Proactive Network Management Using Remote Monitoring and Artificial Intelligence Techniques

Analúcia S. M. De Franceschi',Marco A. da Rocha*, Henrique L. Weber, and Carlos B. Westphall

Federal University of Santa Catarina (UFSC)
Network and Management Laboratory (LRG) PLAGERE Project (ProTeM-CC-II)
P.O. Box 476, 88040-970 Florianópolis - SC - Brazil
Phone: +55-48-2319739 extension line 235 Fax: +55-48-2319770
e-mail: analucia@lrg.ufsc.br, henrique@lrg.ufsc.br, westphal@lrg.ufsc.br

** National Supercomputing Center and Computer Science Institute*
Federal University of Rio Grande do Sul (UFRGS)
91520-130 Porto Alegre - RS - Brazil
Phone: +55-51-3394699 Fax: +55-51-2260102
e-mail: rock2@cesup.ufrgs.br

Abstract

This work describes the effort to join two different approaches about the development of the Proactive Network Management. The first approach was developed in the Laboratory of Network and Management, at Federal University of Santa Catarina [2][3][4], which uses remote monitoring and simulation tools. And the second one, was developed in the National Supercomputing Center [9][10], which handles the problem with artificial intelligence and *SunNet Manager's* agents usage. The new concept of proactive network management aims at identifying the existing troubles in advance to any performance degradation, as well as providing support for future decision-making actions. The goal of this work is adapting a complete proactive environment which will be use to explore the remote monitors and simulation tools to identify symptoms of proactive management problems. And specially to recognize a problem using artificial intelligence techniques and take proactive actions to solve it, configuring a proactive management application for the prevention of problems in computer networks.

1. Introduction

At present, a new concept for network management is being researched. Proactive management concept enables to identify the existing troubles in advance to any performance degradation, as well as provides support for future decision-making actions. Proactive management aims at maintaining the quality of the offered services at a desirable level, improving the overall network performance [6]. Therefore, organizations will decrease their investments in fault recoveries, and as a consequence save time and gain a more efficient control over resource usage. Moreover, the proactive network management provides a description of the network behavior allowing the creation of a network profile which is statistically correct. Hence, the proactive methodology aims at reducing managerial work in recovering faults that may damage the network performance. Normally, the task of management begins after the already existent performance degradation. Management systems are frequently designed for emiting alarms and for notifying current and presently arising events [5]. This activity is called reactive methodology. On the contrary, proactive management provides system variable pre-search in advance to the service degradation of management applications. Therefore, this paper presents a strategy to join two different implementations of proactive management in computer networks. The first was developed at Federal University of Santa Catarina, and uses the remote monitoring and simulation tools to adapt the proactive concept. The second approach, was developed in the National Supercomputing Center, which handles the problem with artificial intelligence techniques and *SunNet Manager's* agents usage. The goal of this join is establishing a complete environment to test the viability of the proactive network management, especially about the performance aspect.

This work is organized in six sections, Section 2 describes some aspects about the proactive management concept. Section 3 presents the model proposed to implement the proactive management with artificial intelligence. While the test environment and the requirements to implementation of a prototype are described in Section 4. Some results and perliminar tests are disposed in Section 5. Finally, the conclusion and future works will be presented.

2. Proactive network management

As mentioned, the new concept of proactive management must identifying the possible troubles in advance to any performance degradation. For this however, the network behavior must be analyzed to assist

0-8186-7852-6/97 $10.00

in the diagnose process. In parallel to this process, the network was monitored constantly to detect the anomalies and the normal behavior from the network. This information will be used as knowledge to indicate the symptoms and relate them with a known event (a network problem). Through statistical samples within a given period of time it is possible to establish a network profile (called baseline). This baseline may be used as function to determine the normal operation of the network during a new period of time for a specific interval, assessing the levels of traffic at different times on different days.

2.1 Artificial intelligence application

The network management is a hard task and may be associated with another area, such as artificial intelligence and expert systems. Some management systems already are using this association to complement the management functions. In this way, we proposed the use of the artificial intelligence techniques to contribute for development of the proactive management. Furthermore, we used the remote monitoring and simulation tools to support the development of the proactive network. The use of artificial intelligence is justified through the growing of the networks and the necessity of reliable services. A quick cost-benefit survey shows the following advantages: — better quality of service: with the dissemination of the specialist throughout all segments of the network. The administrator's task is facilitated, providing a better performance; — greater agility, lower costs and greater productivity in the execution of services permitted by automation; — higher reliability, with decreased decision-making time; and, — training support for improved human resources preparation. An expert system has four distinct phases: the acquisition of knowledge; the knowledge base; the inference machine; and, the explanatory interface. The acquisition of knowledge involves the extraction and formulation of knowledge from an expert for use in an expert system. In this process, work is performed with "*knowledge engineers*", technicians specialized in the job of helping experts put their knowledge into the expert system using practical rules and knowledge structuring. As the expert puts forth his or her knowledge, the knowledge engineer represents it as a set of heuristic rules that, when coded, drive the process by a mass of information, making the process more efficient. Thus, obtaining these rules is an important step in the acquisition of knowledge. The knowledge base stores the information obtained from the expert and differentiates from a conventional database in that it is active in nature, allowing updates conforming to the context. The structure of the knowledge base will depend on the type of knowledge represented. To have deductive knowledge, the base will usually be composed of rules. To have modeling of physical structures, casual links or interrelationship between models, the ideal structure may be a semantic network. The inference machine selects and applies the appropriate rule during each step in the expert system, manipulating the knowledge base. The inference machine can base itself on premises or elementary bits of information, and tries to achieve its objective through a combination of the two. In this case, it is said that it finds its way forward. The machine can also base itself on an objective and verify the needed premises using the facts involved and arrive at a conclusion. In this case, it is said that it works backwards. Inference machines that use a mixture of these approaches are those which are most successful, since, in most cases, the choices made in the inference process are reproductions of the processes a human would be likely to employ. In an expert system, the knowledge of the problem's domain is organized separately from the other system knowledge, such as the procedures or steps for problem solving, or interaction with the user represented by the explanatory interface, which defines how to present the knowledge. This division is intentional because these systems divide themselves according to knowledge base (the store of specialized knowledge) and inference machine (which unites the procedures for fixing problems, or steps for solving them). The combination forms what are called a knowledge-based system. The base contains facts and rules, and the inference machine decides how to apply these rules and in which order so as to obtain new knowledge. Once the specialized knowledge is separated, it becomes easier for the designer to manipulate procedures [9][10].

3. The proactive architecture

Based on the proactive model presented in [DEF96], and using the techniques proposed in [ROC96] the following architecture (as showed in the Figure 1) is being implemented.

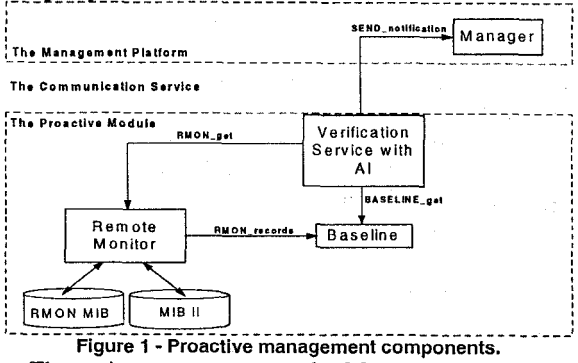

Figure 1 - Proactive management components.

The main components are the Management Platform, the Communication Service and the Proactive Module as proposed in [2]. On the contrary in this work, the atificial intelligence has been attached to the Verification Service, and does not in the Management Platform as demonstrated in [2]. After the diagnosis of problem, the Manager (or the Management Platform) will be notified about the remote network segment situation. Network Management Platform, in a proactive concept, centralizes those management informations derived from sub-networks. At present, for the development of this work the *SunNet Manager* (SNM) version 2.2 from *Sun Connect* is being employed. The Proactive Module is being programmed to notify the management system of any parameters that indicate a decrease in the network performance. These parameters will then be analyzed in

order to provide the corrective actions to be performed by the manager. The Remote Monitor used in this work is the *Beholder* The Next Generation (BTNG) which has been developed by the DNPAP Research Group from the *Delft University of Technology* — Netherlands. The software is public domain and available through FTP at *dnpap.et.tudelft.nl* site. *Beholder* is an Ethernet monitor that employs the SNMP protocol and supports the nine distinct groups of the RMON MIB (*Remote Monitoring Network Management Information Base*). The BTNG agent also enables the creation of table instances and packet filtering as defined in [11]. The communication between the management system and the proactive module was implemented through the employment of sockets which were required for the interprocess communication.

3.1 The artificial intelligence usage

Figure 2 illustrates the modelling of the Verification Service with Artificial Intelligence aspects built-in. The Baseline has been established with monitoring-data obtained with remote agent (*Beholder*). These information will help to determine the symptom, through comparisons with the real-time situation (provided by remote agent).

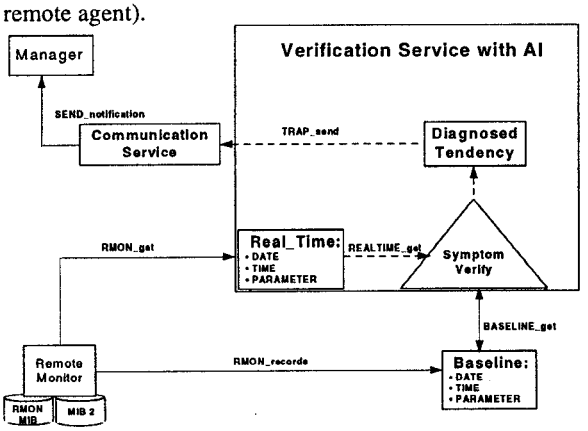

Figure 2 - The verification service schema.

The symptoms are verifyed and help to diagnosis the network troubles tendencies. If some tendency were detected the system must notify the manager (or Management Platform) using a Communication Service (as mentioned in the previous item). The module **Symptom Verify** (in Figure 2) collects data from remote agent in real-time and compares with the baseline data. This activity is realized through the diagnosis process (described in Item 3.2). As result, this function will point a diagnosed tendency and will suggest a mangement action. This information will be sent to the manager through the Communication Service (already mentioned in a previous Item).

3.2 The diagnosis process

The diagnosis process has been elaborated with characteristics of an expert system, according the four phases presented previously (see Item 2.1). A set of trees was built and represents the knowledge base for the system, which had three levels. The lowest is the **Parameter Level**, where the parameter state is identified and evaluated as function of its value, average and standard deviation. The middle level is the **Diagnostic Level** which contains diagnostics made through the analysis of parameters. And finally, in the top of tree, we have the **Suggestion Level**, where the final diagnostics are suggested for the network administrator. The knowledge is represented through facts and rules. The facts has been generated from Baseline files. We are using PROLOG to implement it, operators are being used to define a syntax of rules which represent the knowledge of domain. Thus, the rules are written in a more acessible and adequate way through the specialist, and included in the interpreter to facilitate the implementation of the inference machine.

4. Implementing a minimum prototype

In the following section, the implementation of a minimum proactive prototype is presented and aims at validating the practical results. This prototype is being implemented within the performance management context [7]. The task of monitoring and observing the network is best achieved by agents with resident action in the machines to be managed. Thus, Hostperf was used to measure the performance of workstations interconnected at INF sub-network (see Item 4.1). The following workflow was determined:

- Running Hostperf in INF sub-network to collect statistics in different hours;
- Accessing the remote monitor (*Beholder*) to collect some data from RMON MIB about network performance;
- Establishing a *baseline* from collected data, calculating the average and standard deviation for measurements in normal situations;
- Developing the diagnostic model, so as to activate rules each time the measurement taken is beyond the standard parameters contained in the baseline, stipulated as: — for each measurement taken from the monitoring agents that arrives within reach of the timed measurement in which it occurred, a diagnostic module will be triggered to verify whether of not problems exist according to the module's rules; — if the diagnostic module verifies the a problem exists which could result in a performance drop or congestion, rules will be used to determine the motives for the event; and, — in a third moment, after verifying the previous, measures are taken to avoid the problem, reporting the anomalies found to the network administrator and suggesting corrective measures.

4.1 The test environment

In order to implement the proactive management module and perform the required tests two sub-networks from Federal University of Santa Catarina (UFSC network) were employed. The LRG sub-network contains a Sun SPARC 20 workstation which provides external communication through an Ethernet interface. The INF sub-network is composed of a variety of devices, such as microcomputers, workstations and laser printers interconnected through an Ethernet interface. A Sun

SPARC 10 workstation acts as a host and a gateway for the segment.

4.2 Building a baseline

Simple monitoring of data in short periods of time does not allow to make easy conclusions, such as whether the network is seriously congestioned or not. An another example could be, if were possible to obtain the proposed average at each attempt of transmission, the recovered data would always indicate 0 or 100% of congestion, which would be difficult for one to interpret. This situation is exasperated in the event monitoring mode, since it does not interest the network administrator to receive an event informing that the rate of collisions was, for example, 30% during the last second. On the other hand, if a measure of congestion is obtained during the entire life of the system, the average would be quite low and would not adequately reflect the problems that had occurred during determined periods. In order to provide an efficient control of the congestion, it is necessary to give an average reading over the last N time intervals. The ideal N must be based on modeling of the system's situation considering the real time and the administrative requirements. The Apolo and Atlas workstations and the Venus gateway from INF sub-network (see Item 4.2) were continuously monitored by *Beholder* and *Hostperf* agents during periods of 24 hours. The samples collected had 10 and 60 minutes intervals, and was calculated the average and the standard deviation. After some evaluations, the profile of the network traffic was established. Some objects used to design the profile are: — the cpu utilization from *SunNet Manager's Hostperf* agent used to verify the percentage of CPU usage from workstations; — the `etherStatsPkts` and the `etherStatsCollisions` gathered by the RMON monitor and used to calculate the collision rate; — the `etherStatsPkts64Octets`, the `etherStatsPkts65to127Octets`, `etherStatsPkts128to255Octets`, the `etherStatsPkts256to511Octets`, `etherStatsPkts512to1023Octets`, and the `etherStatsPkts1024to1518Octets` from Statistics Group of RMON MIB used to determine the distribution of packets length; and, — the `hostInPkts`, the `hostOutPkts`, the `hostInOctets`, the `hostOutOctets` and the `hostOutErrors` from Host Group of RMON MIB used to control the traffic of LAN segment. The objects selected are extracted from these archives via a "*parser*" program that is specific for each agent, written in C language, which collects them in an intermediate file, calculating the average and the standard deviation of the values of the object for each hour. This intermediate file is passed by another general parser program, which also calculates the average and the standard deviation of the values of each object, uniting each monitoring day within its respective weekday, thus obtaining the average standard deviation for each hour of each day throughout the period monitored. With this baseline, we have an average standard operation for the network for each hour of each day of the week. Thus, the system has: — the conversion blocks; — the knowledge base; — the inference engine; — the explanation; and, — the interface. The conversion block is responsible for receiving the monitoring and baseline files and converting them to the Prolog fact format. These facts will form the knowledge base. Considering the knowledge base, the inference engine is used for analysis of the values of the parameters and for implying a diagnosis. This diagnosis is supported by an explanation that indicates the motive for the problem, as well as suggesting possible resolutions for it. From the interface, the network administrator receives information from the system as well as the suggestions and has the possibility of expressing an opinion, agreeing or not with the diagnosis given. Beyond this, it should, in whatever situation, describe which of the approaches we followed in trying to solve the problem.

5. Results

The construction of a prototype for proactive management implied the development of performance-related applications for monitoring, analysis and control of network activity. Within this context, some concepts for the development of the proactive management were also considered and included the following activities: — monitoring of a remote sub-network in order to collect information for future performance analysis; — information analysis for the development of the sub-network behavior profile; — defining the knowledge rules; and, — suggesting and controlling the observed parameters. In monitoring network activities, one must be aware that several parameters may have a strong influence on the network overall performance. In Boggs [1] are presented some parameters that are specifically to Ethernet networks (e.g. arrival rate of packets, bit rate, maximum packet length, minimum packet length, packet length distribution). The monitoring of the sub-network was performed in two phases: the first aimed at monitoring the network and collecting the information required in the baseline plotting. The second, performed during the prototype implementation, involved the monitoring of parameters in order to identify any eventual degradation tendencies in the system. At moment, we are defining the knowledge rules according the test environment.

5.1 Preliminary tests

In a preliminary test, one type of problem was detected. Once executed, the prototype automatically converted the monitoring data and consulted the baseline statistics, using the hour and date of the system as a base, it informed the user of the final diagnosis and of suggested actions to be taken by the network administrator. The window of the diagnose has a Diagnose and Suggest screen (see Figure 3). And provides some explanations as an expert system.

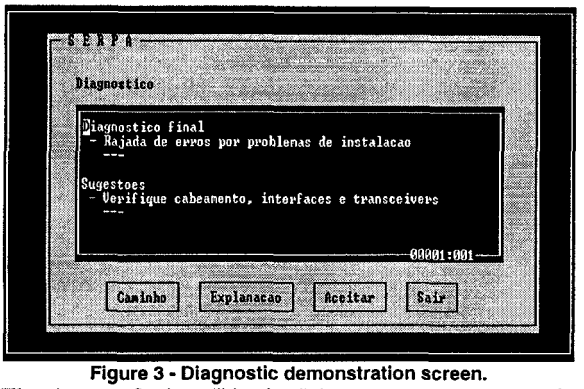

Figure 3 - Diagnostic demonstration screen.

The Accept Option ("Aceitar" in portuguese, as seen in the Figure 3) registers those suggestions that have been accepted by the administrator in a log file; if he or she leaves the system without accepting any, this fact is also registered in the log file. The purpose of the log is to validate the system itself, making it possible to later investigate the situations in which the administrator did not accept the suggestions, so as to create a record that can be used by the inference machine to construct diagnostics. The automation of this process is very important, because the daily work routine of the network administrator normally does not have time to manually procure a log file, principally due to the size of these files. The prototype proved its utility, since the symptom "*flurry of error packets*" enabled the administrator to take rapid measures, due to the fact that notification of the event was received before the event was completed. This system is being improved in sense to minimize the structure of the expert systems to attend the remote service as proposed in Section 3.

6. Conclusion and future works

Some theoretical and practical activities demostrating the necessity of performance evaluation for proactive network management were described in this work. In order to obtain the proper results, the network traffic behavior was monitored and the usage of some resources was analyzed. These activities enabled the verification of the proactive management as a mechanism to improve the network performance and enhance the control over the resource utilization. The work presented proposed an architecture for the development of a proactive management, as well as, a prototype employed for the validation of this new management concept. The experiment demonstrated that UFSC sub-networks are "*well-behaviored*", and has a stable operation and performance, which did not allow us a greater number of problems to be inspected, though the problem that occurred was detected and reported to the administrator, confirming proactive management's capabilities. Its operation and principles can be used as a base for new works to be performed on the proposed model.

Future works will concentrate on implementing proactive management for configuration aspects, as well as analyzing the viability of proactive management concept to other management applications. An important approach would involve the analysis of network performance during the prototype execution and compare the performance observed before the prototype installation. This comparison would be efficient in pointing out any eventual improvement of the network performance. Finally, the model proposed in this work may be extended to support the proactive management on current network technologies such as Token Ring, FDDI, ATM and also be adapted to different management platforms such as Netview, Openview and OSIMIS.

7. References

[1] D.R. Boggs, J.C. Mogul, C.A. Kent, "Measured Capacity of an Ethernet: Myths and Reality", *WRL Research Report 88/4,* Digital Western Research Laboratory, Palo Alto,CA,USA, 1988.

[2] A.S.M. De Franceschi, "An Application to Validate the Proactive Network Management", *M. Sc. Dissertation,* Federal University of Santa Catarina, Florianopolis, SC, Brazil, Feb., 1996.

[3] A.S.M. De Franceschi, L.F. Kormann, C.B. Westphall, "Performance Application for Proactive Network Management" *in Proceedings of the IEEE Second International Workshop on Management Systems*, Toronto, Canada, Jun., 1996. p. 15-19.

[4] A.S.M. De Franceschi, L.F. Kormann, C.B. Westphall, "Performance Evaluation for Proactive Network Management" in *Proceedings of the ICC'96 International Conference on Communications*, vol. I, Dallas, USA, Jun., 1996. p. 22-26.

[5] G. Goldzmidt; Y. Yemini, "Evaluation Management Decisions via Delegation", *in Proceedings of the IEEE/IFIP International Symposium on Network Management*, Apr., 1993.

[6] M. Jander, "Proactive LAN Management", *Data Communications*, Mar., 1993.

[7] A. Leinwand, K.F. Conroy, "Network Management: A Practical Perspective", 2nd Edition, USA : Addison-Wesley, 1996.

[8] R.A. Maxion, F.E. Feather, "A Case Study of Ethernet Anomalies in a Distributed Computing Environment", *IEEE Transactions on Reliability*, Vol.39, No.4, Oct., 1990. Pp. 433-443.

[9] M.A. Rocha, "A Strategy to Implement the Proactive Network Management Using Knowledge-Based Systems", *M. Sc. Dissertation,* Federal University of Rio Grande do Sul (CPGCC, UFRGS), Porto Alegre, RS, Brazil, May, 1996.

[10] M.A. Rocha, "Proactive Management to Computer Networks Using Agents and Artificial Intelligence Techniques", *in Proceedings of the Fifth IFIP/IEEE International Symposium on Integrated Network Management*, San Diego, California, USA, May, 1997.

[11] S. Waldbusser, "Remote Network Monitoring Management Information Base", *Request for Comments 1271,* Nov., 1991.