SEGRE: An Expert System for Pro-active Computer Network Management

Cecília A. Castro Cesar
cecilia@comp.ita.cta.br

Celso de Renna e Souza
celsoren@comp.ita.cta.br

Instituto Tecnológico da Aeronáutica - ITA
Campo Montenegro
São José dos Campos - SP

Abstract

The explosive growth of computer networks requires a more efficient management paradigm, to maintain their services uninterrupted: the pro-active management. Since the necessary knowledge is complex, and network managers often lack the needed experience, the use of Artificial Intelligence appears as a promising venue to obtain a truly pro-active attitude.

This paper presents a pro-active intelligent management system, called SEGRE -A Pro-active Computer Network Managing Expert System – that concentrates the information provided by the RMON - Remote Monitoring - protocol with the additional monitoring of some UNIX parameters.

Keywords: Network Management, Pro-active Management, Artificial Intelligence, Expert Systems, Ethernet, SNMP, RMON, UNIX.

1. Introduction

The enormous growth of computer networks and their daily usage also increased the impact of service interruption: when a network is down, productivity is directly affected.

Trying to reduce these losses, the corporations have invested heavily in network management. This caused a continuous improvement in management tools and protocols, aiming at more homogeneous network standards.

In spite of all this evolution, a general lack of satisfaction with operations performance remains [INS, 97]. What is desired is a management so efficient that it should be able to foresee possible problems, that is, a pro-active (and not reactive) type of management. Proactive management consists in, essentially, detecting deviations from normal network behavior and trying to avoid serious problems that may be announced by these symptoms. The earliest the anomalies are detected, the smallest their impact in the user's activities.

However, this “earliest” has an associated cost. The more intelligent tools are used, and the less the manager’s experience is relied upon, the sooner an acceptable solution may be found. The price to be paid for service continuity is exactly that of adding intelligence to the system, and that’s where the use of modern techniques of AI appears as a promising approach to pro-active management.

The highly specialized knowledge necessary for the detection, diagnostic, and correction of network problems should be represented in an Expert System, and activated when necessary. The Expert System itself could be transported to points where local experience is unavailable, this last problem being a frequent manager complaint [Rich,93].

For the present paper, tools that could supply the necessary data about network behaviour were searched, and expert knowledge was added to them. It is clear that it is not enough to have copious data about the network operations if it is not known how to use it. That is why high-level knowledge must be incorporated into an intelligent system.

This knowledge was obtained from network experts and mapped into an Expert System, with the use of an available shell. The ES, called SEGRE – “Sistema Especialista para a Gerência Pró-Ativa de Redes de Computadores” was built, and was tested for the target environment, the present ITA NET.

2. Network Management

Up to a few years ago, the available network management tools were very limited with respect to several facilities found today, as, for example, the possibility of managing different equipment architectures, furnishing network maps, or offering compatibility with widespread data bases.

The need for this kind of information exchange standards emerged in the 70’s. Several standards were defined and updated, and continue to be discussed through the Internet Requests for Comments (RFCs).

Through the end of the 70’s, the only real available management communication tool was the ICMP (Internet Control Message Protocol), used by the famous PING (Packet Internet Groper) which, for a whole decade, was enough for the then-existing networks.

At the end of the 80’s, network complexity demanded more. Since the TCP/IP set of protocols was sufficiently developed to garantee interoperability, efforts were made to develop a network management protocol. A simple and basic protocol to deal with emergent problems was launched: SNMP – Simple Network Management Protocol – for TCP/IP environments [Stalling,93]. The definition of a management information base, MIB, containing information about managed objects, was then applied.

In the evolution of SNMP, the basic MIB was extended, to include information about net segments as wholes, and not only about individual objects. The specified agent was called RMON (Remote Monitoring) [Waldbusser, 95]. The use of the RMON agent helps the network staff in problem identification [3COM, 97].

The continuous use of these proposed tools exposed their shortcomings. SNMP was far too simple, and work was done to correct this, including in it safety measures.Thus SNMPv2 appeared. Also as an evolution, in 1997 RMON2 was proposed.The RMON2 agents collect data that permit the analysis of the traffic among the various corporation networks.

2.1 Pro-active Management

The most commonly used network management strategy is the reactive one. It consists simply in reacting against the problems pointed at by some monitoring procedure. Frequently, this monitoring system is the user himself! Since no special tool is used, the first steps in problem identification are taken only after the user's complaint. Obviously, such an approach is bound to bring about large delays and general irritation.

Problems must be foreseen and treated before they become critical, this being the core of the pro-active strategy [Rocha, 96].

A common procedure to perform this task consists of the following steps:

- Establish a normal network operational baseline, that is, monitor important parameters in order to obtain a history for an appropriate long period;
- Constantly monitor the network, comparing the obtained profile with the normal baseline. Define thresholds for each parameter;
- Once a deviation from the normal profile is found, start the anomaly treatment procedure.

Pro-active management must be more than just a report generation activity. It must produce statistics of the network behavior over extended periods, taking into account different times and days of the week [Jander, 93].

Both for the baseline definitions and for constant monitoring, RMON agents have been considered as one of the best tools [Matlack, 95].

3. Pro-active approach detectable problems

One may ask, what types of deviations or failures may be detected using a pro-active approach?

Firstly, failures indicated by monitoring parameters given by RMON agents. Beyond that, monitoring by SNMP may be extended to other application critical parameters, as found in the MIBs of key network servers.

In the work herein reported, a careful search was made of the various problems that may appear in an Ethernet network and their solutions, as proposed by network experts. An analysis of the RMON statistics subgroup found that this subgroup permits the evaluation of several problems that may yet appear, and that careful monitoring and apropriate action may avoid.

Some of these problems, treated by SEGRE, and their relation to RMON parameters are briefly presented in Sections 3.1 to 3.7. Section 3.8 contains problems, also treated by SEGRE, related to the UNIX system, widely used in network workstations.

3.1 Overload

The variables *etherStatsPkts* e *etherStatsOctets*, taken together, may offer a good estimate of the network load, and are used as such by SEGRE.

All the installations have a normal use profile, with valleys and peaks in the packet and octet graphs. More attention should be given to the peaks, as overshoots may indicate inadequate use, as with badly dimensioned processes or a clear overload.

Every time the normal thresholds are surpassed, an investigation of which processes are running and who is responsible for most of the usage should be made. Eventually, the baseline should be adjusted if the user profile is changing rapidly.

In any event, in an Ethernet network, in a scale from 0 to 100%, the usage should not surpass 35 to 40% of nominal capacity, accepted to be the saturation level, considering Ethernet access algorithms. Beyond this point it is not advisable to increase the load, since the actual flow will not increase in proportion [Gonsalves, 88].

These two variables are not only important as a load measure. Other parameters are related to them, that are used by SEGRE in other problems. For instance, the number of network errors should not exceed a given fraction of the packet number.

3.2 Excess of broadcast/multicast packets

One of the worst problems that happens because of broadcast packets is the so-called "broadcast storm". This problem can be described, in a nutshell, as a great quantity of packets sent to the broadcast address, causing problems for the stations in a given network segment.

Broadcast datagram costs are high, since all the stations in the segment have to process them. In high enough volume, they may saturate the systems even before the channel itself saturates [Bosack, 88].

This problem is known to exist for at least 10 years, and several RFCs bring advice on how to avoid broadcast storms. Even so, non-standard broadcast addresses and careless use of them continue to cause problems.

Many situations may produce a broadcast storm. Some of the identified ones are the following:

- A given station is configured to identify broadcasts with bits in 0, as in 161.24.1.0. Another one is configured to identify them with bits in 1, as in 161.24.1.255. If, for instance, a packet addressed to 161.24.1.255 is sent, the stations that do not recognize the address are going to try to identify it.

  In this process, they will use ARP (Address Resolution Protocol). They will send an ARP request, also to the broadcast address. The situation may become worse if the ARP, badly set up, answers the request also to the broadcast address. The more stations are badly configured, the worst the storm.

- When a host is being configured for the network, or the station is diskless, it needs information on its own IP address, gateway address, network mask, etc. To this end, the RARP (Reverse Address Resolution Protocol) is used.

Ideally, this information should be obtained using a single broadcast message [Mogul, 85]. In some cases, it is enough to obtain the network mask. The ICMP protocol was then extended, permitting the host to ask for the mask using the Address Mask Request message. Storm problems appear when other hosts reply with the Address Mask Reply to the broadcast address! In general, this may happen when the soliciting IP address is not known. If the asking station does not save the mask, the storm may happen each time it is booted up.

- Some applications that use the network to talk to partners may, in some cases, use too much the broadcast address, when this could better be done with multicast. For instance, a game being played between two stations may force all the hosts in a segment to "play" also!

Multicast packets are less dangerous, since fewer stations are affected, and the equipment itself must be able to handle them. Even so, their level should also be monitored. SEGRE uses both the *etherStatsBroadcastPkts* and *etherStatsMulticastPkts* RMON variables. According to E. Saenz [Saenz, 96], broadcast packets up to 8% of the total packet flow may be acceptable.

3.3 Excessive CRC or Alignment Errors

Packets with CRC error in excess of 2 or 3% of the traffic flow, according to specialists, may affect network performance and their causes should be investigated.

These errors may have different causes, such as signal reflections, electrical noise and inadequate frequency or voltage at transceivers.

Generally, stations transmitting packets with CRC errors have problems in their transceivers or Network Interface Card (NIC). When more than one station has high CRC error rate, cabling, repeaters and hubs should be checked.

For these cases, SEGRE uses the RMON variable *etherStatsCRCAlignError*.

3.4 Too many small packets

Some of the problems explained in 3.3 can also generate packets with less than 64 bytes. The variables used by SEGRE for this are *etherStatsUndersizePkts* and *etherStatsFragments*.

RMON hostTopN group does not gather stations generating small packets. A good starting point is to search for stations generating the largest number of errors. Another approach is to start a capture session using the filter and capture groups.

A certain level of small packets may be acceptable, because they may come from normal collisions. If the number of collisions is not high, though, the errors may be probably coming from a single station.

Analysing the network traffic, small packet origin addresses are good candidates for investigation. Since the addresses themselves may be corrupted, the addresses in the following packets are also good candidates.

3.5 Packets too large

Too large packets have more than 1518 bytes. As with CRC errors, packets too large may be caused by physical problems; however, the problem will, most likely, be caused by a single station. Except for collision caused errors, the procedures here are similar to those for small packets.

Jabbering happens when data bits are transmitted non stop within a packet, causing its length to exceed 1518 bytes and present CRC errors.

When collisions are detected by the network nodes, they transmit the JAM signal to stop transmission. Sometimes, because of high segment traffic, a few nodes may continue transmitting the JAM signal, which will cause higher collision and CRC errors. However, if the traffic is normal, this may indicate that a transceiver or NIC is not recognizing the signal and assuming that collisions have not stopped, continuing then to issue the JAM signal.

SEGRE, for these problems, uses the *etherStatsOversizePkts* and *etherStatsJabber* variables.

3.6 Excessive collisions

The collision rate is the main parameter when evaluating Ethernet performance. Collisions, in the CSMA/CD method, are considered normal, but excessive collisions may indicate serious problems.

Excessive collisions may be a consequence of heavy use, since they increase exponentially with traffic. Collision rates of less than 2% of traffic may be accepted, while a rate of less than 1% is ideal.

In searching for a culprit, RMON´s hostTopN again does not indicate the most collision generating stations. The investigation may start with those generating the most traffic. If the traffic is not high, again, cabling, repeaters and hubs should be checked.

To identify potential collision problems, SEGRE uses the *etherStatsCollisions* variable.

3.7 Packets discarded by RMON

This parameter is a measure of the RMON agent´s capacity to deal with the packet volume in the segment. If its value is too high, indicating that the agent´s resources are not enough to handle the segment demand, then the other computed statistics may become unreliable.

A first obvious solution would be to install RMON in another station with more resources, specially main memory. It should be seen also if the tables constructed in the various RMON groups cannot be reduced in size, or if groups can be activated only when needed.

For SEGRE, in the ITA NET environment, a threshold of 10% was set, to avoid alarm excess. The variable that contains this information is *etherStatsDropEvents*.

3.8 UNIX Parameters

The pro-active management strategy should not be restricted to RMON parameters. The problem with parameters outside the RMON scope is that there is no standard for them. Several manufactures have proposed ad hoc extensions for their own MIBs.

For instance, many important network services run in UNIX servers. Why not keep them also under surveillance? The following measurements should be controlled:

- **CPU use**: activities that require too much CPU time should be identified. A given station should not stay constantly at over 90% of its CPU´s capacity, since then it may not be attending to all its assigned tasks. It is possible to find out for which machines this is happening, which processes are consuming the largest shares, and to suggest a better load balancing (even in some cases, as SEGRE does, to kill processes that are running amok);
- **Disk use**: overloaded disks should be identified. One of the problems that most affect the users is lack of available disk space. Pro-active management should not wait for an application´s demand not met to find that the space is not there. It is possible to define a desirable maximum, above which the expert system is activated. Since it is not advisable (unless your telephone is disconnected) to erase users´s files, it is possible to eliminate some of the systems logs or temporary files.

4. SEGRE

The problems investigated as part of this work, and presented in section 3, generated a set of rules and this knowledge was inserted in a Knowledge Base (KB).

For the development of this base and posterior activation, three external tools were integrated: AionDS, Netview and BTNG. These tools will be briefly presented next.

4.1 Employed tools

- **AionDS:** This tool allows the development of applications based on knowledge. The logic in the KB is expressed through the proper language in Aion: KDL (Knowledge Definition Language). However, the KB needs external communication to receive the alarms from problems in the network and to get parameters from network for problem analysis.

  This is possible through Aion's open architecture that allows integration through a C language programming interface, or ADSAPI (Aion Development System Application Interface).

- **Netview**: Currently, the Tivoli company merged with IBM and the network management software, Netview, became part of TME10 (Tivoli Management Environment), another ampler package for the management of networks. However, since its birth, Netview has remained as an important software of this busy area. A subgroup of the functions of the Netview was used in this work:

Data Collector - one object of interest of the MIB is defined and it is configured how this data will be collected, as for example the frequency of collection and the threshold that, if exceeded, will generate an alarm.

Programming Interface - the API was used to send the alarms to an external program, expanding in this form the functionality of Netview.

- **BTNG**: ' Beholder - The Next Generation ': Is a member of the family of network evaluation software created by the DNPAP group - Network Date Performance Analysis Project -at the University of Delft in Holland [BTNG, 94]. This public domain software is the constructor of diverse RMON MIB groups for monitoring Ethernet networks, that can be interrogated remotely through the SNMP protocol.

4.2 Architecture of SEGRE

Since Netview offers an interface with C and Aion also does, this is the way to integrate them. Figure 4.1 sketches the architecture of SEGRE. The darkest boxes have been developed as part of SEGRE and in white are the external tools.

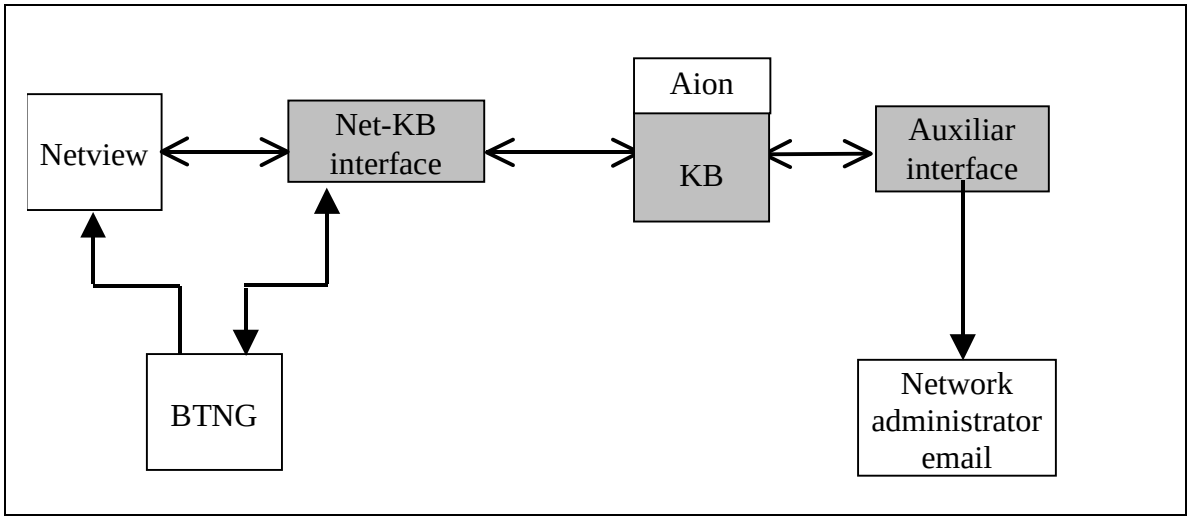

Figure 4.1 - Architecture of the SEGRE

The manner of activating SEGRE will be presented briefly. For more details see [Cesar, 98].

BTNG builds the MIB with monitored objects. Netview consults the MIB and compares it with the baseline. When there is a discrepancy the alarm is sent to a C program that fires the knowledge base. The KB, in turn, can, in the process to find a solution, request other network parameters. This request goes back to the *Net-KB interface* that directly consults BTNG objects. When necessary, the KB activates another C program - *Auxiliar interface* - to execute ancillary tasks as the sending of e-mail to the network administrator or running remote commands in UNIX stations.

It is interesting to notice that most of the communication of SEGRE with the administrator happens by e-mail. This is due to the fact that the management is pro-active: the signaled problems are still potential, that is, the administrator has time to read the message and to implement the corrective action, before the problem occurs effectively. Although they are important notifications, they are not still urgent, there being time to wait that the administrator reads his mailbox.

Beyond e-mail communication, SEGRE also has an interface that allows interaction with the network administrator. This interface allows the observation of the problem research. It is possible to inform SEGRE that corrective actions were executed and, in the positive case, SEGRE goes back to inspect the network to verify if the important problem parameters are below of critical thresholds.

Only after this verification the problem can be considered solved and consequently closed.

The problems have handling in stages. For instance, in the collision problem, the cause can be excess of use or hardware problems. If it seems to be a hardware problem, in the first step, a given advice is emitted. After some interaction with the user, SEGRE goes on to the second stage. As the last sent e-mail is saved, and the last step is known, it is possible to continue from the stopping point: other parameters can be requested and a further advice can be emitted.

In the cases where SEGRE finds that the problem is related to a specific station, it then tries to investigate what is amiss with this machine, consulting some variables directly from its MIB, or running UNIX-remote commands as 'ps ' that lists the processes that are running there. There are cases where SEGRE itself tries to eliminate a process, notifying the user and the network administrator of this occurrence.

Of course, the scope of action is limited if the machine doesn't have an SNMP agent, or if the operating system is not UNIX. In these cases SEGRE only reports the machine address.

4.3 Structure of the Knowledge Base

Each problem was treated in a separate “state”, allowing a modular organization. Rules related to each specific problem, and not all of them, are investigated in a specific session, thus otimizing project and performance during a consultation.

Figure 4.2 illustrates the internal structure of the knowledge base.

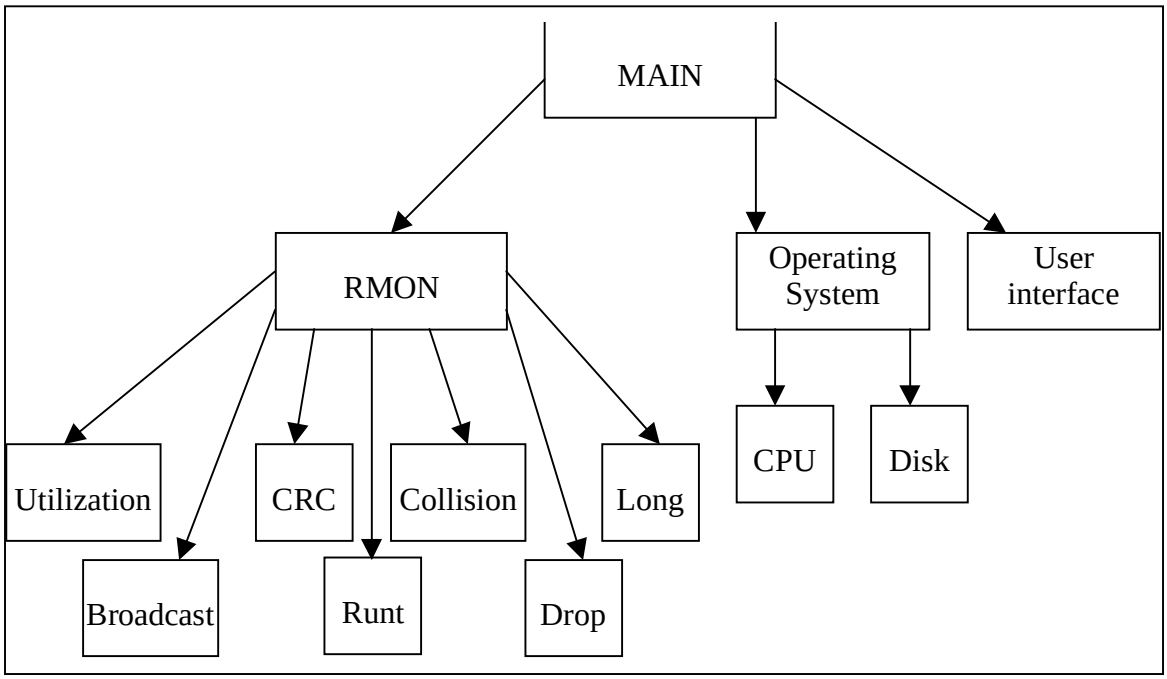

Figure 4.2 - Structure of the knowledge base

5. Validation

During the period when SEGRE was running in ITA NET, two thresholds were exceeded:

- **Overloaded CPU**: one of the most important station of the network was constantly with more than 90% of CPU use. When activated, SEGRE, in its turn, activated a command that constructs a report of the processes that consume more CPU.

  The report was annexed to an e-mail. An inspection of the processes stack indicated that one definitive process that deals with errors in the station, consumed alone almost 50% of CPU. Examining the errors log and correcting the problem, the CPU passed to an average of 10% of use.

- **Excess of short packages (Undersize):** The network administrator received from SEGRE a message suggesting a capture session to find the machine generating this type of packages. In the absence of a capture tool, an analysis of log of events was made using Netview and a research was made to locate the station with problems.

  After isolating this station, a new monitoring indicated the rollback to the normal limits of the *etherStatsUndersizePkts* parameter.

Only these two thresholds were exceeded naturally. To test the remaining portion of the knowledge base, a program was created that generated fictitious parameters, extracted from a file, in the requested order for the rules, as if they were network extracted parameters in a transparent form for the knowledge base.

Using another strategy, aiming at coming close to real cases, traps were also generated artificially, refering to each one of the problems to be tested.

6. Conclusions

In the interviews with the specialists, one noticed that they knew what to do in faulty situations and they did not know so well what to do in degradation situations. In the cast of difficulties for the development of SEGRE, the difficulty in first place was to get the knowledge. What really can be happening, when a parameter deviates from the expected one, with all its possible causes and solutions is very difficult to raise.

The necessity of a change of mentality in the network management was also verified. Of little use will be a sophisticated potential problem alert system if the administrator will wait, before acting, for the potential problem to become a real problem! The mentality today still is very " reactive " and, to the measure that the pro-active systems will be appearing and gaining confidence, this situation is expected to revert: to take care of important matters before they become emergencies.

The ideal would be that SEGRE should be integrated with the tools for cable check, transceiver testing, etc, to advance the maximum with its proper legs, pointing with more accuracy where the problem is. Perhaps in the future, these tools can be better combined and the intervention of human beings minimized.

One of the positive points of SEGRE is the ability to maintain more than one problem open simultaneously. There were occasions when a problem was notified to the administrator, but he still did not execute corrective actions while another problem appeared, not necessarily related to the first one. SEGRE has "memory " and keeps a list of the problems treated and the respective emitted advice. Another positive point of the SEGRE is the interaction with the user allowing sending of messages related to the administrator actions, and not emitting duplicate messages relative to the same problem.

References

[3COM, 97] 3COM Corporation “RMON Methodology”. Site 3Com (fev. 97).URL: http://www.3com.com/nsc/500251.html visited in may/98.

[Bosack, 88]. Bosack, L. e Hedrick, C. “Bridges and Routers, observations, comparisons and choosing problems in Large LANs” *IEEE Network Magazine*. Vol 2. Iss:1. Jan, 88. p49-56.

[BTNG, 94] DNPAP group. “BTNG - Beholder – The Next Generation”. Site et.tudeft (Abr, 94). URL: ftp://dnpap.et.tudelft.nl/pub/btng visited in nov/96.

[Cesar,98] Cesar, Cecília A. C. and Souza, Celso R. *Um Sistema Especialista para a Gerência Pró-Ativa de Redes de Computadores.* 1998, 115p. Dissertação (Mestrado em Informática). Divisão de Ciência da Computação - Instituto Tecnológico da Aeronáutica. São José dos Campos, 1998.

[Gonsalves, 88] Gonsalves, T. and Tobagi, F. "On the Performance Effects of Station Locations and Access Protocol Parameters in Ethernet Networks" IEEE Transactions on Communicatios, Vol. 36, No. 4, Apr. 1988, pp. 441.

[INS, 97] International Network Surveys. "Network Operations Centers Survey". Site INS (97). URL: http://www.ins.com/surveys/noc_results/index.html visited in 09/10/98.

[Jander, 93] Jander, Mary. “Proactive LAN Management”. *Data communications*. Vol 22. n 5. March 1993. p 49-56.

[Matlack, 95] Matlack, Sallie S. “Shopping for an RMON Agent”. Site: IBM (96) URL: http://pscc.dfw.ibm.com/aixtra/issues/mar95/rmon.html visited in 20/10/96.

[Mogul, 85] Mogul, J. e Postel, J. “Internet Standard Subnetting Procedure”. Request for Comment - RFC 950, 1985. URL: http://ftp.sunet.se/ftp/pub/Internet-documents/rfc visited in 12/03/98.

[Rich, 93] Rich, Elaine e Knight, Kevin. *Inteligência Artificial*. Makron Books, São Paulo, 1993.

[Rocha, 96] Rocha, Marco A. , Fernandez, Luiz F.N. and Westphall, Carlos B. "Gerência Pró-ativa de Redes de Computadores usando agentes e técnicas de IA". 14o Simpósio Brasileiro de Redes de Computadores. Recife, maior 1996, p97-117.

[Sáenz, 96] Sáenz, Esmilda e Tarouco, Liane. “Um sistema especialista para Gerência Pró-ativa Remota”. 14o Simpósio Brasileiro de Redes de Computadores. Recife, maior 1996. p118-137.

[Stallings, 93] Stallings, William. *SNMP, SNMPv2 and CMPI: The Practical Guide to Network Management Standards* - Addison Wesley. Massachusetts, 1993.

[Waldbusser, 95] Waldbusser, S. “Remote Network Monitoring Management Information Base: RMON MIB. Request for Comment - RFC1757, 1995. URL:: http://ftp.sunet.se/ftp/pub/Internet-documents/rfc visited in 17/04/97.