Observation and Analysis of Wireless Network Performance by an Intelligent Technology

Dr. Muhsin. J.Qubian
Amarah Institute of technology

Abstract: With the current concerns over computer networks management, there is a need for intelligent tools to monitor, analyze and manage computer networks.

This research presents the design of a mobile agent based network management framework (Traffic Watching) to monitor a network performance.

This mobile agent used in this work carries the code and the data with itself when the agent migrates among the nodes in the network.

This mobile agent is designed to capture internet and network packets that transferred to/from the Local Host and to/from the nodes in the network.

The information carried by the packet header is analyzed by the system to measure the rate of bandwidth used by the nodes in the network.

The statistical calculations which are done by the system are displayed in graphical and mathematical modes which give the system the easiest way to trace the network performance that can be analyzed and observed by Network manager .

The designed system can actively migrate among nodes in a heterogeneous networks environment. In addition, these data and forms gave the system best way to tackle its performance.

Due to using this agent, the network management policy has been changed and adapted quickly to the changes of today dynamic network environment.

This work has been implemented on a Pentium 4 PC, and programmed with C++.

3. Introduction: The number of users using networks is increasing exponentially every day, as a consequence, many managers try to observe and control the network remotely.

Network Management means different things to different people. In some cases, it involves a solitary network consultant monitoring network activity with an outdated protocol analyzer. In other cases, network management involves a distributed database, auto polling of network devices, and high-end workstations generating real-time graphical views of network topology changes and traffic. In general, network management is a service that employs a variety of tools, applications, and devices to assist human network managers in monitoring and maintaining networks [1].

Network management has thrived on centralized or weakly distributed hierarchical models for many years. Soon after the advent of open systems in the second half of the 1980's, proprietary solutions gradually gave way to two open protocols, Simple Network Management Protocol (SNMP) and Common Management Information Protocol (CMIP, in the first half of the 1990's [2]. These protocols primarily addressed what was then perceived as the most critical feature lacking in existing network management systems: interoperability between multiple vendors. SNMP was widely adopted by the IP world to manage Local Area Networks (LANs), Wide Area Networks (WANs), Intranets, etc. In parallel to this wide-scale development, CMIP, richer but more complex than SNMP, found a niche market in the telecommunications world, as the International Telecommunication Union, Telecommunication Standardization Sector (ITU-T) decided to adopt the Open Systems Interconnection (OSI) model as the basis for its Telecommunications Management Network (TMN) model. The use of both SNMP and CMIP has been questioned in the recent past, together with their common underlying models. Why is it that more and more network administrators are now demanding Distributed Network Management (DNM), when the same people were happy with centralized or weakly distributed hierarchical models a couple of years ago? What triggered this sudden and massive shift toward DNM?

Network management can be defined as OAM&P (operations, administration, maintenance and provisioning.[3]) of network and

services Although the term "network" seems restrictive, it is very convenient to use, as the backbone for all the Information Technology services is still the computer network.

In the rapid pace of growing technology, networks tend to be large and complex, filled with many types of equipment from different vendors. Managing such a network is increasingly more difficult, involving multiple management tools and protocols to support different proprietary devices on the network. The goal of network management is to ensure that users of a network receive the information technology services with the quality service that they expect. Therefore, the network administrator needs a network management tool that can monitor the network's availability, utility and performance with the most industry-acceptable standards as possible to reduce the conflict of the network management system.

4. Intelligent Mobile Agent: The two main communication protocols used for Network Management, SNMP and CMIP, are characterized by centralization, which is usually considered the source of a low degree of flexibility and scalability in large systems [4]. A network manager residing on a central station contains most of the management logic and processes the data collected from physically distributed agents. The agents are rigid servers that are closely associated with the hosting network components. The involved management paradigm concentrates most of the processing into a single manager that usually interacts with a large number of agent servers. Hence, these two protocols do not provide the required scalability that is needed in today's predominantly complex networks due to the large number of network components, vast topologies, unpredictable network dynamics, etc.

The basic idea to solve these problems is to bring management intelligence as close as possible to the managed resources. One of the most prominent techniques providing a solution is Management by Delegation (MbD) [5]. It represents a clear effort towards decentralization and increased flexibility of management functionality. Instead of the traditional methods of exchanging client/server messages, the management station can specify a task to be carried out by locating a program (an agent) on involved devices, where the actual execution of the task takes place. Such execution is completely asynchronous, producing a higher degree of parallelism, thus enabling the management station to perform other tasks. The

management station action of delegating a specific function to a remote process is described as *function delegation*; this *autonomy* is what determines the *agency* of the remote program.

Another property that enhances the functionality of the Intelligent Mobile Agent is the ability to travel from one node to another. This capability is termed *mobility*. Code Mobility is defined to be "*the capability to reconfigure dynamically, at runtime, the binding among software components of applications and their physical location within a computer network*" [6]. The roaming program is termed a Mobile Agent.

The third capability from which Network Management systems can benefit is the integration of Artificial Intelligence technologies, which capture the human manager expertise in solving network problems. Thus, the management station delegate's tasks to autonomous intelligent code that is capable of making independent decisions on the manager's behalf. The decisions are based on the data that the code analyzes locally at the hosting nodes without human manager involvement.

Combining these three capabilities (i.e., agency, mobility, and intelligence) is a software entity coined as **Intelligent Mobile Agent**. Through the integration of the three properties, many of the Network Management activities including fault management, performance analysis and configuration management can be automated.

The main goal from designing the proposed system is to capture the internet and internal network packets along the network and monitoring them, the system work is applied to the internet nearly real time, not a simulated system, besides it can work with different kinds of network hardware (Ethernet cards, and modems). The data flow diagram is shown in Figure 1.1.

User Mode

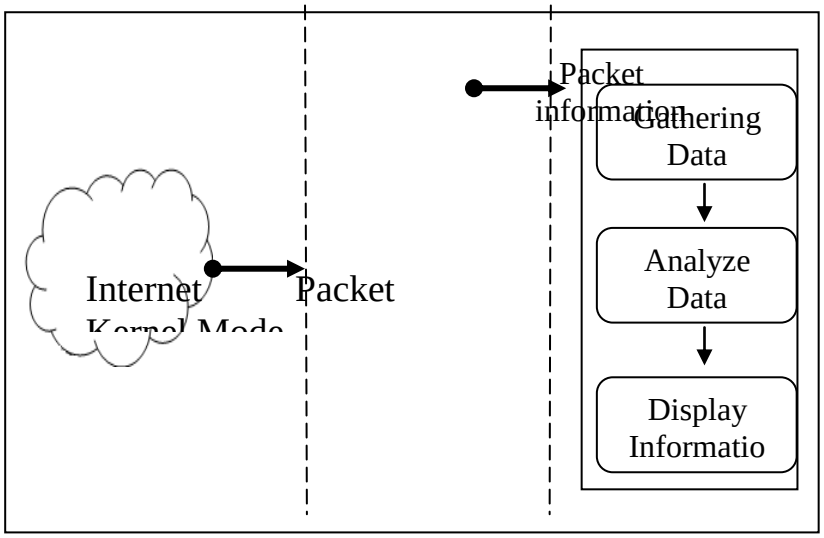

Figure 1.1: The System Data Flow.

5. System Specification

In the design and implementation of the proposed system we need the following specification:

6. Hardware and software requirements:

(See figure 1.2).

1- 3 PC's type Pentium 4, with the following requirements (Processor 1.7 GHz, RAM 512MB, Speed 333 MHz).
2- 3 LAN cards type Ethernet based 10/100 Mbps.
3- Unshielded Twisted Pairs (UTP) cables.
4- Hub base 10/100 Mbps.
5- Internet connection.

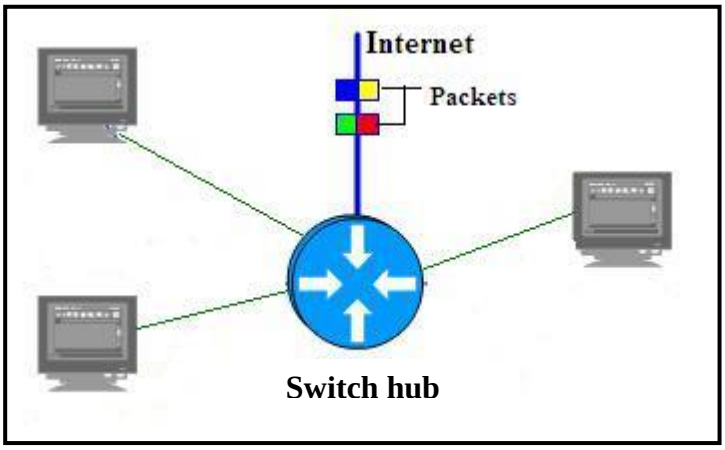

Figure 1.2: System Hardware

1- Microsoft Windows XP Professional.
2- Microsoft Visual C++ Dot Net.
3- Windows 2000 SDK.

7. System in Use :

The proposed system operation starts by initializing the network services and loading all the drivers that will be needed for packets capturing in the kernel mode (such as the Winsock2). See Figure 1.3.

```
//check if winsock2.x is available
   WSADATA WinsockData;
   if( WSAStartup(MAKEWORD(2,2), &WinsockData) != 0)
   {
       AfxMessageBox("This program requires Winsock
2.x", MB_ICONHAND );
           return FALSE;     //quit program
   }
```

Figure 1.3: Checking the Drivers in the Kernel Mode.

The system will need then to get a real network connection (assigned IP address, gateway address, and the DNS). See Figure 1.4.

```
BOOL CPacket::Open(int i, DWORD bufsize, DWORD kernelbuf
, BOOL promiscuous)
{     if (i<0)
      {
            //search for an adapter and use the first
 valid found
            i=0;
while (!IsValidIPAdapter(i)&&(i<=nAdapterCount))
                  i++;
      }
      //check if given Adapter is available
      if (i>=nAdapterCount)
      {     AfxMessageBox("no valid adapter found!");
            return FALSE;
      //invalid adapter number or not initialized yet
      }
      nActiveAdapter = i;
      //save the number of the selected adapter

      //try to open the adapter
      Adapters[nActiveAdapter].pAdapter                  =
OpenAdapter(nActiveAdapter);              //open      the
adapter
      if (Adapters[nActiveAdapter].pAdapter == NULL)
      {
            AfxMessageBox("could  not  open  Adapter  in
CPacket::Open()");
            return FALSE;                        //could
not open adapter
}
```

Figure 1.4: The Source Code for Initializing the Network Adapter.

The system must be informed whether the packets capturing is for the Local Host or for the Local LAN, this is done by giving a value to the flag (m_local). This value is either “TRUE” or “FALSE” (TRUE= Local Host, FALSE= Local LAN).

When an Internet packets is captured, the system will save packets in a buffer in order to recognize them to see if it is an IP or a networked packet, then it categorizes them into six categories (Web, Mail, File sharing, News, UDP, Other), depending on its recognition on the port that being used by the packets (TCP→ Use port 6, UDP→ Use port 17, News→ Use port 80 and 443). See Figure 1.5a for the source code and Figure 1.5b for the flow chart[7].

```
//we are the client, using the awayport
            case 119:   //nntp
            case 563:   //nntp over ssl
                  m_services[S_NEWS] += (DWORD)len;
                  break;
            case 80:
            case 443:   //http over ssl
                  m_services[S_WEB] += (DWORD)len;
                  break;
            case 25:    //smtp
            case 109:   //pop2
            case 110:   //pop3
            case 143:
            case 220:
            case 585:
            case 993:
            case 995:   //pop3 over ssl
                  m_services[S_MAIL] += (DWORD)len;
                  break;
```

Figure 1.5a: Part of the Source Code of the System Mechanism.

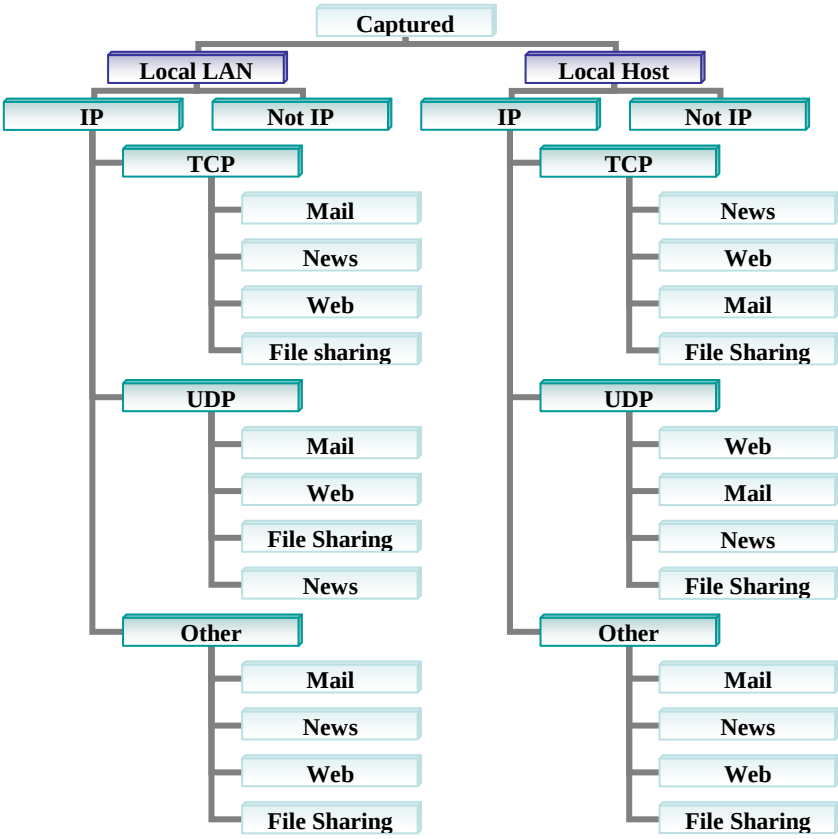

Figure 1.5b: The Flow Chart of the System Mechanism.

The system is designed not to send copy of the packet to the application, instead it sends the buffer address, reads packet information header, and then analyzes them to get statistical information on the amount of bandwidth that each category uses [8].

8. Results and discussion

Applications using mobile agent technology in network communication shows that mobile agent can improve significantly network performance and is very useful for distributed applications.

The system is composed of four main windows, "Current Traffic, Detailed Traffic, History, and Options". The following is a brief description of these windows (see figure 1.6).

8.1 Current Traffic

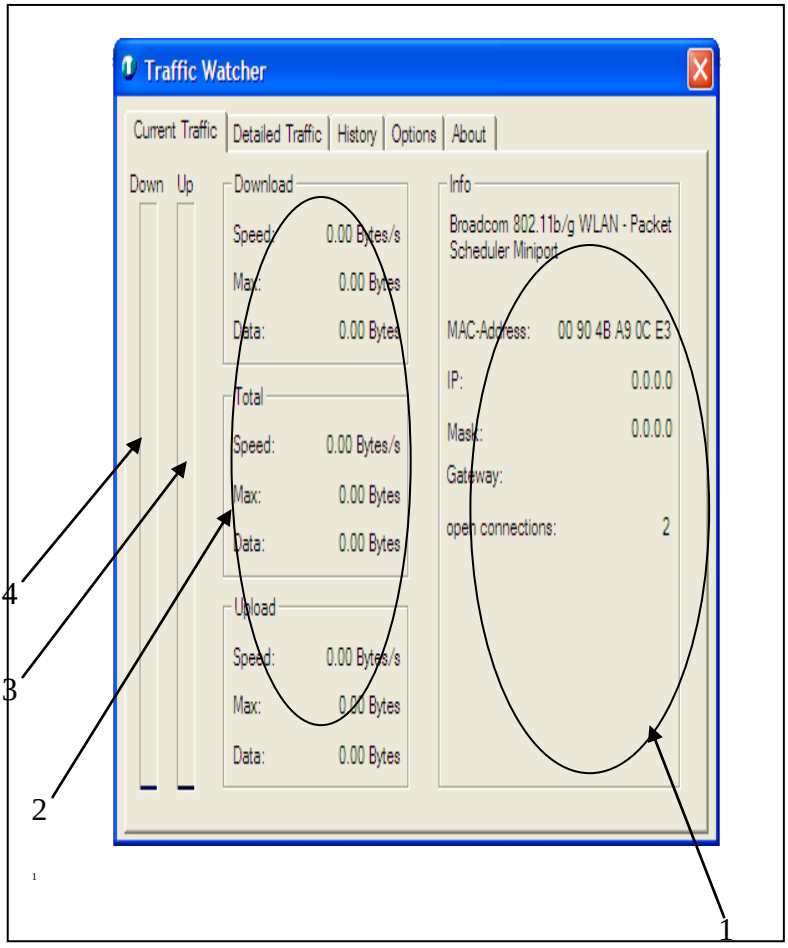

Figure 1.6: Current Traffic Window

1: Contains the information about the Local Host. This information is: the IP address of the Local Host, the sub net mask, the Gateway address, and the number of the connections (adapters). The Local Host must have a fixed IP address and gateway address [9].

2: Represents the speed of the bandwidth and the maximum amount of bytes downloaded in the computer. This amount is calculated by the number of packets that arrive to the Local Host. Also it contains the speed of the uploaded data; this is measured by the number of packets transfer from the Local Host. Finally, it will measure the maximum amount of the download and the upload, measures the total bandwidth for the downloaded data and the uploaded data.

3, 4: Display graphically the amount of download and upload; represents by a bar divided into two sections, the "Down" and the "Up". "Down" bar will increase in length at the arrival of the packets from the Local Network to the Local Host (Downloading), and the "Up" bar will increase in length whenever the Local Host sends packets to the network (Uploading).

At the first startup of the program all this information (point one to four above) are zero. See Figure 1.7.

When the Local Host is connected to the internet or to the local network the system starts to read the captured packet header information, like its IP address and the subnet mask… etc, so when the nodes in the network start to download or upload information from the internet or transfer information among them, the counters will increase and it will give the rate of using the bandwidth in the network as follows. See Figure 1.7.

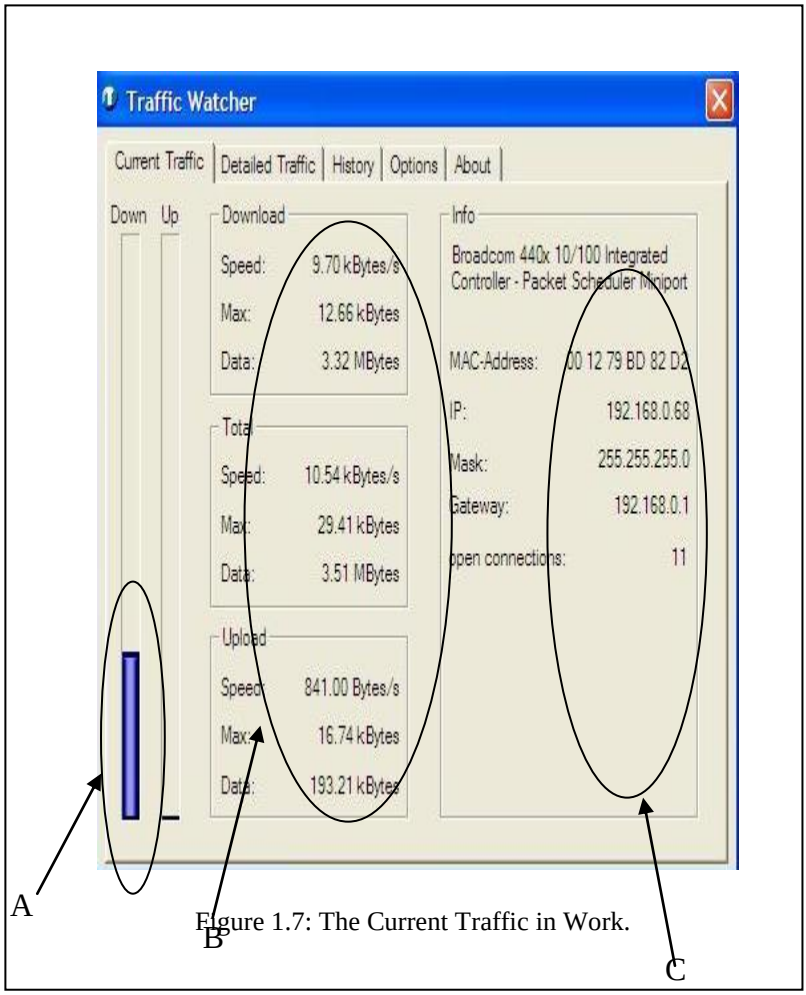

Figure 1.7: The Current Traffic in Work.

Figure 1.7: the regions A, B, and C refer to:

A: gives the amount of the downloading in the network graphically.

B: gives the values of the network speed and the maximum for both download and upload and the total for them.

C: is the information of the Local Host such us its MAC address, IP address, Subnet Mask, and the Gateway address. Also, it gives the number of the connections in the network (the number of the adapters in the network).

8.2 Detailed Traffic

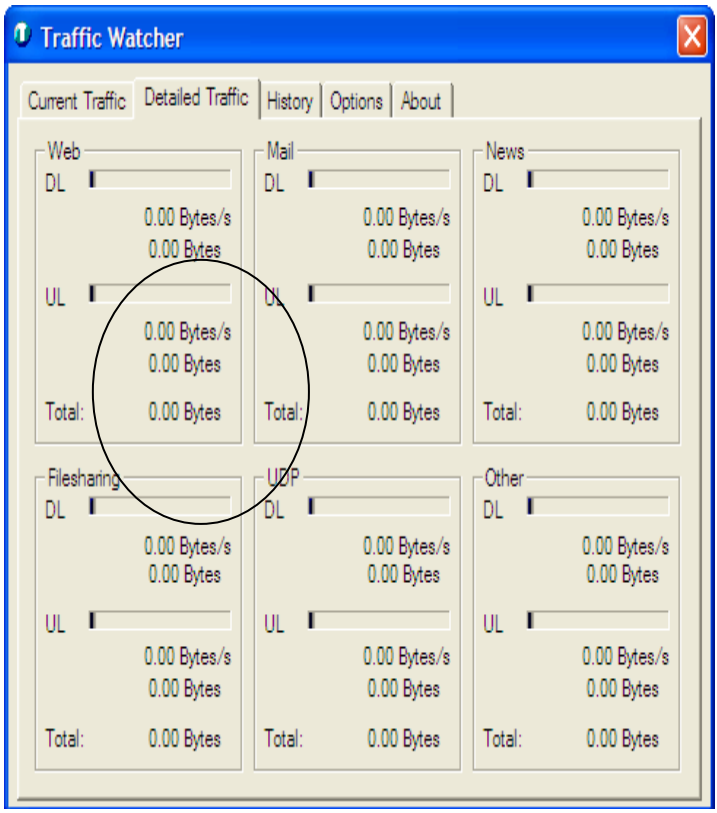

Figure 1.8: The Detailed traffic Window on the initial startup of the System

As you see in the Figure 1.8 above, at the initial start up all the accounts are zero's and as soon as the connection starts the system starts to calculate all the values of the six categories, to see which one has the load on it. Figure 1.9a and Figure 1.9b show the upload and the download for the six categories in a certain day [10]. This system is applied to a network connected to the internet, and you can notice the load is in the web category.

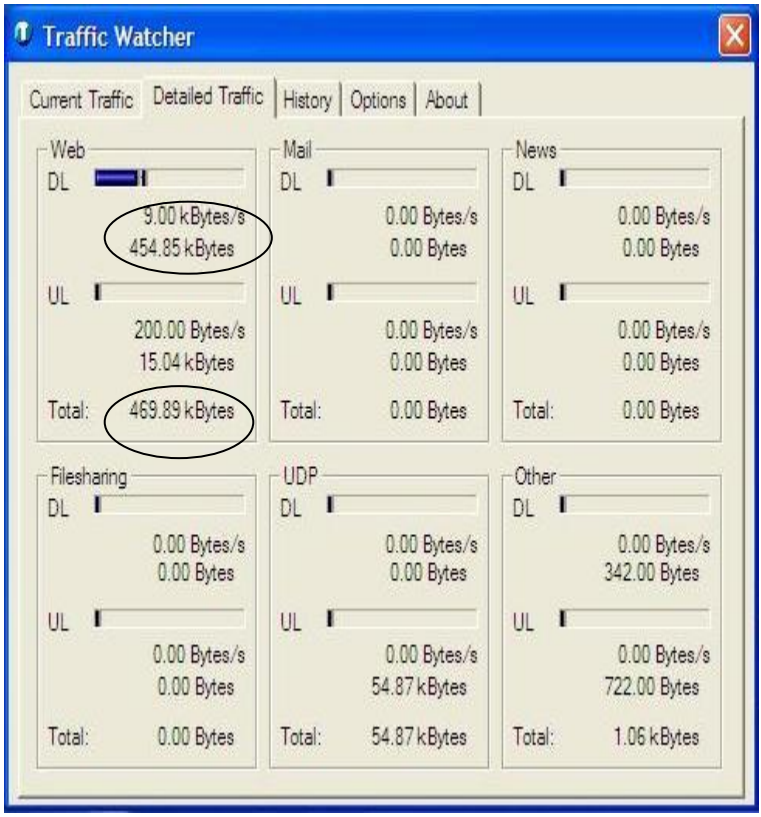

Figure 1.9a: The Upload and the Download for a network connected to the Internet.

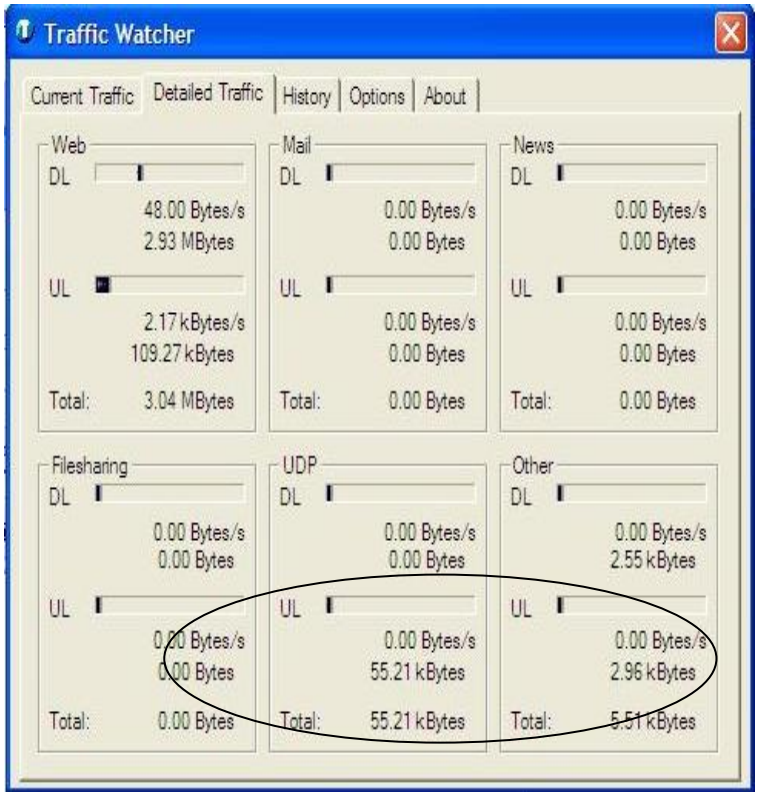

Figure 1.9b: The Upload and the Download for a network connected to the Internet.

In Figure 1.9b the use of the UDP protocol increases and also there is another protocol used by the network, you can see it in the Other category [11].

8.3 History Window

This window illustrates the history of the use of these categories, it saves the information for every day and adds it to the next day until the end of the month it will clear the days bar and starts again. Also, it saves the information of each month and adds it to the next month, then clear the month bar at the end of the year.

Figure 1.10 shows the history of Tuesday "June 27 2006".

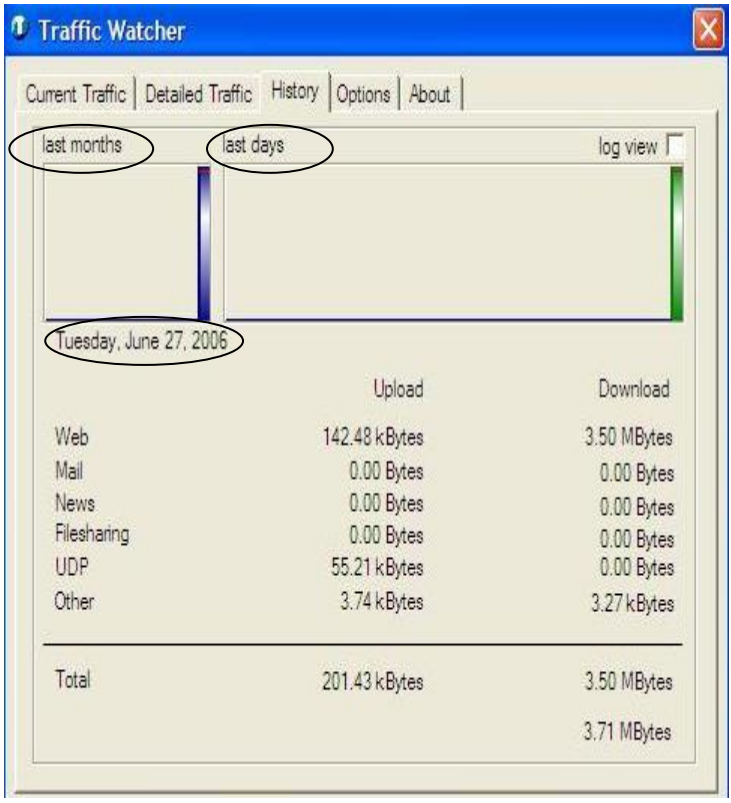

Figure 1.10: The History of Capturing Packets for "June 27 2006".

8.4 The Option Window

From this window the user can choose the maximum rate of download and the maximum rate of upload for the Local Host, and the transfer rate among the computers in the network.

Also the user can choose the way that the system starts and its life time. The user can also operate the system at the initialization of the windows by checking the check box "Start with Windows". for example, the user can choose to run the system in the log on of the windows (start with windows). The user can also see the values in bits instead of bytes [12]. This window gives the user the facility to choose his adapter in case he has more than one adapter. See Figure 1.11.

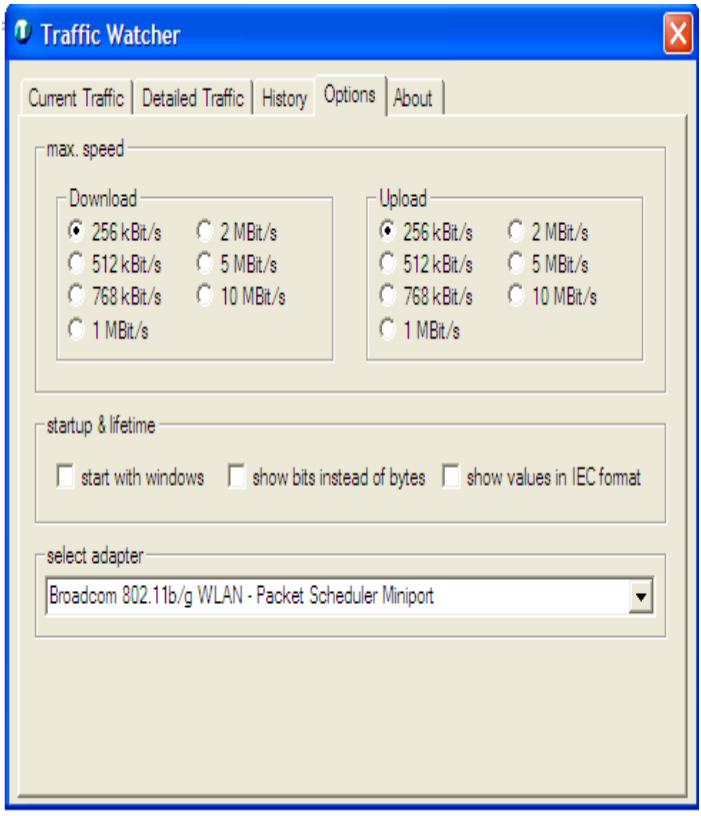

Figure 1.11: The Option Window

9. References

[1] John D. Musa, Anthony and Kazuhira Okumoto, "Software Reliability: Measurement, Prediction, and Application", Cambridge University Press, 2005.
[2] Peng Xu, Ralph Deters, "MAS and Fault Management", Handbook of IEEE, Oxford University, 2005.
[3] M. G. Ban, "Fault Identification in Computer Networks: A Review and a New Approach",
Kyoto, Japan, April 15-18, 2004.
[4] CISCO, "Network Management System: Best Practices White Paper", IEEE, Infocom, 2005.
[5] Man F. J., "Fault Management: A Functional View of Root Cause Analysis and Correlation", www.acm.com.
[6] Edidiong Uyai and behrouz Homayoun, "Distributed Fault Management using Intelligent Software Agents", Mc Graw Hill, 2005.
[7] Abeck, Andreas Koppel, Jochen Seitz, "A Management Architecture for Multi-Agent Systems", McGraw – Hill, New York, 2004.
[8] Kim and Peter Mellquist, "SNMP++", Banker, Spring Verlag , New york, 2005.
[9] Jeffrey D. Case, Mark S. Fedor, Martin L. Scho_stall, and James R. Davin, "A Simple Network Management Protocol (SNMP), Prentice Hall, 2004.
[10]. Carzaniga, A.; Picco, G.; and Vigna, G., "Designing Distributed Applications with a Mobile Code Paradigm". International Conference on Software Engineering, Boston, May 1997.
[11] Robin J., "Computational Intelligence and Knowledge", Prentice Hall, 2006.
[12] Mohsen Kahani, H.W. Peter Beadle, "Decentralized Approaches for Network Management: Intelligent Agents".

استخدام وكيل متنقل معدل في إدارة شبكة محلية

د. محسن جبار كبيان
المعهد التقني في العمارة

الملخص :

مع تطور علوم شبكات الحاسبات والاهتمام المتزايد عليها، دعت الحاجة لاستخدام أساليب ذكية لإدارة شبكات الحاسوب ومراقبتها وتحليل النتائج.

البحث يقترح تصميم وكيل متحرك (نقال) يعمل في إطار إدارة شبكة حاسوب ومتابعة أداءها (تتابع مرور حزم الشبكة).

الوكيل الذكي المستخدم في هذا العمل ينقل الرمز والبيانات للمعلومات أثناء تنقله بين عقد وأجزاء الشبكة.

صمم هذا العميل لالتقاط حزم الشبكة أو حزم الانترنيت المنتقلة من والى المضيف المحلي (الخادم) ومن والى عقد الشبكة (الحاسبات المربوطة في الشبكة) وبالتالي فان المعلومات المحمولة بواسطة هذه الحزم تحلل بواسطة هذا النظام لتحسب معدل عرض الموجة للحزم الموجودة على كل حاسبة في الشبكة.

والنتائج التي تم الحصول عليها من خلال حركة العميل الذكي عرضت بشكل أو نمط صوري ورياضي على شكل واجهة يمكن تحليل نتائجها وتمييزها من قبل مدير الشبكة (أو حتى المستخدم) لدراسة أداء وإدارة الشبكات تلقائيا من خلال هذه الإحصاءات والبيانات والأشكال وبالتالي أعطت النظام طريقة سهلة لمتابعة أداء الشبكة.

وباستخدام هذه الطريقة قد تغيرت سياسة إدارة الشبكة بصورة تلقائية وذكية تبنت إدارة الشبكة لتواكب التغيرات السريعة في بيئة إدارة الشبكات.

نفذ العمل بهذا النظام باستخدام حاسبات بنتيوم ٤ وتمت البرمجة باستخدام لغة c++.