and from anywhere with Internet access.

2.1 General Architecture

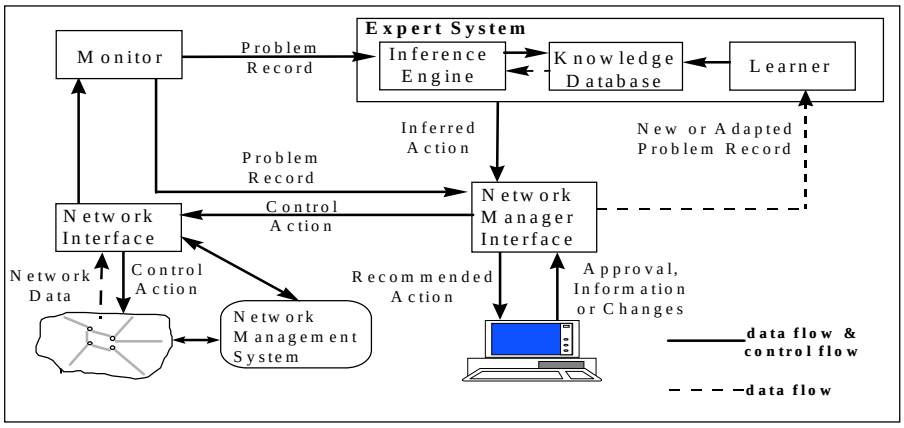

Figure 1 General Architecture of ExNet

The ExNet architecture shown in Figure 1 is composed of four modules.: Monitor Module, the Network Interface Module, the Network Manager Interface Module, and the Expert System Module. The "Monitor" is involved in monitoring the network. It keeps track of the changes that are occurring within the network. Further if it deems any event to be critical to the performance of the network, it reports the event to the expert system module. The "Expert System" performs an analysis of the situation provided to it by the monitor. Then based on its internal reasoning, it decides a specific course of action to alleviate the given problem. In this paper we investigate the use and applicability of both rule-based and case-based reasoning expert systems. The "Network Interface" is responsible for transferring information about events as they happen in the network. Also it will be responsible for implementing the strategies recommended by the expert system, and approved by the human manager. The "Network Manager Interface" will be the contact point between the human administrator and the ExNet system. This interface will present the information about the network, such as the current status of various nodes and links to the manager. It will also present the solution that the expert system module has come up with to the human manager. In what follows, we discuss our approach to implement these modules in more detail.

The Monitor Module

The Monitor module obtains and stores all the relevant information about the network. The information about the network will be obtained from a network management system. This information will come to the monitor via the network interface module. The information provided by the Network Interface will consist of either general events or special traps. The general events are reports on values of the observable parameters across the managed network. These reports are generated synchronously by polling each node. The special traps are generated by nodes when some extraordinary activity takes place in the network. These events include instances of network nodes going down or coming up, and a link either failing or a new link being formed. These are generated asynchronously.

After obtaining this information, the monitor makes a preliminary analysis to determine whether or not the given situation is abnormal. However it must be noted at this point that the Expert System is the final authority in determining if the current conditions are abnormal or not. For these cases all parameters are within normal ranges and do not represent any abnormal conditions. Since any network is expected to be performing normally most of the times, such a preliminary screening will serve to reduce the load on the Expert System, and consequently enhance the performance of the overall system. The cases which indicate slightly the possibility of an abnormal conditions in the network, will be passed on to the Expert System for further analysis.

The Network Interface Module

The primary tasks for this module is to provide a seamless communication between any network management system (NMS) and the Monitor and Network Manager Interface modules. Consequently, this module transfers network data from the NMS to the monitor and also receives control instructions from the Network Manager Interface and delivers

it to the NMS. The expert system may request additional information regarding the history of the system to reach a decision on the current problem. Moreover the network manager interface may obtain any information requested by the human network administrator. The events and traps, described earlier, are stored by the NMS in the "Event and Trap Logs." The information about the topology is stored in the "Topology Databases." The network interface can query these logs and databases for previously recorded information about the network. Apart from providing information to the other modules, the network interface is also responsible for implementing the actions recommended by the expert system and approved by the network administrator. The solutions that can be implemented will depend on the support provided by the NMS. It must be noted at this point that the network interface need not be entirely dependent on a network management system for functioning. The network interface can be implemented using the support provided by the operating system. Operating systems, such as UNIX, provide extensive utilities that can be used for network monitoring and management.

The Network Manager Interface Module

The main function of this interface module is to provide the human managers information, in a user-friendly manner, about network states and conditions and the expert systems' recommended actions. Also it takes directives from the user and converts them into instructions for other modules of ExNet. For example the user might want to query about the state of some particular node of the network. The most important interaction between this interface and the user is when a solution to the problem is presented to the user. The interface must be able to present all the facts pertinent to the current problem and should also specify the solution in a lucid manner to the user. The user's job will be to provide a sanity check to the proposed solution. The user may question the solution itself or may suggest some modifications.

The Expert System Module

The expert system module will analyze each scenario presented by the monitor, identify various problem symptoms, diagnose faults, isolate probable causes, and generate solutions. We use two approaches to design the expert system: rule based and case based reasoning methods.

- *Rule Based Expert System*

  A generic rule-based expert system consists of a working memory, a rule-base, and a control procedure. The rule base contains all the rules about the problem domain. It is the knowledge base where the "expertise" of the system is stored. The knowledge base is represented as a set of rules. Rules are "if-then" structures. If certain criteria are met, then the system is to take certain action. The expert system's problem solving capabilities are directly dependent on the number of rules it has in its rule-base. By increasing the number of these rules, the expert system will be able to solve many and more complicated problems. The working memory contains the rules that are directly applicable to the given problem under consideration. These are the rules that need to be "fired" – executed since their "if" clauses have been satisfied. The system updates the working memory by asserting, modifying, or retracting working memory elements. For a network application, the working memory typically contains a representation of characteristics of the network related to the current problem, including topological and state information. The rule-base represents knowledge about what operations to perform when the network enters an undesirable state.

- *Case Based Reasoning System*

  A case-based reasoning system is a powerful mechanism for exploiting past experiences in planing and problem solving. A case-based reasoner solves problems by applying previously successful solutions rather than generating a new solution to the problem from a scratch. The advantage of this approach is the ability to use the large-grained knowledge representation of "cases" - the previously successful solutions - rather than finer-grained rules, hence potentially enhancing the real-time performance of the network management system. Figure 2 shows the case based expert system.

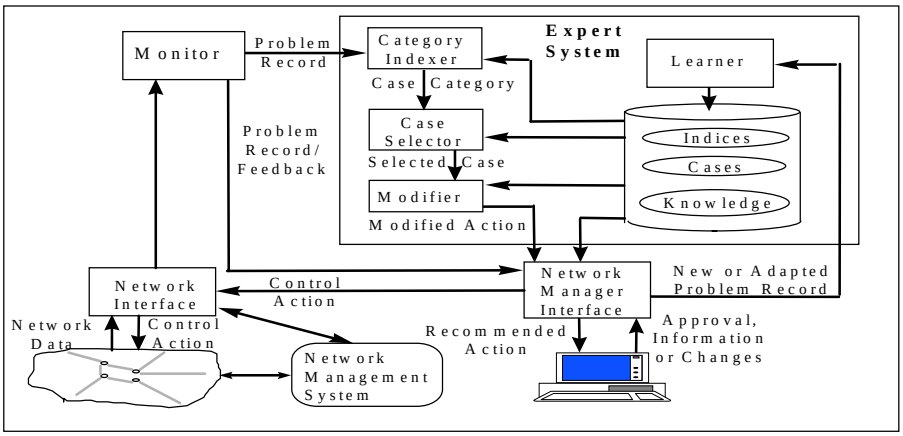

Figure 2 Case Based Reasoning Expert System

Learning new cases will also be required in order to support changes in network management practices, as demanded by the fast evolution in network technology, number of network elements, and support systems. A large number of cases will be identified to describe all significant problems that can be encountered in large scale high speed networks. We provide the system with a "case generalization" mechanism that will enable the system to modify previous cases to handle new problems. It does so by defining specific circumstances under which a case may "borrow" attributes from another case which is more or less similar case.

3. Implementation

We have developed a rule based expert system prototype, with rules for load monitoring and traffic routing. The Network Manager Interface has been implemented using web technology. Using the web-based interface we can perform basic network monitoring services. The network interface has been implemented using the IBM NetView/6000 for AIX Network Management System. The NetView/6000 is an SNMP based networking tool primarily meant for data collection in IP networks. It can also monitor various network parameters against previously defined thresholds and generate traps for network operators. We can use these traps to trigger other programs, such as the monitor.

3.1 Expert System Prototype

The prototype of the rule based ExNet is tested for a hypothetical network consisting of eight hosts and four gateways. Our aim is to simulate conditions for a real network and to route data between two given hosts. The system was built using CLIPS [GIAR 89]. The main tasks performed in this implementation were 1) monitoring congestion in the network by periodically gathering information on system status, 2) deciding whether the congestion has exceeded thresholds or not and what nodes are more congested than others, 3) choosing a new route for the data to be routed between the two hosts, and 4) suggesting a corrective action to the human manager.

The database containing the network information is provided by generating text files using IBM NetView/6000 for AIX. The monitor decides whether or not the problem in the current routing was related to performance or some system fault. Performance related problems include high latency, congestion or high system load. System faults include a crashed gateway or host. To determine a system fault, the monitor module would ping the host or gateway to inquire the status. The expert system analyzes the problem, requested more information from the network interface and after considering the alternatives decide on the new route to be implemented between the two given hosts. If there is a problem on an intermediate host/gateway, such as high load or too many users, the expert system would suggest a set of corrective actions to be taken.

In this implementation example, we have used four main rules: 1) Check CPU State, 2) Check-Routing-ResponseTime, 3) Diagnose-network-status, and 4) Determine New Route. The first set of rules checks the loads on each computer to determine its load status. The heavily loaded nodes once identified are excluded from future path selections. The second set of rules evaluates whether the response time associated with the current network route meets the desired performance requirements. If the latency or delay is excessive and does not meet the user requirement, the current route needs to be changed. The third rule shows how the system can prevent nodes from sending new requests to highly loaded machines or network devices. The last set of rules compute a new path from the source to the destination. It first determines which gateways/hosts are still available for path selection. From the available gateways, the path with the least delay is selected.

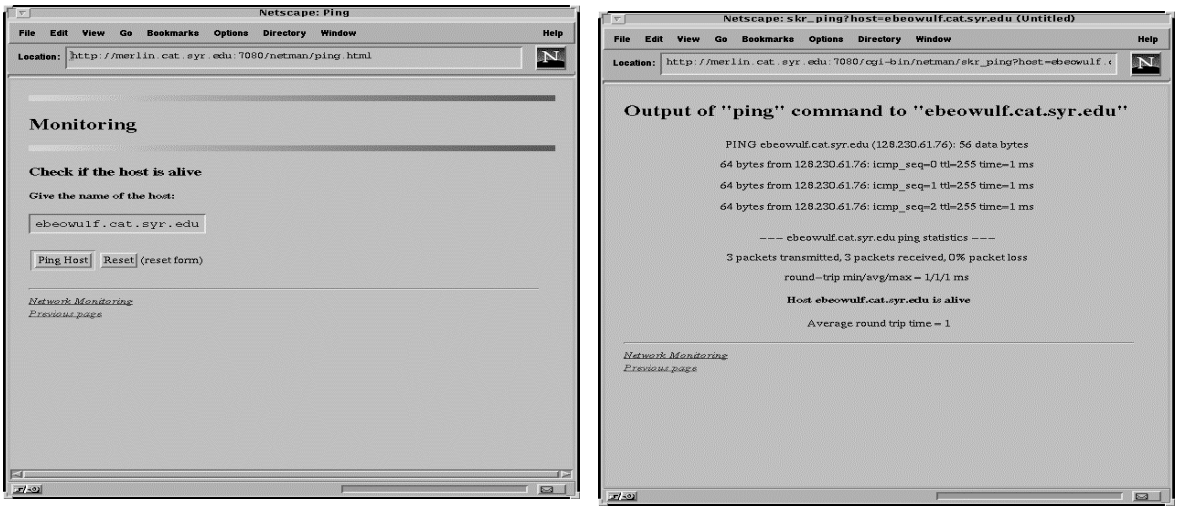

Figure 3 Web Interface to Determine IP Addresses

3.2 Web-based Graphical User Interface

We have developed CGI scripts to perform elementary tasks of network management over the Internet. We have also developed web interfaces to certain services offered by the IBM NetView network management system. The basic capabilities that we have implemented are 1) Ping: We can ping any host to check if it is alive and to determine the network response time, 2) Interface Statistics on a Host: This function is based on the *netstat* command in UNIX. It displays the traffic in packets per second for the given host, 3) Host Activity: It shows the current activities taking place on the host. It gives information such as the users logged on to the system and the system resources being used by each user, 4) IP: Given the name of the host, we can determine the IP address of the machine. An example of the interface for this service is given in Figure 3, 5) Name: given the IP address of a host, we can determine the name of the host, 6) Platform: This is used to determine the platform of the host, and 7) Operating System: This is used to determine the type of operating system running on a given host.

We have developed some scripts to run some NetView command and utilities. The commands used include 1) *xnmgraph*: We use this command to graph the traffic characteristics on each of the interfaces of the selected host. An example of the interface for this service is given in Figure 4, 2) *ovobjprint*: We use this command to obtain some system information about a host within the network managed by the NetView, and 3) *snmpColDump*: This command is used to display the traffic conditions recorded by the NetView on a host.

We also developed certain scripts that obtain and store in a text file the amount of traffic on hosts. These scripts are based on the *netstat* command in UNIX. *Netstat* command returns the total amount of traffic that has passed through its interfaces since the machine was booted last. Our script runs the *netstat* command on an hourly basis. The data returned by *netstat* contains the total number of input octets, total number of input collisions, total number of output octets and total number of output collisions. This data is stored in a file. Since we have the total traffic scenario on an hourly basis, we can obtain the hourly traffic scenario by simply taking the difference of the consecutive terms in the

first data file. The corresponding hourly traffic data is stored in a second file. On the very first data collection of the day, another section of the same program updates a link in the calendar presented on the web.

We have also configured IBM's NetView/6000 to trigger software applications, such as ExNet, on receiving certain events from the network. In a SNMP based scenario that we are considering, the health of the network is monitored by the SNMP agents by measuring some parameters, such as load on a computer or traffic passing through a router. Some of the network activities may overload the system. In such a case, the overload will cause some network parameters to cross the previously defined threshold of acceptable values for these parameters. On detecting any an abnormal condition, the SNMP agent responsible for that particular host or node will generate a trap. This trap will be received by the NetView daemons on the management station [CASE 90]. These daemons will trigger the ExNet System by passing the appropriate parameters concerning the event. ExNet will then decide on the corrective measures to be taken.

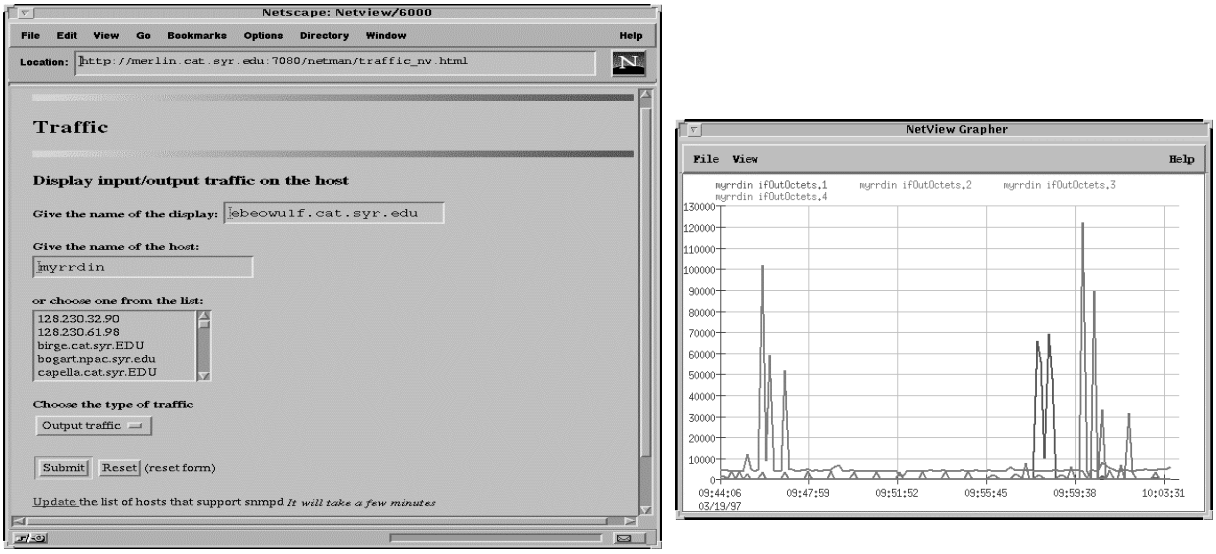

Figure 4 Web Interface to Graph Traffic

4. Conclusion

We have presented two approaches to implement the expert system module of the ExNet prototype to achieve intelligent network management. These two approaches are rule-based and case-based expert systems. We have implemented the rule-based expert system in our prototype. It is efficient for problem solving in domains that are constant and where the expertise for solving the problems is relatively fixed. Unfortunately, today's networks are dynamic with new components, new technologies, new protocols and new applications being introduced routinely. Case based reasoning approach provides an interesting approach to address these limitations. We are planning to develop an intelligent network management system using case-based reasoning and also improve the web-based graphical user interface to support a wide range of control and management functions.

References

1. [BEGG 94] : I. M. Begg, J. Gnocato, A. Haman, "Architectural issues in real time intelligent user interface technology," *IEEE Network Operations and Management Symposium*, 1994.
2. [CASE 90] : Jefferey D. Case, "Management of High Speed Networks with The Simple Network Management Protocol" 1990.
3. [GIAR 89] : Joseph C. Giarratano, "CLIPS users guide, artificial intelligence section," *Lyndon B Johnson Space center*, Oct. 1989.
4. [QUAH 96] : Tong-Seng Quah, Chew-Lim Tan, K. S. Raman, B. Srinivasan, "Towards integrating rule based expert systems and neural networks," *Decision Support Systems*, 1996.