RELIABLE NETWORK MANAGEMENT TOOL THROUGH INTERNET

E. Magaña J. Villadangos J. Aracil J. R. González de Mendívil

Universidad Pública de Navarra
Departamento de Automática y Computación
Campus Arrosadía s/n
31006 Pamplona – SPAIN
{eduardo.magana, jesusv, javier.aracil, mendivil}@upna.es

ABSTRACT

This paper presents a number of issues which are relevant to the development of network management tools based on Web interfaces. Specifically, the following technologies for the network management console are evaluated: HTML, HTML/Java and Tcl/Tk. The pros and cons of the abovementioned technologies are discussed in terms of reliability, probe/console transmision delay, CPU use, data processing and alarm reception capabilities. To do so, we analyze the implementation issues of a novel network management tool, PROMIS, which performs traffic monitoring functions with information collected from a number of network probes.

KEYWORDS: *Distributed sniffer, network management tools, user interfaces for traffic analysis.*

1. INTRODUCTION

Due to the severe congestion conditions that are encountered in current data networks it becomes necessary to use network management tools to monitor network traffic. Currently used protocol analyzers (sniffers) do not provide a global vision of the network since their use is restricted to the analysis of a single network segment. On the other hand, SNMP/RMON management platforms suffer a lack of reliability because of the use of UDP as transport protocol. Not only SNMP packets can be lost during congestion periods but the polling scheme from manager to agent is unable to provide the accuracy and on-line monitoring capabilities which are due for the current Internet[1,2]. Furthermore, such network management tools are usually based on a single management console. Thus, the monitoring information reported by the agents can only be accesed from a unique terminal.

In order to overcome the abovementioned limitations a Web-based management interface can be used to provide a fully distributed network management system. However, network monitoring is a CPU-intensive task and requires efficient, real-time graphic display capabilities. Thus, the suitability of Web technologies for network monitoring and control purposes still remains an open issue. This paper provides an analysis of Web-based technologies for network monitoring purposes in wide are networks. The following network management interfaces are analyzed in the paper: HTML–only browser, HTML/Java browser and Tcl/Tk application. The case study is a novel distributed network management tool called PROMIS, which serves the purpose of network traffic monitoring over the Internet.

The abovementioned interfaces are evaluated in terms of the communication and computing resources that they require. Each of the user interfaces under study behaves in a different way regarding bandwidth consumption, response delay, CPU utilization and alarm handling. The remaining of this paper is structured as follows: Section 2 presents the architecture of the system under analysis. Section 3 presents the performance evaluation of the management interfaces. Finally, the conclusions that can be drawn from this study are outlined in section 4.

2. ARCHITECTURE

For the sake of completeness , the basic characteristics of the PROMIS are presented in this section. A more detailed description can be found in [3].

PROMIS components are probes and consoles. The probe is implemented on a PC with Linux operating system and it is in charge of 1. gathering the traffic from the network, 2. processing traffic statistics and 3. serving the management information. Probes deliver such management information to the console, which is implemented using a multiplatform interface.

The information offered by the probe can be broadly classified as on-line and off-line, depending on whether the information is being displayed continuously at the console graphic interface or not. A larger amount of network resources are needed for on-line information since the information must be delivered to the network management console in a timely fashion. A desirable characteristic of on-line monitoring tools is small transfer time, to allow the network operator to react quickly in the event of an alarm. On the other hand, off-line parameters are requested on-demand from the console. Thus, they are

stored as HTML pages in the probe, which can be accessed via the probe Web server.

The management information delivered by PROMIS includes global traffic parameters like bytes/sec, packets/sec and mean utilization, filtering by machine, protocol, service or application. PROMIS functionalities are not only limited to on-line and off-line network monitoring but alarms and expert system functions are also available. Should a problem be detected in a network segment, the probe inmediately generates an automatic alarm that is sent to the consoles.

A key issue in the design of PROMIS is the console user interface. This program allows the user to access the monitoring information provided by the different probes. Such information is presented to the user as graphics, tables or reports. In order to ensure that the console is not tied to a single machine, a multiplatform design for the user interface is undertaken. The following architectures are evaluated :

- HTML-only browser: The probe updates internal web pages which are made available to the console through a standard Web server. Thus, the user interface acts as a standard Web browser, namely Web pages are downloaded from the probe and presented to the user on-demand. As a consequence, bulk data transfers take place between probes and console, since the whole web page has to be downloaded. Furthermore, the console does not provide users with on-line monitoring capabilities.
- HTML/Java browser: The use of Java permits the execution of code in the client and, thus, a program can be downloaded to the console in order to provide on-line network monitoring features. On the other hand, the off-line capabilities of the HTML-solution can also be integrated in the console.
- Tcl/Tk application: The above mentioned on-line and off-line monitoring features can be also provided by using a custom user interface. Tcl/Tk is a script language which allows for graphical interface rapid prototyping. In comparison to Java, Tcl/Tk offers the possibility to access hard disk in the console, which reduces the use of communication and computing resources substantially, as will be explained in the next section.

3. PERFORMANCE EVALUATION

Since the volume of data that is processed and exchanged between probe and console is significant, a careful analysis of the communication and computing resources that they require is in order. Specifically, we focus on the following performance parameters for the three design paradigms under consideration: reliability, transmission delay between probe and console, console CPU use, data processing and alarm reception capabilities.

3.1. Reliability

PROMIS uses TCP as reliable transport protocol in contrast with SNMP/RMON platforms, that use UDP. In order to evaluate UDP performance, we emulate SNMP behavior with ICMP ECHO packets of variable length from console to probes, using different network configurations ranging from local area network to wide area network. The results are shown in figure 1.

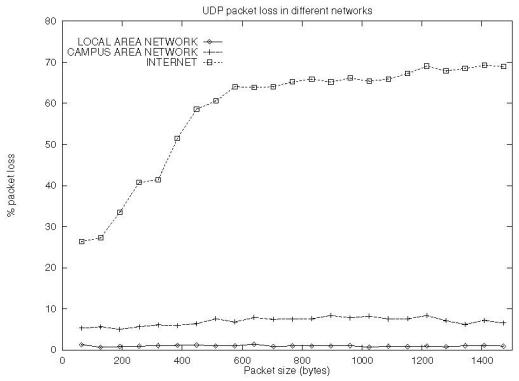

FIGURE 1: UDP packet loss in different networks

An increase in packet loss with packet size and number of hops between console and probe is observed. So, SNMP management platforms are not robust to perform traffic monitoring over Internet. On the other hand, the use of HTTP in the HTML-only solution implies that the underlying protocol is TCP. As far as the HTML/Java and Tcl/Tk applications are concerned, SOCK_STREAM sockets are used in order to have TCP as the transport protocol. As a result, PROMIS outperforms SNMP-based tools as far as realiability features are concerned.

3.2. Transmission delay

On-line network monitoring requires a high degree of interaction between probe and console. Thus, transmission delay from probe to console should be kept as a minimum. Figure 2 shows the transmission delay for the HTML only, HTML/Java and Tcl/Tk interfaces. The experiment is performed with probe and console being placed in the same local area network, which represents the best case concerning transmission delay. In order to test the on-line monitoring performance, the probe sends information updates to the console once per second. Regarding the HTML-only interface, we note that the download of an entire graphic of about 30KB takes place for each information retrieval from probe to console. On the contrary, only a timestamp and the parameter value have to be transferred for the HTML/Java and Tcl/Tk solution, namely 16 bytes of total data. Moreover, figure 2 shows that the increase of network load makes information transmission delay even more variable.

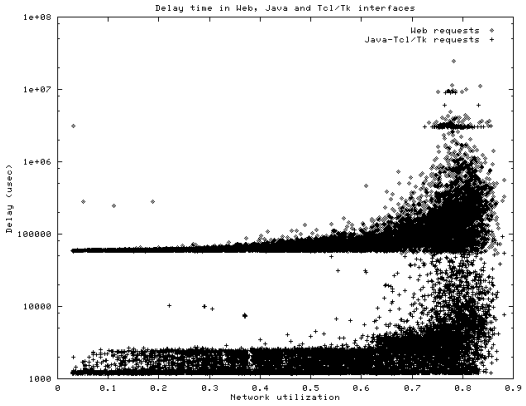

FIGURE 2: Transmission delay in HTML-only , HTML/Java and Tcl/Tk interfaces

3.3. Console CPU use

Since the HTML-only solution is based on information retrieval from the probe, the processing burden in the console is negligible. On the other hand, Java and Tcl/Tk heavily load the CPU, due to the dynamic nature of the graphic that is being displayed. Assumme that the graphic displayed in the console is updated once per second with the following simple algorithm:

```
While(true) {
        receive new value x;
        rescale the graphic;
        plot a box with amplitude x;
}
```

The resulting CPU load from running this simple algorithm in an SparcStation 167MHz, 128 RAM is shown in figure 3. Java improves Tcl/Tk in CPU efficiency, possibly because of bytecode generation in Java. Such bytecode is an intermediate step to generate machine-dependent code. Therefore, program interpretation is more efficient.

3.4. Data processing

Both Java and Tcl/Tk allow to process data in the console side. However, the use of Java as a web applet has some limitations due to security reasons. Specifically, Java applets are not allowed to save data to hard disk. On the other hand, Tcl/Tk allows to save data to the hard disk. Thus, traffic captures can be downloaded and stored for future use, instead of having the traffic capture downloaded from the probe every time a request for the capture is issued.

3.5. Alarms reception

Alarm are triggered in the probe as a consecuence of abnormal conditions in the network, which are detected by a threshold on certain parameters, such as the packet error rate. Such alarms are inmediately sent to the console, which is in standby for incoming alarms.

Note that it is not possible for the console to open a socket in listen mode with the HTML-only interface, since only HTML page retrievales from the probe to the console are performed. As the only possible solution, the probe could mantain an alarm list on its hard disk, that could be downloaded from the console. Note that this approach requires intensive polling from console to probe, which would constrain the on-line features needed for the delivery of alarms.

On the other hand, even though a Java program allows to open a socket in listen mode, we note that a Java applet would only accept connections from the machine from which it has been downloaded, due to security reasons. Finally, a Tcl/Tk application, as any other application written in a general purpose programming language, is able to listen to alarms coming from any probe. Thus, Tcl/Tk outperforms Java and HTML-based solutions in alarm reception capabilities.

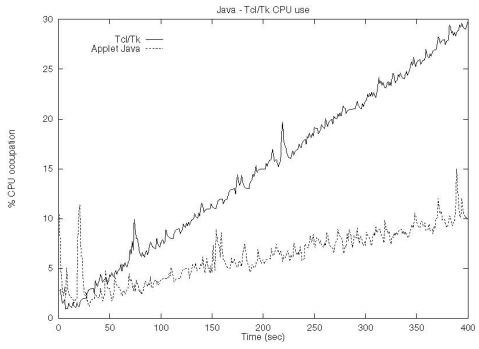

FIGURE 3: Performance comparison Java versus Tcl/Tk

4. CONCLUSION

The design of network management tools for traffic monitoring over the Internet presents several trade-offs regarding reliability, CPU and storage use, transmission delay and alarm handling. While HTML-only and HTML/Java interfaces provide portability, a Tcl/Tk interface provides the functionalities that are available in any programming language. Our findings show that the best design criteria is to combine both HTML/Java and Tcl/Tk. An HTML/Java interface can be used to access management data previously stored in the probe or to perform on-line parameter monitoring, using a standard Web browser. On the other hand, Tcl/Tk can be used to code applications that perform a more rigorous on-line network control since they provide better alarm reception, storage capabilities and an ad-hoc graphic interface.

References

[1] E. Mier, "Network world, Bell Labs evaluate SNMP on bridges". *Network World*, April 1991.

[2] E. Mier, "Network world, Bell Labs test router's SNMP agents". *Network World*, July 1991.

[3] E. Magaña, J. Aracil, J. Villadangos. "PROMIS: A Reliable Real-time Network Management Tool for Wide Area Networks". *Proc. of IEEE Euromicro'98*, Vaasteras, Sweden, August 1998.