A network monitoring tool for student training

Miguel A. Mateo Pla, M.P. Malumbres
Departamento de Informática de Sistemas y Computadores (DISCA)
Facultad de Informática (FI)
Universidad Politécnica de Valencia (UPV)
{mimateo,mperez}@disca.upv.es

Keywords: Network monitoring, students training, Internet discovering, TCP/IP protocols.

Abstract: Computer networks are an important element in any modern organization. Computer science students must be able not only to use the computer networks but also to tune them in order to maximize their performance. In our Computer Networks course[7], we have introduced some laboratory sessions where students learn the TCP/IP protocol suite using a set of software tools.

This paper presents one of them, a network-monitoring tool, that was developed to help our students. This tool gathers the information that TCP/IP protocols exchange and displays it to the user in such a manner that it is easy to understand the behavior of TCP/IP protocol family. It has the following features: Free for students, simple to use and configure, it has a friendly user interface, and its design is based on educational purposes.

1. INTRODUCTION

Nowadays, more and more institutions and companies use computer networks as an important part of their decision and production systems. These networks are usually connected to Internet, so the main protocol suite used is TCP/IP. Computer science students will be network managers, networking software developers, network designers, etc., so they need to know and understand how computer networks works from the low level (hardware) up to the application level.

The course of Computer Networks at Technical University of Valencia presents computer networks using the OSI reference model, that is, from physical layer to application layer [5][6].

The first part of this course is focused on the three lower level layers of network architecture. In other words, the main topics are related to transmission medium, signal codification and modulation, noise perturbation, point-to-point communication protocols, local area networks (with special interest on Ethernet networks) and an introduction to internetworking.

The second part of this course is dedicated to study the TCP/IP protocol stack [3] and the Internet application level (sockets interface[4], client-server model and standard applications).

There are several practice sessions in the laboratory related to the content of theory sessions[2]. With respect to the first part of the course, we propose several practice sessions: Develop a software that shows the fundamentals of Fourier analysis, connect two computers using a RS-232 serial interface, evaluate point-to-point protocols and, finally, use the NDIS interface to access network devices.

With respect to TCP/IP protocol suite and Internet applications, we have proposed several laboratory sessions to build a simple TCP/IP stack and to design Internet applications.

The TCP/IP stack is developed from network access to transport protocols, reusing the protocols debugged in previous sessions to build the next one in the hierarchy.

All laboratory work is developed under Windows platform. We will use VPACKET [9], which allows us to access to NDIS interface, as the base level. The following protocols are proposed: ARP, ICMP, IP and UDP, and a simple version of TCP [1].

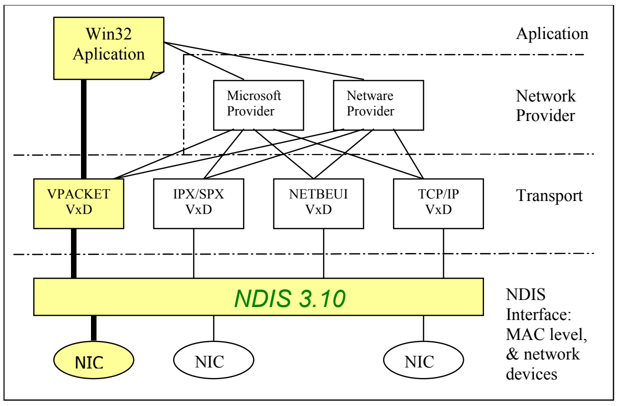


Figure 1. Windows network system architecture.

After developing a protocol, the students have to test it. To test a protocol we proposed a standard application that works as their corresponding upper level. The protocol stack is made down to up, so upper protocols are not available. The students use the services offered by the developed protocol through a special application. The students act as upper levels and ask to protocols for implemented services, and then the applications show the received data from protocols. The students test this data to see if the protocols work correctly.

In many cases it is difficult to know what goes wrong with the student's protocol implementations, so hard debugging process have to be performed, wasting a lot of time and sometimes without positive results

The rest of this paper is organized as follows: First, we define the reasons that lead us to develop a customized monitoring tool. Then, we describe the monitoring tool main characteristics. In addition, some practical examples of use are explained. Finally, some conclusions and future work are drawn.

2. MOTIVATION

In the real world, one of the most valuable tools in network management is a network monitor. This kind of tool allows the network manager to determine the characteristics of the network traffic. It can also be used to analyze the contents of network packets and detect failures in protocol implementations.

We found that commercial monitoring tools use to be too complex, sometimes are hard to manage and their cost is, in general, very high. So, we decided to develop our own network monitoring tool. This decision allowed us to introduce educational objectives in its design.

The proposed network-monitoring tool has the following characteristics:

– It has to be cheap, better if free. If we build our own application, the cost will depend on the programming environment and used tools.
– The interface must be easy to use and to understand. A graphic user interface would be desirable.
– The tool will be useful when studying TCP/IP protocol suite, because it would allow analyzing in detail the behavior of TCP/IP protocols.
– It has to be designed with a high modularity degree, so new modules (protocols) can be easily added.
– The results given by the tool must be useful and easy to understand by our students. These results could help students and teacher in detecting bugs in protocol implementations.

In addition, from our experience we know that students do not have a global vision of what really happen in a computer network. They make network applications in laboratory, but their vision of the network architecture is limited to the applications behavior. It is important to know, for example, what happens in our host, network devices, and the server host, when a Netscape user asks for an URL. The monitoring tool helps students to understand what is happening in network and the communicating hosts, and learn with real traffic the principles of TCP/IP and Internet.

Finally, a statistics module has been added to application. This module supplies a lot of information about our LAN traffic (protocol distribution, network errors, packet counters, etc.)

3. MONITORING TOOL

Now, we will present the main features of our tool. We will make a tour through the application, describing the network characteristics related with each user interface and the obtained data.

In any case, the main objective in the design of the interface was simplicity, that is, the student would understand what the application will do if he changes any parameter, and the data windows have to reflect this change as soon as possible.

The main objective is to develop a network-monitoring tool for TCP/IP protocol family. This tool has to allow the study of TCP/IP main protocols as: ARP, IP, ICMP, UDP and TCP. The relationship between these protocols and OSI levels is shown in *Figure 2*.

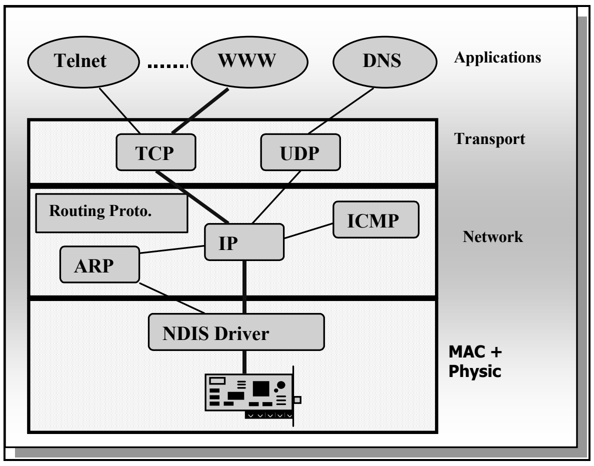


Figure 2. TCP/IP and OSI levels.

The tool is able to:

- Capture network packets. It filters those packets in a flexible and efficient way. The filter parameters can be modified and tune to better fit the educational objectives of each practice session.
- Decode captured packets in order to display them in a readable form.
- Store and retrieve sessions to/from disk. This allows to analyze the captures off-line.
- Collect statistic information. This information can reflect the use of network or the activity of a specific protocol.

The tool is designed to use it in Ethernet networks, but other networks technologies could be used after the addition of the corresponding network access module. In any case, Ethernet network is the one used in laboratory, and it is also widely used in industry.

3.1 Overall description

The main application form is shown in *Figure 3*. Once started, the network-monitoring tool allows the following operations:

1. Select the network interface to monitor. This step is necessary when the application is unable to configure by itself network connection properly. For example, when the host has more than one network adapter or interface. This dialog is shown in *Figure 4*, where we can see the config information.
2. Create a new capture session. Before the capture process a filter must be defined. The filter determines which packets will be displayed to user. The capture sessions are very useful to understand the protocol operation.
3. Create a statistic session. If the user is only interested in the global network behavior, the statistic session will give a lot of information that shows the network activity.
4. Load a previous saved session. Some times is desirable to compare the results of one session with a previous one. User must be able to store and load data obtained in both kinds of sessions: capture and statistic.

Different sessions can be created, but only one can be active at the same time. Whenever a capture session is started, it is possible to extract statistic information so there is an option to do that (See *Figure 4*).

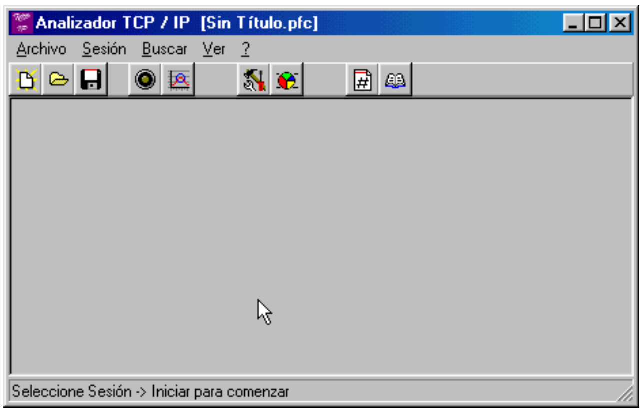
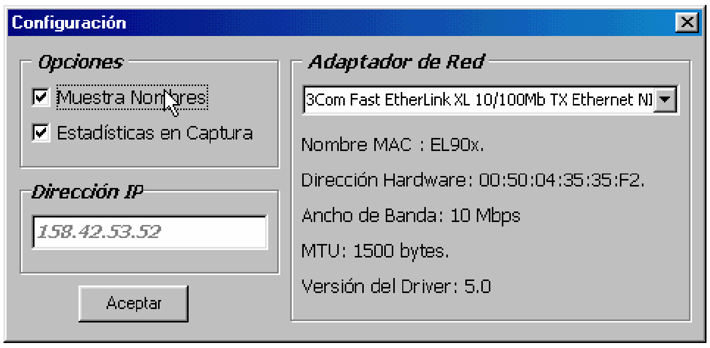

Analizador TCP / IP [Sin Título.pfc]
Archivo Sesión Buscar Ver ?
Seleccione Sesión -> Iniciar para comenzar

Figure 3. Main application form.


Configuración

Opciones
☑ Muestra Nombres
☑ Estadísticas en Captura

Dirección IP
158.42.53.52

Adaptador de Red
3Com Fast EtherLink XL 10/100Mb TX Ethernet NI

Nombre MAC : EL90x.
Dirección Hardware: 00:50:04:35:35:F2.
Ancho de Banda: 10 Mbps
MTU: 1500 bytes.
Versión del Driver: 5.0

Aceptar

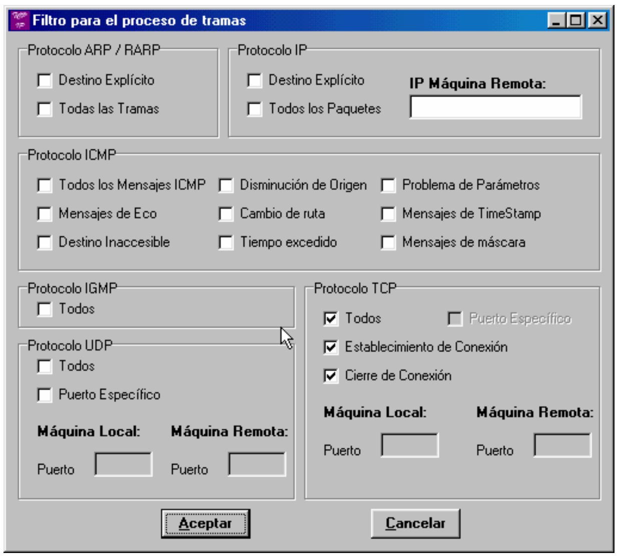
Figure 4. Configuration dialog box.

3.2 Capturing packets

The first step to capture packets consist of a filter definition. The filter allows the user to select packets that will be displayed during and after the capture session. The interface to define the filter is shown in *Figure 5*. In the example of *Figure 5*, all TCP packets will be captured.

As shown in the same figure, the filter only determines the capture decision in base of the TCP/IP protocols and the information that packets have in their headers.

As soon as packets are captured, they are shown to the user in a list box. When the user selects any entry of this list, a second window appears with the header information of the selected packet. This second window is a tabbed window, and has as many sections as protocols involved in the captured packet. The *Figure 6* shows an ICMP capture session.


Filtro para el proceso de tramas

Protocolo ARP / RARP
- ☐ Destino Explícito
- ☐ Todas las Tramas

Protocolo IP
- ☐ Destino Explícito
- ☐ Todos los Paquetes

IP Máquina Remota:

Protocolo ICMP
- ☐ Todos los Mensajes ICMP
- ☐ Disminución de Origen
- ☐ Problema de Parámetros
- ☐ Mensajes de Eco
- ☐ Cambio de ruta
- ☐ Mensajes de TimeStamp
- ☐ Destino Inaccesible
- ☐ Tiempo excedido
- ☐ Mensajes de máscara

Protocolo IGMP
- ☐ Todos

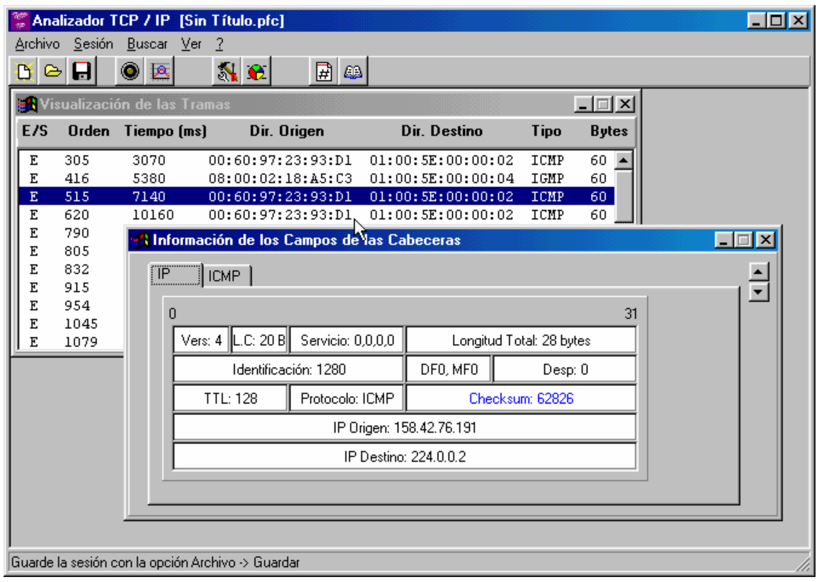
Protocolo UDP
- ☐ Todos
- ☐ Puerto Específico

Máquina Local: Puerto
Máquina Remota: Puerto

Protocolo TCP
- ✓ Todos
- ☐ Puerto Específico
- ✓ Establecimiento de Conexión
- ✓ Cierre de Conexión

Máquina Local: Puerto
Máquina Remota: Puerto

Figure 5. Filter definition dialog box.


Analizador TCP / IP [Sin Título.pfc]

Visualización de las Tramas

E/S	Orden	Tiempo (ms)	Dir. Origen	Dir. Destino	Tipo	Bytes
E	305	3070	00:60:97:23:93:D1	01:00:5E:00:00:02	ICMP	60
E	416	5380	08:00:02:18:A5:C3	01:00:5E:00:00:04	IGMP	60
E	515	7140	00:60:97:23:93:D1	01:00:5E:00:00:02	ICMP	60
E	620	10160	00:60:97:23:93:D1	01:00:5E:00:00:02	ICMP	60
E	790					
E	805					
E	832					
E	915					
E	954					
E	1045					
E	1079					

Información de los Campos de las Cabeceras

0			31
Vers: 4	L.C: 20 B	Servicio: 0,0,0,0	Longitud Total: 28 bytes
Identificación: 1280		DF0, MF0	Desp: 0
TTL: 128	Protocolo: ICMP	Checksum: 62826	
IP Origen: 158.42.76.191			
IP Destino: 224.0.0.2			

Guarde la sesión con la opción Archivo -> Guardar

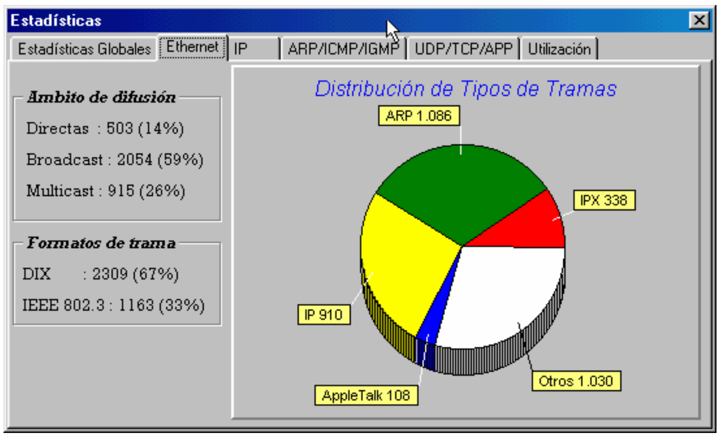
Figure 6. An ICMP capture session.

3.3 Statistics presentation

As said before, we can obtain statistics when a capture session is started. This solution is not good if we only want to get statistics because the capture session must be stored and spends a lot of memory.

A statistic session can be started alone. In this case, the application asks the user for two parameters: statistic period (in seconds) and the interval used to display data in the statistic window. The statistic window is shown in *Figure 7*.

The six categories, or tabs, corresponds to three levels of protocol stack and two for general information. The network level is split in two sections, one for IP protocol and another for ARP, ICMP and IGMP protocols. The utilization graphic is calculated for Ethernet networks. If the tool is used in other kind of networks, the utilization computation should be adjusted.



Ámbito de difusión

Directas : 503 (14%)

Broadcast : 2054 (59%)

Multicast : 915 (26%)

Formatos de trama

DIX : 2309 (67%)

IEEE 802.3 : 1163 (33%)

Figure 7. Statistics window showing data about different frame types.

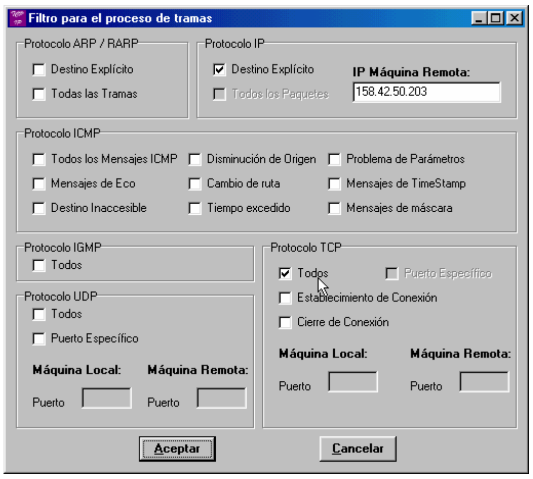
4. EXAMPLE OF USE

Now, an example of use is presented. This example will show how the tool can be used to learn or debug a protocol. The selected protocol is POP3, a TCP based mail protocol.

The first step is to define what we are going to do and then configure the appropriate filter. The POP3 protocol is used to transfer mail from a mail server to local host. The POP3 protocol is a client driven protocol, that is, the client initiates all the transactions.

We suppose that the involved hosts do not have other connection opened and that we do not know which ports will be used. So, we need all TCP traffic between hosts. The required filter is shown in *Figure 8*. Also, we could configure a more strict filter, indicating, not only the server IP address, but also the server remote port (POP3 well-known port: 80). If we do that, the filter would only provide packets belonging to the POP3 session, filtering any other TCP packets between both hosts.

Once the filter is defined, the monitor starts to capture data. To allow the monitor to capture data we can start a mail client program and read the mail in server. While mail client is reading the mail, the monitor is capturing the transferred information. In a typical session with MS-Outlook with no new message to be read, all the transaction is done using 31 packets.


Filtro para el proceso de tramas

Protocolo ARP / RARP
Destino Explícito
Todas las Tramas

Protocolo IP
[✓ Destino Explícito]
Todos los Paquetes
IP Máquina Remota: 158.42.50.203

Protocolo ICMP
Todos los Mensajes ICMP
Disminución de Origen
Problema de Parámetros
Mensajes de Eco
Cambio de ruta
Mensajes de TimeStamp
Destino Inaccesible
Tiempo excedido
Mensajes de máscara

Protocolo IGMP
Todos

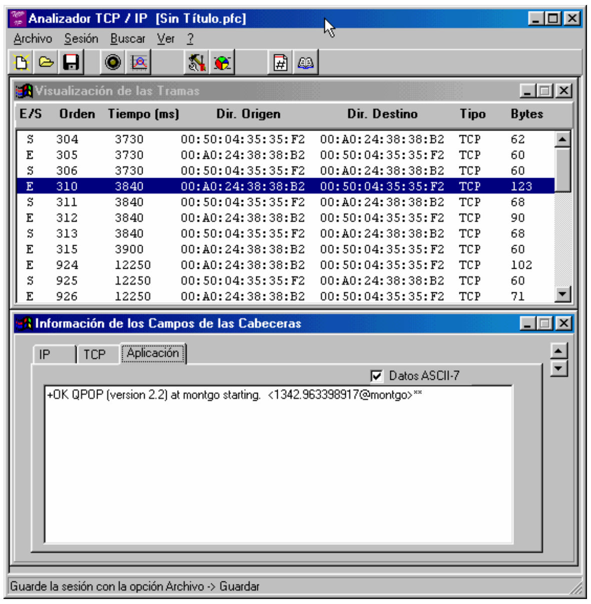
Protocolo UDP
Todos
Puerto Específico
Máquina Local: Puerto
Máquina Remota: Puerto

Protocolo TCP
[✓ Todos]
Puerto Específico
Establecimiento de Conexión
Cierre de Conexión
Máquina Local: Puerto
Máquina Remota: Puerto

Figure 8. Filter example.

Then, the contents of these packets can be viewed and analyzed (*Figure 9*). The sequence observed in a POP3 session starts with the TCP connection establishment. After this, the application protocol could start to transfer information. As many other protocols, POP3 formats messages with ASCII-7 character set, so it is easy to understand what is happening. The messages are sorted in time.

Other example can be proposed using *ping*, *traceroute* or name server (DNS) based applications. *Ping* and *traceroute* use ICMP and DNS protocol uses UDP. Almost all Internet users know *ping* and *traceroute* applications, but the way they operate is not well known. We show in *Figure 10* the result of capturing ICMP and UDP messages when performing the command; '*ping www.ieee.org*'.

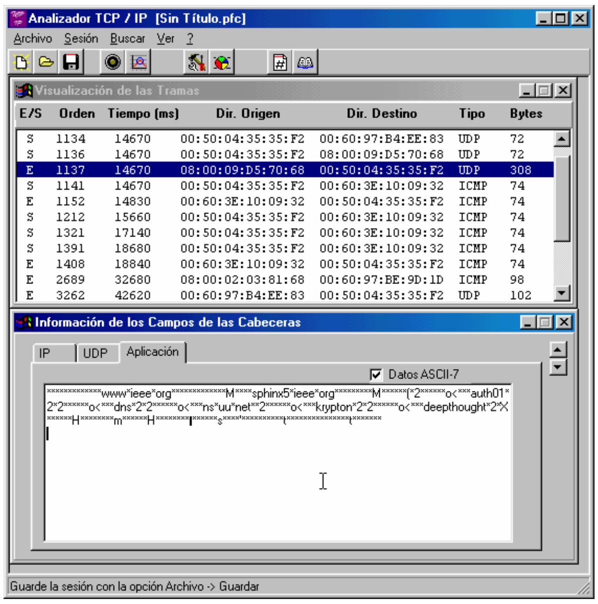

Analizador TCP / IP [Sin Título.pfc]
Archivo Sesión Buscar Ver ?
Visualización de las Tramas
Información de los Campos de las Cabeceras
IP
TCP
Aplicación
Datos ASCII-7

E/S	Orden	Tiempo (ms)	Dir. Origen	Dir. Destino	Tipo	Bytes
S	304	3730	00:50:04:35:35:F2	00:A0:24:38:38:B2	TCP	62
E	305	3730	00:A0:24:38:38:B2	00:50:04:35:35:F2	TCP	60
S	306	3730	00:50:04:35:35:F2	00:A0:24:38:38:B2	TCP	60
E	310	3840	00:A0:24:38:38:B2	00:50:04:35:35:F2	TCP	123
S	311	3840	00:50:04:35:35:F2	00:A0:24:38:38:B2	TCP	68
E	312	3840	00:A0:24:38:38:B2	00:50:04:35:35:F2	TCP	90
S	313	3840	00:50:04:35:35:F2	00:A0:24:38:38:B2	TCP	68
E	315	3900	00:A0:24:38:38:B2	00:50:04:35:35:F2	TCP	60
E	924	12250	00:A0:24:38:38:B2	00:50:04:35:35:F2	TCP	102
S	925	12250	00:50:04:35:35:F2	00:A0:24:38:38:B2	TCP	60
E	926	12250	00:A0:24:38:38:B2	00:50:04:35:35:F2	TCP	71

```
+OK QPOP (version 2.2) at montgo starting.  <1342.963398917@montgo>**
```

Guarde la sesión con la opción Archivo -> Guardar

Figure 9. Monitoring a POP3 session.


Analizador TCP / IP [Sin Título.pfc]
Archivo Sesión Buscar Ver ?
Visualización de las Tramas
Información de los Campos de las Cabeceras
IP
UDP
Aplicación
Datos ASCII-7

E/S	Orden	Tiempo (ms)	Dir. Origen	Dir. Destino	Tipo	Bytes
S	1134	14670	00:50:04:35:35:F2	00:60:97:B4:EE:83	UDP	72
S	1136	14670	00:50:04:35:35:F2	08:00:09:D5:70:68	UDP	72
E	1137	14670	08:00:09:D5:70:68	00:50:04:35:35:F2	UDP	308
S	1141	14670	00:50:04:35:35:F2	00:60:3E:10:09:32	ICMP	74
E	1152	14830	00:60:3E:10:09:32	00:50:04:35:35:F2	ICMP	74
S	1212	15660	00:50:04:35:35:F2	00:60:3E:10:09:32	ICMP	74
S	1321	17140	00:50:04:35:35:F2	00:60:3E:10:09:32	ICMP	74
S	1391	18680	00:50:04:35:35:F2	00:60:3E:10:09:32	ICMP	74
E	1408	18840	00:60:3E:10:09:32	00:50:04:35:35:F2	ICMP	74
E	2689	32680	08:00:02:03:81:68	00:60:97:BE:9D:1D	ICMP	98
E	3262	42620	00:60:97:B4:EE:83	00:50:04:35:35:F2	UDP	102

```
************www*ieee*org************M****sphinx5*ieee*org*********M******(*2******o<***auth01*
2*2******o<***dns*2*2******o<***ns*uu*net**2******o<***krypton*2*2******o<***deepthought*2*X
******H********m******H********|******s****'**********t***************t*******
```

Guarde la sesión con la opción Archivo -> Guardar

Figure 10. Ping example.

5. CONCLUSIONS AND FUTURE WORK

In this paper, a network monitor tool has been presented. This tool has been specifically developed for computer network courses, so educational objectives has been introduced in the design and development.

We use this tool at Computer Science Faculty of Technical University of Valencia in order to learn and debug TCP/IP network protocols.

The main objectives have been achieved and students have found the tool useful in their laboratory sessions. Some bugs have been corrected and a new version is under development with suggestions from students.

Even though the tool was designed to be used in Ethernet based networks, the used development environment and libraries allows its operation in other networks. We test it through a PPP phone connection and it works as fine as in Ethernet in all protocols above network level. The only protocol that will not work fine is ARP and the utilization graphic is not correctly calibrated.

The success of the tool has made us to consider adding more protocols and application decoders to it.

6. REFERENCES

[1] Manuel Pérez Malumbres, Román García García. *La arquitectura TCP/IP: ejemplos practicos sobre Windows95.* Universidad Politécnica de Valencia. ISBN 84-7721-595-2

[2] R. García, M. Pérez, J. Pons, *Redes1: Practicas Documentadas*, Servicio de publicaciones de la UPV nº 97.913

[3] D. E. Comer, *Internetworking with TCP/IP Vol. I: Principles, Protocols, and Architecture*. Prentice-Hall 1995, ISBN 0130183806

[4] D. E. Comer, *Internetworking with TCP/IP Vol. III Client-Server Programming and Applications-Windows Sockets Version*. Prentice-Hall 1997. ISBN: 0138487146

[5] W. Stallings, *Data and Computer Communications*. Prentice-Hall 1996; ISBN: 0024154253

[6] A. Tanenbaum, *Computer networks*, 3 Edition, Prentice Hall 1995.

[7] Computer Network Courses main page: http://www.redes.upv.es

[8] RFCs, Internet Request For Comments

[9] Chris Clap: Vpacket VxD developer. Camberra University