Republic of Iraq
Ministry of Higher Education and ScientificResearch
Al-Nahrain University
College of Science

Construct a Monitor System For Multi nodes

A Thesis
Submitted to the College of Science, Al-Nahrain University in Partial Fulfillment of the Requirements for The Degree of Master of Science in Computer Science

BY
Shayma Mohammed Reda Hamandy
(B.Sc. 2002)

SUPERVISORS

Dr. Lamia H. Khalid **Dr. Sawsan K. Thamer**

٢٠٠٦ ١٤٢٧

Abstract

To guarantee using the computers on the network in a right manner in colleges, companies and other network places, it is important to use remote monitoring systems to observe users and protect any computer from misuse. In addition when the users know that they are monitored this will lead to improve there performance.

This work concerns the implementation of multi-node remote monitoring system MRMS (online and offline monitoring) in a Local Area Network (LAN) environment. In online monitoring one or more remote PCs in the network can be monitored with some control on there activities. The proposed system provides information about users activates on a remote Personal Computers (PCs) by monitoring the performance of the keyboard, mouse, user screen events, and programs that are running by the users. The system allows the Administrator to control the remote computers by sending warning messages, stopping any illegal program, or shutdown the computer. In offline monitoring any illegal process is blocked by the system according to a list specified by the Administrator also all the information about the keyboard and processes of the remote PCs are saved on a hidden files on the host computers. The Administrator can request any information from these files on the host computers.

Acknowledgment

First of all, Praise is to my God who enables me to achieve this work.

I would like to express my sincere gratitude and appreciation to my supervisors ***Dr. Lamia H. Khalid*** and ***Dr. Sawsan K. Thamer*** for their able guidance, supervision and untiring efforts during the course of this work.

I would like also to express my thanks to the college of science, dean of the collage, **Dr. Taha S. Bashaga** the head of computer science department and all staff and friends for the continuous support and encouragement during the period of my study.

My special thanks to my Husband, for always being there for me, and his continual love, support, and patience.

I wish to thank my family specially my mother for their continuous support and encouragement during the period of my study.

Finally, I would like to thank all the people that give me some kind of help to continue this work.

Chapter One
Introduction

1.1 Introduction to Network Monitoring

Management of computing resources in an enterprise network is becoming an increasingly complex task because many device components are frequently added, upgraded, and replaced. Performance Management is the process of planning, defining, measuring, analyzing, reporting and tuning the performance resources including data networks, hardware, operating systems, applications and services.

Performance management of a computer network comprises two broad functional categories: monitoring and controlling. The monitoring function tracks activities on the network. The controlling function enables performance management to make adjustments to improve overall network performance. System monitoring tends to be an important part of management goals. Monitoring includes a range of activities: check for system component failures, configuration errors, overload conditions, auditing of usage, and intrusion detection [Cis06].

Many large-scale computing environments, such as university campus networks, tend to be relatively open without any firewalls or even physical barriers in accessing computers in public-labs. An intruder can access a computer over the network quite easily, or physically reboot a machine. In such large, open environments it is desired that a system administrator can actively monitor all nodes for suspicious activities. Besides network management needs, monitoring services are also required by many of today’s emerging applications [Ana02].

Monitoring software is legal because the vast majority of its developers are law-abiding people who create their programs exclusively for legitimate purposes. There are many situations when monitoring the computer activity is perfectly legal: the parents can use key loggers to protect their children from online abuse, the companies may use monitoring software to ensure that their employees don't misuse corporate internet connection and so on . Most typically monitoring software is any product that is installed on a computer that records the user's activity without the user having any knowledge of the monitoring session like spy software which can capture chats, instant messenger conversations, email, desktop activity. Monitoring software gives the ability to spy on computers, record keystrokes and many other functions [Joh04].

Network monitoring is the act of recording all computer and internet activity that occurs on the PCs in the network, including all Web surfing, email, and instant messaging, in order to provide management with detailed information regarding agent performance **[**Der03**]**.

The monitoring can be considered as a one of the security mechanisms since it is responsible for watching or observing users behavior. By monitoring changes, the administrator may be alerted before the system crashes, stops responding, or attacked, which allows the administrator more flexibility in mitigating the risks of its occurring. The benefits of security monitoring is the early identification of wrongdoing or new security vulnerabilities. This early identification can help to block wrongdoing or venerability before harm can be done or at least to minimize the potential impact [Kel02,Off04].

In general the monitoring application distinguish two types of monitoring: online and offline. Online monitoring allows observation and potentially control of an application at run-time. In case of online monitoring, a monitoring application poses real-time constraints on the overall time it takes to generate process, disseminate, and present monitoring data. In case of Offline monitoring, a monitoring application poses no such time constraints. Offline monitoring code is executed with a different process, potentially on a different machine [Ben04, Nik04, Wil03].

1.2 Literature Survey

Various efforts in the field of monitoring system were introduced, some of these efforts are summarized below:

- ***Hisham Mohamed[His02]*** Remote performance monitoring system was designed and implemented which enables network administrator to watch remote personal computer (PC) desktop screen, mouse motion, and keystrocks. In addition, sending warning messages to remote PC's user and have some control to logoff windows. It is not possible to get information from remote PC unless there is an agent installed on that PC to send the needed information through the network connection. The administrator can monitor only one computer at the time, and has only online monitoring; this system is accomplished using combination of C++ and Java programming languages.

- ***Manar Saad Salih Al-Taie [Man02]*** developed a type of malicious code called Remote Access Trojan (RAT) works under windows environments. RAT provides an access to remote computer and can be used in two ways, first as an attacking tool, where the attacker can perform malicious functions on victim computer, second as a monitoring tool by getting information of desktop screen of the remote PC. It can show only one screen shot at a time and send the agent program through email using Simple Mail Transfer Protocol (SMTP) which make it independent from any user email agent; this system is accomplished using Java programming language.

- ***Dalal Naeem Al-Zaidi[Dal04]*** Developed two types of target monitoring works under windows environments. These are Online Monitoring and Offline Monitoring. In online monitoring the screen picture, mouse motion, and keystrokes can be monitored. The administrator can send warning messages to the remote computer user. Offline monitoring limited to the keyboard only. The administrator can monitor only one computer at the time and the software must installed in each side without having the ability to migrate it, and because of the big size of the screen image, it must divided in to small packets and send them one after another; this system is accomplished using Visual Basic programming language.

- ***Remote Spy Software [Off04]*** It is a remote PC monitoring software which gives the power to monitor computers remotely without the need of actual physical access. Simply send the remote spy software module to a remote PC via email and begin monitoring. It monitors chats, instant messengers, chat rooms and keystrokes.

- ***Network LookOut Administrator [Net06]*** An application for monitoring remote computers, It allows to see live screens of remote computers, has administrator and client sides, this application can monitor more than one computer without having any control on them, and has only online monitoring.

1.3 Aim of Thesis

The aim of this thesis is to construct a multi-node remote monitoring system that performs the following tasks:

- provides information about users activities on a remote PCs by monitoring the performance of keyboard, mouse, desktop events, and programs that are running by the users .
- provides some control by stopping any illegal process, send warning message, logoff or shutdown the computer.
- protects any computer in the LAN from misuse, or avoid any attempt of misuse by providing a firewall against illegal programs.
- Offline monitoring of any remote PC by storing the performance of the keyboard and the running programs.

1.4 Thesis Outline

This thesis contains the following :

- **Chapter Two**

  This chapter covers the theoretical basics of networking, network monitoring, and types of monitoring.
- **Chapter Three**

  This chapter presents the proposed system architecture, and the algorithms that used to implement this system.
- **Chapter Four**

  This chapter presents the user interface and the testing results for the proposed system.
- **Chapter Five**

  This chapter explores conclusions of this work, and the suggestions for future work.

Chapter Two

Network Monitoring Concepts

2.1 Network System

Networks are connections between groups of computers and associated devices that allow users to transfer information electronically. These connections can be direct (through cables) or indirect (through a modem); and classified into several types depending on inter-processor distance: Local Area Network (LAN), Metropolitan Area Network (MAN) and Wide Area Network (WAN) as shown in table 2.1[Tan03].

Networks can allow computers, servers, and other devices to talk to each other. The architecture of a network refers to the way in which computers participate in a network. The type of architecture chosen by an organization may include geographical location, number of users, the requirements of specific application packages, level of technical support available, existing systems, and of course cost. Network Architecture issues include determining how data will flow between the computers on a network. The connection of two or more networks is called an internetwork [Enc05].

Table 2.1 Classification of interconnected processors by scale

Interprocessor distance	**Processors Located in same**	
1 m	Square meter	Personal area network
10 m	Room	Local area network
100 m	Building	
1 km	Campus	
10 km	City	Metropolitan area network
100 km	Country	Wide area network
1000 km	Continent	
10,000 km	Planet	The Internet

Personal Area Networks (PANs) is short-range data communications systems that are primarily used to interconnect peripheral equipment (such as a mouse or keyboard) with a local computer or computing system. LANs are designed to reliably transfer large amounts of data quickly and error-free over a very small area such as an office. MAN’s facilitate LAN-to-LAN information exchange in a local telephone exchange area. The use of a WAN allows for information to be exchanged between LAN’s located at significant distances from each other [Law03].

1. Local Area Network (LAN) :

LAN is a collection of interconnected computers that can share data, applications, and resources, such as printers. Computers in a LAN are separated by distances of up to a few kilometers and are typically used in offices or across university campuses. A LAN enables a fast and effective transfer of information within a group of users and reduces operational costs.

Figure 2.1 shows several of the most popular LAN topologies and their configurations. Some data networks are setup as bus networks (all computers share the same bus), as star networks (computers connect to a central data distribution node), or as a ring (data circles around the ring) [Law03].

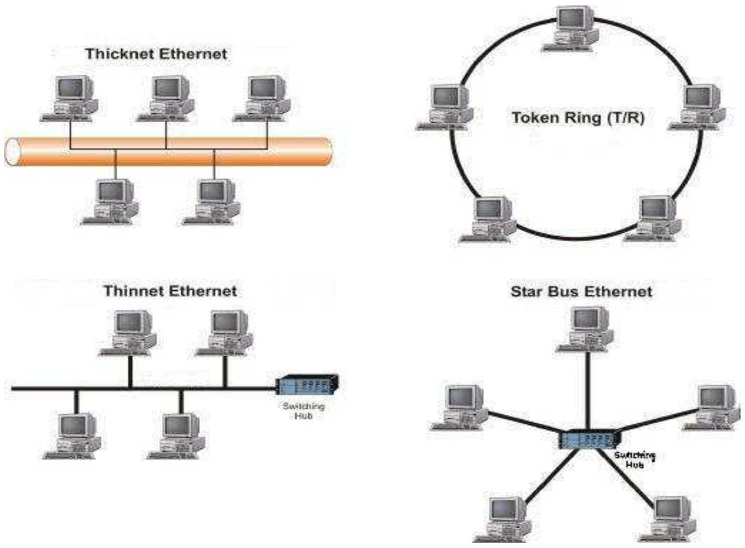


Figure 2.1 Local area network

Besides operating in a limited space, LANs include several other distinctive features. LANs are typically owned, controlled, and managed by a single person or organization. Most LANs are built with relatively inexpensive hardware such as Ethernet cables, network adapters, and hubs. Wireless LAN and other more advanced LAN hardware options also exist [Wik05-a].

Specialized operating system software may be used to configure a LAN. For example, most flavors of Microsoft Windows provide a software package called Internet Connection Sharing (ICS) that supports controlled access to LAN resources.

The most common type of LAN is an Ethernet LAN. The smallest home LAN can have exactly two computers; a large LAN can accommodate many thousands of computers [Bra04].

2. Metropolitan Area Network (MAN):

MANs are large computer networks usually spanning a campus or a city. They typically use wireless infrastructure or optical fiber connections to link their sites. For instance a university or college may have a MAN that joins together many of their LANs situated around site of a fraction of a square kilometer. Then from their MAN they could have several wide area network (WAN) links to other universities or internet [Wik05-b].

There are three important features which discriminate MANs from LANs or WANs [Gor01]:

- The network size falls intermediate between LANs and WANs. A MAN typically covers an area of between 5 and 50 km diameter. Many MANs cover an area the size of a city, although in some cases MANs may be as small as a group of buildings or as large as the North of Scotland .
- A MAN (like a WAN) is not generally owned by a single organization. The MAN, its communications links and equipments are generally owned by

either a consortium of users or by a single network provider who sells the service to the users. This level of service provided to each user must therefore be negotiated with the MAN operator, and some performance guarantees are normally specified .

- A MAN often acts as a high speed network to allow sharing of regional resources (similar to a large LAN). It is also frequently used to provide a shared connection to other networks using a link to a WAN.

A typical use of MANs to provide shared access to a wide area network is shown in the figure 2.2 [Lam03].

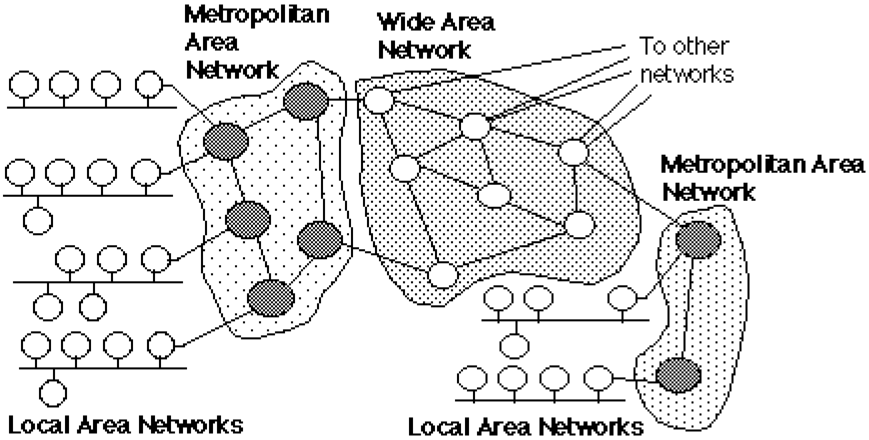


Figure 2.2 Metropolitan Area Network

3. Wide Area Network (WAN):

WANs are networks that span large geographical areas, such as a state, province or country. WANs often connect multiple smaller networks, such as LANs or MANs. Computers can connect to these networks to use facilities in another city or country [Enc05].

Several LANs can be linked together so that computers in one LAN can exchange data with computers in another LAN. An example of a WAN can be an education group wishing to connect all its schools over a wide area by one network. To do so, each school's LAN is interconnected via dedicated lines offered by the telecommunication company as shown in figure 2.3 [Lam02].

A WAN is different from a MAN because of the distance between each of the networks. In a WAN, one network may be anywhere from several hundred miles away to across the globe in a different country [Hop05].

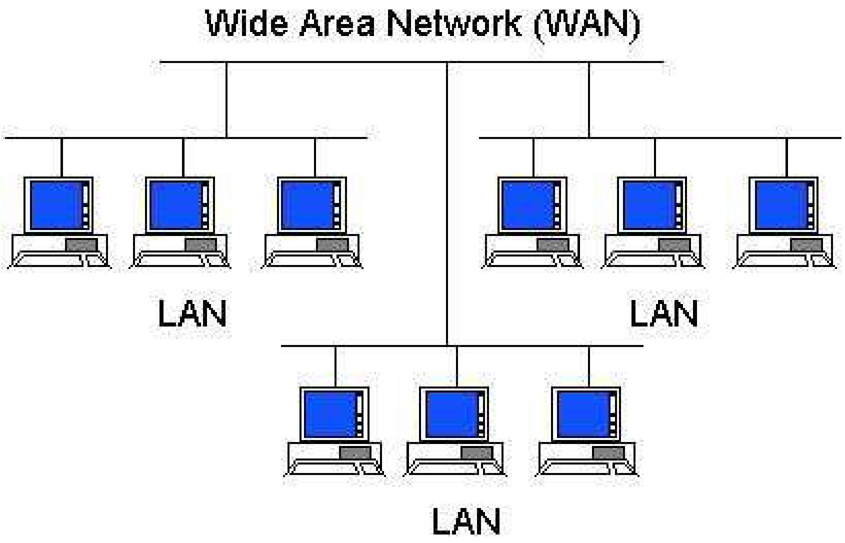


Figure 2.3 Wide Area Network

2.2 Network Transmission Techniques

Three types of transmission technology could be used with network system: Client/Server Networks, Three-tier Networks and Peer-to-Peer Networks [Enc05].

1. Client/Server (two-tier) Architecture:

A network architecture in which each computer or process on the network is either a client or a server. Servers are powerful computers or processes dedicated to manage disk drives (file servers), printers (print servers), or network traffic (network servers). Clients are PCs or workstations on which users run applications. Clients rely on servers for resources, such as files, devices, and even processing power .

The client component is a complete; standalone personal computer offers the user its full range of power and features for running applications. The server component, which can be another personal computer, minicomputer, or a mainframe, enhances the client component by providing the traditional strengths offered by minicomputers and mainframes in a time-sharing environment: data management, information sharing between clients, and sophisticated network administration and security features. The general structure of the client\server system is shown in figure 2.4 [Flo05].

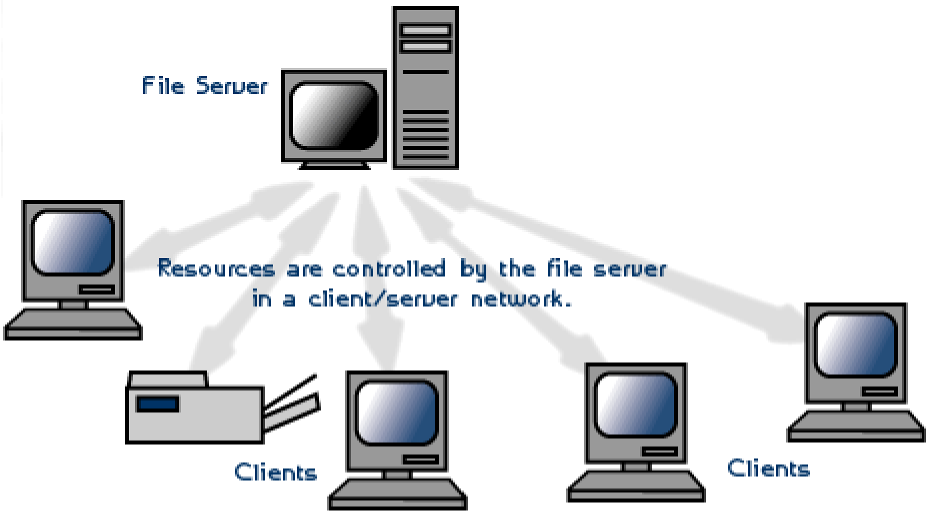

File Server
Resources are controlled by the file server in a client/server network.
Clients
Clients

Figure 2.4 Client/Server Architecture

The server is called "dedicated" because it is optimized to serve requests from the "client" computers quickly. A server is simply a computer that is running software that enables it to serve specific requests from other computers “clients".

A server provides many benefits including [Rus03]:

- **Optimization:** server hardware is designed to serve requests from clients quickly.

- **Centralization:** files are in one location for easy administration.

- **Security:** multiple levels of permissions can prevent users from doing damage to files.
- **Redundancy and Back-up:** data can be stored in redundant ways making for quick restore in case of problems.

Client-server architectures are called two-tier architectures in which the user interface runs on the client and the database is stored on the server. The actual application logic can run on either the client or the server [Van05].

2. Three-tier Architecture:

A special type of client/server architecture consisting of three well-defined and separate processes, each running on a different platform as shown in figure 2.5 [Thi06]:

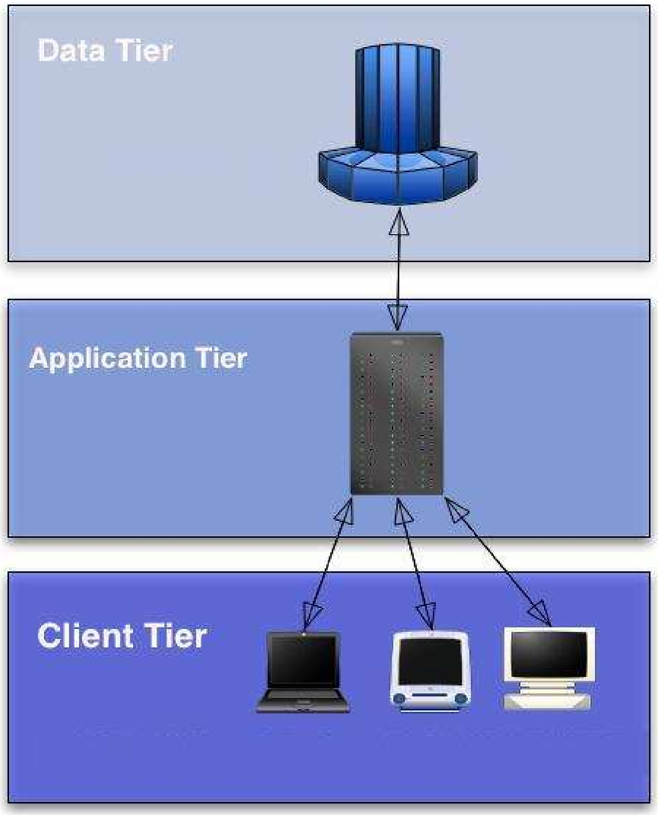


Figure 2.5 Three Tier Architecture

- The user interface, which runs on the user's computer (the client).
- The functional modules that actually process data. This middle tier runs on a server and is often called the application server.
- A DataBase Management System (DBMS) that stores the data required by the middle tier. This tier runs on a second server called the database server.

The three tier client-server model of networking is the way to go for larger organizations. It should provide the network with more flexibility as the needs to change. For example, as network traffic increases, there is the ability to add another server to handle the additional load. Also considering spreading out tasks to various servers, ensuring that they are performed in the most efficient manner possible. Most importantly, a client-server network is much easier to secure and back up, greatly improving the reliability and confidentiality of the data [Rus03].

The three-tier design has many advantages over traditional two-tier design, the chief ones being [Van05]:

- The added modularity makes it easier to modify or replace one tier without affecting the other tiers.
- Separating the application functions from the database functions makes it easier to implement load balancing.

3. Peer-to-peer Architecture:

Peer-to-peer network, also called a workgroup, computers directly communicate with each other and do not require a server to manage network resources. A peer-to-peer network is most appropriate when fewer than ten computers are located in the same general area. The computers in a workgroup are considered peers because they are all equal and share resources among each other. Each user decides which data on his or her computer will be shared with the network. By sharing common resources, users can print from a single printer, access information in shared folders, and work on a single file without transferring it to a floppy disk. Every person can communicate with one or more other people, there is no fixed division into clients and servers as shown in figure 2.6 [Flo05, Tan03].

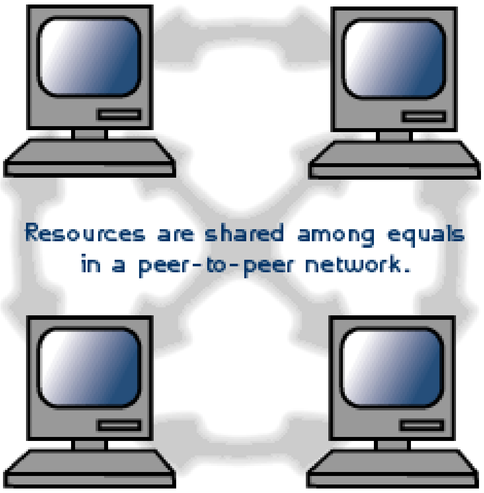


Figure 2.6 Peer-to-Peer Architecture

In a peer-to-peer network there are no dedicated servers or hierarchy among the computers. All of the computers on the network handle security and administration for themselves. The users must make the decisions about who gets access to what. Beyond that there are more similarities between the types of network than differences. All of the computers must have network cards. Also the same cables, the same hubs and switches, and the same protocols are used with a client-server model. The only difference is that there isn't a server.

Security on a peer-to-peer network is not very powerful. In a peer-to-peer network, the users handle administration. This means that all the users need to be trained in how to share files, folders, and printers. In peer-to-peer network, each computer that attaches to another computer, whether for printing or for file sharing, takes up system resources on the hosting computer. So a peer-to-peer network is sometimes the perfect (and cheap) solution for connecting the computers at a small nonprofit, so the advantages of peer-to-peer network is[Pet05, Tom03]:

- Easy to configure.
- No requirement for server hardware/software.
- Users can mange their own resources.
- No need for a network administrator.
- Reduce total cost.

The disadvantages of peer-to-peer network are[Pet05]:

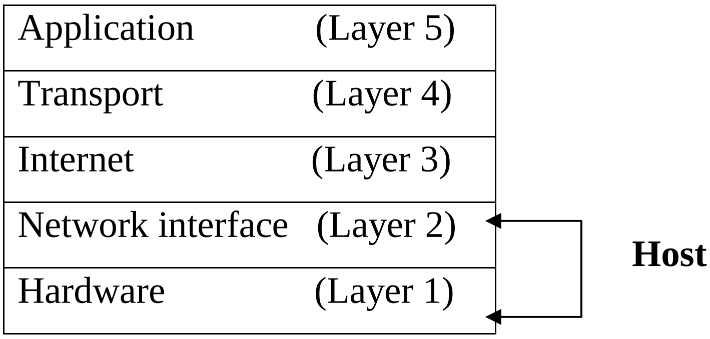
- Provide a limited number of connections.
- May slow performance of nodes.
- Do not allow central management.
- Do not have a central store of files.
- Users responsible for managing own resources.
- Offers very poor security.

2.3 Network Models

A network model (also referred to as *protocol suits*) reflects the design or architecture to accomplish communication between different systems. A network model usually consists of layers. Each layer of a model represents specific functionality.One of the most popular Network models is TCP/IP[Hel00].

The standard model of a layered network is the 7-layer International Standards Organization (ISO) Open Systems Interconnection (OSI) Reference model. The entire OSI model is not implemented, where the most common layered set of protocols in use is the Transmission Control Protocol/Internet Protocol (TCP/IP) set of protocols. TCP/IP works in a vary similar manner to the OSI model in that it takes a layered approach to provide network services. Each layer in the TCP/IP model communicates with the layers above and below it in the same way that the layers in the OSI model do[Hel00]. (For more information about OSI see Appendix B)

The TCP/IP network model takes its name from two of its protocols, the Transmission Control Protocol (TCP) and the Internet Protocol (IP). Figure 2.3 shows the five-layer represent of the TCP/IP model [Tan03].



Application	(Layer 5)
Transport	(Layer 4)
Internet	(Layer 3)
Network interface	(Layer 2)
Hardware	(Layer 1)

Figure 2.7 the TCP/IP Model

Each of the top three layers of the TCP/IP model actually consists of multiple protocols (as shown in figure 2.4). The following are the most popular protocols of ***Internet, Transport*** and ***Application*** Layers are illustrated in figure 2.4[Tan03]:

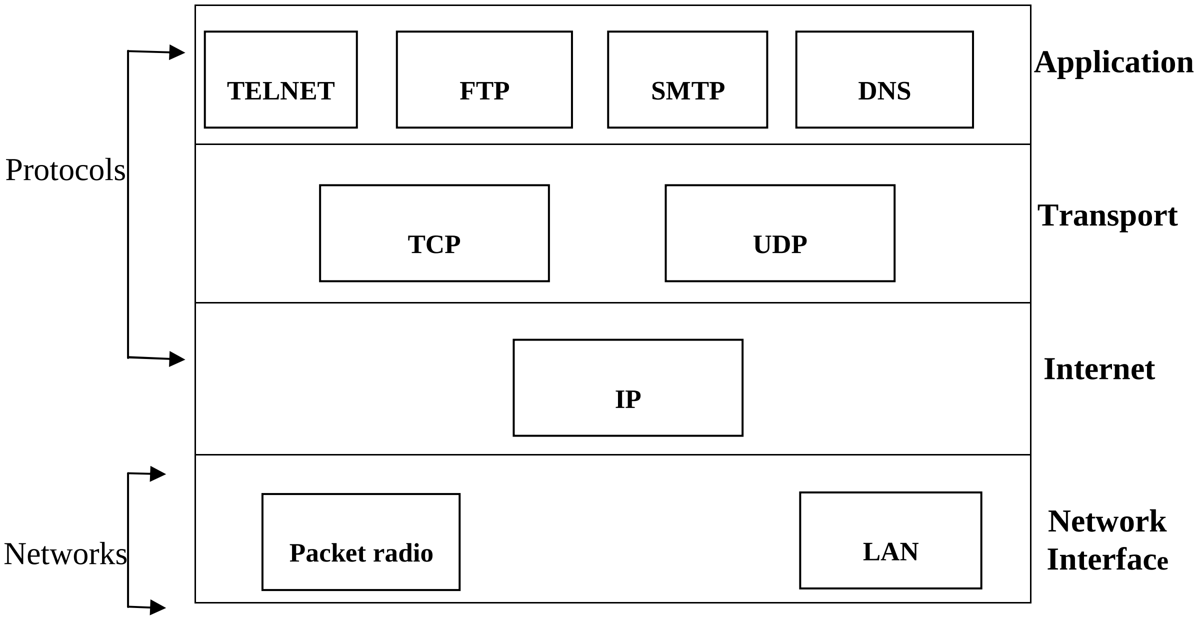


Figure 2.8 Protocols and networks in the TCP/IP model

2.3.1 IP (Internet Protocol)[Tan03]

Internet Protocol resides into Internet layer. Its main tasks are addressing of information datagrams (Packets) between computers and managing the fragmentation process of these datagrams. The protocol has a formal definition of the layout of a datagram of information and the formation of a header composed of information about the datagram.

IP is responsible for the ***routing of a datagram***, ***determining where it will be sent*** and ***devising alternate routes in case of problems***.

Another important aspect of IP's purpose has to do with unreliable delivery of a datagram. Unreliable in the IP sense means that the delivery of the datagram is not guaranteed ،because it can get *delayed, misrouted,* or *mangled in the breakdown*. IP has nothing to do with flow control or reliability: there is no inherent capability to verify that a sent message is correctly received. IP does not have a checksum for the data contents of a datagram, only for the header information.

Part of the IP system defines how gateways manage datagrams, how and when they should produce error messages, and how to recover from problems that might arise.

2.3.2 Transport layer

At the ***transport layer***, the two common protocols are the Transmission Control Protocol (TCP) and the User Data-gram Protocol (UDP):

1-TCP (Transmission Control Protocol)[Tan03]: The transmission control protocol provides a considerable number of services to the IP layer and the upper layers. Most importantly, it provides a connection-oriented protocol to the upper layers that enable an application to be sure that a datagram sent out over the network was received in its entirety. In this role, TCP acts as a *message-validation protocol providing reliable communications*. If a datagram is corrupted or lost, TCP usually handles the retransmission rather than the applications in the higher layers.

TCP manages the flow of datagrams from the higher layers to the IP layer, as well as incoming datagrams from the IP layer up to the higher-level protocols. TCP has to ensure that priorities and security are properly respected.

The isolation of all these services in a separate layer enables applications to be designed without regard to flow control or message reliability. Without the TCP layer, each application would have to implement the services themselves, which is a waste of resources. Because TCP is a connection-oriented protocol responsible for ensuring the transfer of a datagram from the source to destination machine (end-to-end communications), TCP must receive communication messages from the destination machine to acknowledge receipt of the datagram.

2- **UDP (User Datagram Protocol)[Hel00]:** UDP is based upon the datagram method of transport for application for which an occasional lost packet is not considerd serious. Thus, UDP represents a connectionless, unreliable, best-effort transport service.

UDP does not issue acknowledgments to the originator upon receipt of data nor provide order to incoming datagrams. UDP does not provide error detection or capabilities to recover from the situation where packets become lost. Instead, it is up to the application to detect lost or missing data, typically by noting the absence of a response within a predefined period of time and then if appropriate retransmitting the data that was presumed to be lost.

2.3.3 Application Layer

TCP/IP model provides a set of protocols at application layer. For example, Domain Name System (DNS), Simple Mail Transfer Protocol (SMTP), File transfer Protocol (FTP), and others.

2.4 Inter-process Communication IPC

Inter-Process Communication (IPC) is a set of techniques for the exchange of data between two or more threads in one or more processes. Processes may be running on one computer or on two or more computers connected by a network as shown in figure 2.7. The method of IPC used may vary based on the bandwidth and latency of communication between the processes, and the type of data being communicated [Car06].

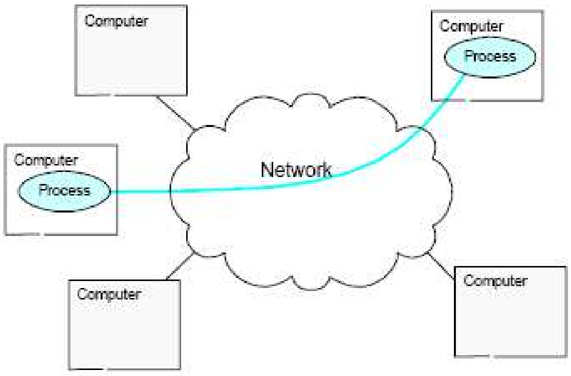


Figure 2.9 IPC through network connection

The IPC mechanisms can be classified into the following categories [Hir04]:

1. **Pipes**: unidirectional flow of communication between processes within the same system. In other words, they are half-duplex that is, data flows in only one direction.
2. **First In, First Out (FIFO)**: are similar to the working of pipes. FIFOs also provide half-duplex flow of data just like pipes. The difference between FIFOs and pipes is that the former is identified in the file system with a name, while the latter is not. That is, FIFOs are named pipes. FIFOs are identified by an access point which is a file within the file system, whereas pipes are identified by an access point. Another major difference between FIFOs and pipes is that FIFOs last throughout the life-

cycle of the system, while pipes last only during the life-cycle of the process in which they were created. FIFOs exist beyond the life of the process. Since they are identified by the file system, they remain in the hierarchy until explicitly removed using disconnect, but pipes are inherited only by related processes, that is, processes which are descendants of a single process.

3. **Shared memory**: refers to memory that is accessible by more than one process, if these processes shared the virtual address space; hence, any process sharing the memory region can read or write to it. Shared memory is used to facilitate inter-process communication.
4. **Sockets:** Sockets provide point-to-point, two-way communication between two processes. Sockets are very versatile and are a basic component of inter-process and inter-system communication. A socket is an endpoint of communication to which a name can be bound. It has a type and one or more associated processes. There are five types of socket [Dav99]:
   - **Stream socket:** provides two-way, sequenced, reliable, and unduplicated flow of data with no record boundaries. A stream operates much like a telephone conversation. The socket type is `SOCK_STREAM`, which, in the Internet domain, uses Transmission Control Protocol (TCP).
   - **A datagram socket:** supports a two-way flow of messages. A socket on a datagram socket may receive messages in a different order from the sequence in which the messages were sent. Record boundaries in the data are preserved. Datagram sockets operate much like passing letters back and forth in the mail. The socket type is `SOCK_DGRAM`, which, in the Internet domain, uses User Datagram Protocol (UDP).

- **A sequential packet socket:** provides a two-way, sequenced, reliable, connection, for datagram of a fixed maximum length. The socket type is SOCK_SEQPACKET. No protocol for this type has been implemented for any protocol family.
- **A raw socket:** provides access to the underlying communication protocols.

These sockets are usually datagram oriented, but their exact characteristics depend on the interface provided by the protocol.

In this thesis TCP/IP Protocol is used for make a connection between computers and stream socket as inter-process communication.

2.5 Network monitoring

A network monitoring is primarily concerned with observing and analyzing the status and behavior of the end systems, intermediate systems, and sub networks that create the configuration that is to be monitored and managed.

WAN monitoring is more complicated than LAN monitoring. WANs differ from LANs in several important ways. Like the internet, most WANs are not owned by any one organization but rather exist under collective or distributed ownership and management [Nat04].

Network monitoring encompasses the four functional areas: Fault, Accounting, Security, and Performance Monitoring.

1. Fault monitoring

The objective of fault monitoring is to identify faults as quickly as possible after they occur and to identify the cause of the fault so that remedial action may be taken. In a complex environment, locating and diagnosing faults can be difficult. There are specific problems associated with fault observation as indicated by the following [Cis06]:

- Unobservable faults: Certain faults are inherently unobservable locally. For example, the existence of a deadlock between cooperating distributed processes may not be observable locally. Other faults may not be observable because vendor equipment is not instrumented to record the occurrence of a fault.
- Partially observable faults: A node failure may be observable but the observation may be insufficient to pinpoint the problem. A node may not be responding because of the failure of some low level protocol in an attached device.
- Uncertainty in observation: Even when detailed observations of faults are possible, there may be uncertainty and even inconsistencies associated with the observations. A lack of response time from a remote device may mean that the device is locked up, the network is partitioned, congestion caused the response to be delayed, or the local timer is faulty.

Once faults are observed, it is necessary to isolate the fault to a particular component. The following is a list of problems that can arise [Cis06]:

- Multiple potential causes: When multiple technologies are involved, the potential points of failure and the types of failure increase. This makes it harder to locate the source of a fault.
- Too many related observations: A single failure can affect many active communication paths. A failure in one layer of the communications architecture can cause degradations or failures in all dependent higher layers. A failure in a line will be detected in the routers as a link failure and in the workstations as transport and application failures. Because a single failure may generate many secondary failures, the proliferation of

fault monitoring data that can be generated in this manner can obscure the single underlying problem.

- Interference between diagnosis and local recovery procedures: Local recovery procedures may destroy important evidence concerning the nature of the fault, thereby disabling diagnosis.
- Absence of automated testing tools: Testing to isolate faults is difficult and costly to administer.

The first requirement of a fault monitoring system is that it must be able to detect and report faults. At a minimum, a fault monitoring agent will maintain a log of significant errors and events. These logs, or summaries, are available to authorized manager systems. A system that operates primarily by polling would rely on these logs. The fault monitoring agent has the capability to report errors in independently to one or more managers.

In addition to reporting known, existing faults, a good fault monitoring system will be able to anticipate faults. Generally, this involves setting up thresholds and issuing a report when a monitored variable crosses a threshold. For example, if the fraction of transmitted packets that suffers an error exceeds a certain value, this may indicate that a problem is developing along the communications path. If the threshold is set low enough, the network manager may be alerted in time to take action that can avoid a major failure in the system. The fault monitoring system should also assist in isolating and diagnosing the fault. Examples of tests that a fault monitoring system should have at its command include [Tan03]:

- Connectivity test.
- Data integrity test.
- Protocol integrity test.

- Data saturation test.
- Connection saturation test.
- Response time test.
- Loopback test.
- Function test.
- Diagnostic test.

An effective user interface is required more for fault monitoring than for other areas of network monitoring. In complex situations, faults will be isolated, diagnosed, and ultimately corrected only by the cooperative effort of human user and monitor software [Net04].

2. Account monitoring

Accounting monitoring is primarily a matter of tracking user usage of network resources. The requirements for this function vary widely. In some environments, accounting may be quite general. An internal accounting system may be used only to assess the overall usage of resources and to determine what proportion of the cost of each shared resource should be allotted to each department. In other cases, particularly for systems that offer a public service or systems with only internal users, it is required that usage be broken down by account, by project, or even by individual user for the purposes of billing. The information gathered by the monitor system in this case must be more detailed and more accurate than that required for a general system.

Examples of resources that may be subject to accounting include the following [Cis06]:

- Communication facilities: LANs, WANs, leased lines, dial up lines.
- Computer hardware: Workstations and servers.

- Software and systems: Applications and utility software in servers, a data center, and end user sites.

For any given type of resource, accounting data are collected based on the requirements of the organization. The following communications related accounting data might be gathered and maintained on each user [Cis06]:

- User identification: Provided by the originator of a transaction or a service request.
- Receiver: Identifies the network component to which a connection is made or attempted.
- Number of packets: Count of data transmitted.
- Security level: Identifies the transmission and processing priorities.
- Time stamps: Associated with each principal transmission and processing event such as transaction start and stop times.
- Network code status indicates the nature of any detected errors or malfunctions.
- Resources used: Indicates which resources are invoked by this transaction or service agent.

3. Security Monitoring

Security Monitoring is a method used to confirm that the security practices and controls in place are being adhered to and are effective. Monitoring consists of activities such as the review of [Joh04]:

- Automated intrusion detection system logs.
- Firewall logs.
- User account logs.
- Network scanning logs.

- Application logs.
- Data backup recovery logs.
- Help desk logs.
- Other log and error files.

The purpose of the Security Monitoring Policy is to ensure that information resource security controls are in place, are effective, and are not being bypassed. One of the benefits of security monitoring is the early identification of wrongdoing or new security vulnerabilities. This early identification can help to block the wrongdoing or vulnerability before harm can be done, or at least to minimize the potential impact. Other benefits include Audit Compliance, Service Level Monitoring, Performance Measuring, Limiting Liability, and Capacity Planning [Cis04].

4. Performance Monitoring

It allows the measuring and tracing of the progression of the actual usage and performance of various network features, e.g. the number of users logged onto a server, Input/Output monitoring, mouse & keyboard monitoring, and desktop monitoring depending on the implementation, the progression of the network activity might be stored into a data base to allow further statistical processing and extraction [Spy04].

An absolute prerequisite for the management of any communications network is the ability to measure the networks performance or performance management. A system or activity cannot be managed and controlled properly unless its performance can be monitored. The difficulty facing the network manager is the selection and use of appropriate indicators that can measure the networks performance in an efficient manner and give the necessary information needed to solve a fault. Among the many problems that may arise in selecting the proper performance indicators are the following [Cis06]:

- There are too many indicators in use.
- The meanings of most indicators are not yet clearly understood.
- Some indicators are introduced and supported by certain manufacturers only .
- Most indicators are not suitable for comparison with each other.
- Frequently, the indicators are accurately measured but incorrectly interpreted .
- In many cases, the calculations of indicators take too much time, and the final results can hardly be used for controlling the environment.

Performance monitoring encompasses three components: performance measurement, which is the actual gathering of statistics regarding network traffic and timing; performance analysis, which consists of software for reducing and presenting the data; and synthetic traffic generation, which permits the network to be observed under a controlled load environment.

Performance measurement is often accomplished through agent modules within the devices that are attached to the network such as hosts, routers, and bridges. These agents are in a position to observe the amount of traffic into and out of a node, the number of connections, and the traffic per connection, as well as other measures that provide a detailed picture of the behavior of that node. This measurement has the disadvantage of having to process resources within the node. Measurements of data packet size, packet type, packet inter-arrival time, channel acquisition delay, communication delay, and collision and transmission [Net05].

2.6 Visual Basic .NET:

Visual Basic .NET, the next generation of the Visual Basic language, is a fast and easy way to create .NET-based applications, including Web services and Web applications.

Visual Basic .NET has many new and improved features that make it a powerful object-oriented programming language, including inheritance, interfaces, and overloading. Other new language features include free threading and structured exception handling. Visual Basic .NET also fully integrates the .NET Framework and the Common Language Runtime, which provide language interoperability, garbage collection, enhanced security, and improved versioning support.

An important new language feature is garbage collection, which is administered by the Common Language Runtime and provides better memory management. The universal type system allows for greater interoperability, also contributing to the enhanced power and flexibility found in Visual Basic .NET [MSD03].

2.7 Choosing Communication Type in Visual Basic .NET

The .NET Framework provides several ways to communicate with objects in different application domains, each designed with a particular level of expertise and flexibility in mind. For example, the growth of the Internet has made Extensible Markup Language (XML) Web services an attractive method of communication, because XML Web services are built on the common infrastructure of the HyperText Transmission Protocol (HTTP).

These are public standards, and can be used immediately with current Web infrastructures without worrying about additional proxy or firewall issues.

Communication in Visual Basic .NET could be by Active Server Base (ASP) .NET or .NET remoting. Both ASP.NET and .NET remoting are inter-process communication implementations. ASP.NET provides an infrastructure, hosted by Internet Information Services (IIS), that handles basic types well and is familiar to Web application developers. .NET remoting is a generic and extensible inter-process communication system that you can use to create XML Web services hosted in IIS (and have all the security, scalability, and session and application state of ASP.NET and IIS), or applications that use any other type of communication protocol or serialization format [Jon06, MSD03].

2.8 Using .Net Remoting for Inter-Process Communication in Visual Basic .Net

Microsoft created remoting in .Net for communication across applications. .NET objects are exposed to remote processes, thus allowing IPC. The processes can be on the same computer or across the network.

Event though remoting is useful in certain situations, the focus has now shifted to using web services for exposing remote functionality.

Remoting consists of a server and a client. A remotable object is passed between the client and the server to expose the desired functionality.

.NET remoting provides the tools for any number of comprehensive communication scenarios that include, but are not limited to, XML Web services these tools are [Jon06]:

- Publish or consume services in any type of application domain, whether that domain is a console application, a Windows Form, Internet Information Services (IIS), an XML Web service, or a Windows Service .
- Preserve full managed code type system fidelity in binary formatted communications .

- Pass objects by reference and return to a particular object in a particular application domain .
- Control activation characteristics and object lifetimes directly .
- Implement and use third-party channels or protocols to extend communication to meet your specific needs .
- Participate directly in the communication process.

Chapter Three

Design of The Proposed System

3.1 Introduction

This chapter contains the implementation of a network monitoring system for a client\server LAN which performs some functions of the security and performance monitoring.

This project is implemented to provide information about user’s activities on remote PCs by monitoring the performance of keyboard, mouse, desktop events, and programs that are running by the users on the LAN, and provides some control on these PCs.

The proposed system, Multi-node Remote Monitoring System (MRMS), consists of two integrated components: The Administrator component and the agent component, the Administrator watches all the activities of the users and has some control on them like closing any running program, sending warning messages, log off the user computer or shutdown it. The agent component run silently in the user's computers and collects information about user activities and sends this information to the Administrator, as shown in figure 3.2 which represents the full architecture of MRMS.

MRMS is built using Microsoft Visual Basic .Net 2003. It is tested on a LAN with bus architecture in which all the computers are connected to the network through a HUB and running under windows XP operating system.

Microsoft Visual Basic .Net was chosen as a programming language because it supports network operations and at the same time has an API that enabled the programmer to get needed information from the host operating system.

3.2 MRMS Architecture

MRMS architecture depends on the concept of client/server system, at least two computers must be existed, one PC should works as a LAN Administrator and the others are the LAN users, the Administrator can monitor and control any PC (or no. of PCs) on the LAN as shown in figure 3.1.

The proposed system consists of two parts: the first part (Administrator subsystem) resides in Administrator computer and the second one (Agent subsystem) is resides in the remote monitored computers as shown in figure 3.2.

The Administrator subsystem creates Administrator Socket and the Agent subsystem creates Agent Socket, These socket establish the connection between Administrator and Agent subsystems and responsible of sending and receiving requests and data between the subsystems.

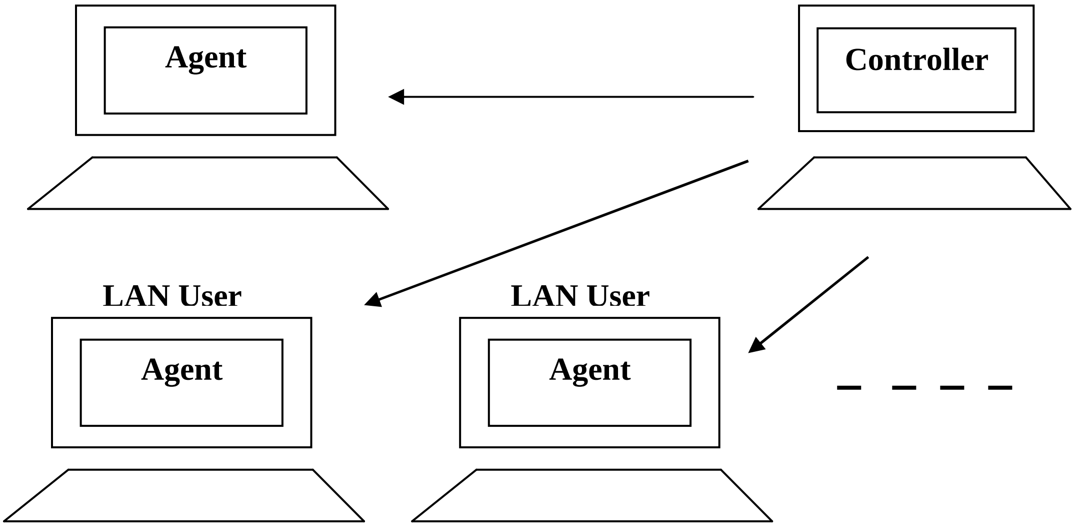


Figure 3.1 Remote Monitoring in a LAN

The main function of the Agent subsystem is to collect information about the keyboard, mouse, desktop screen, and programs that are running by the user, and send it to the LAN Administrator.

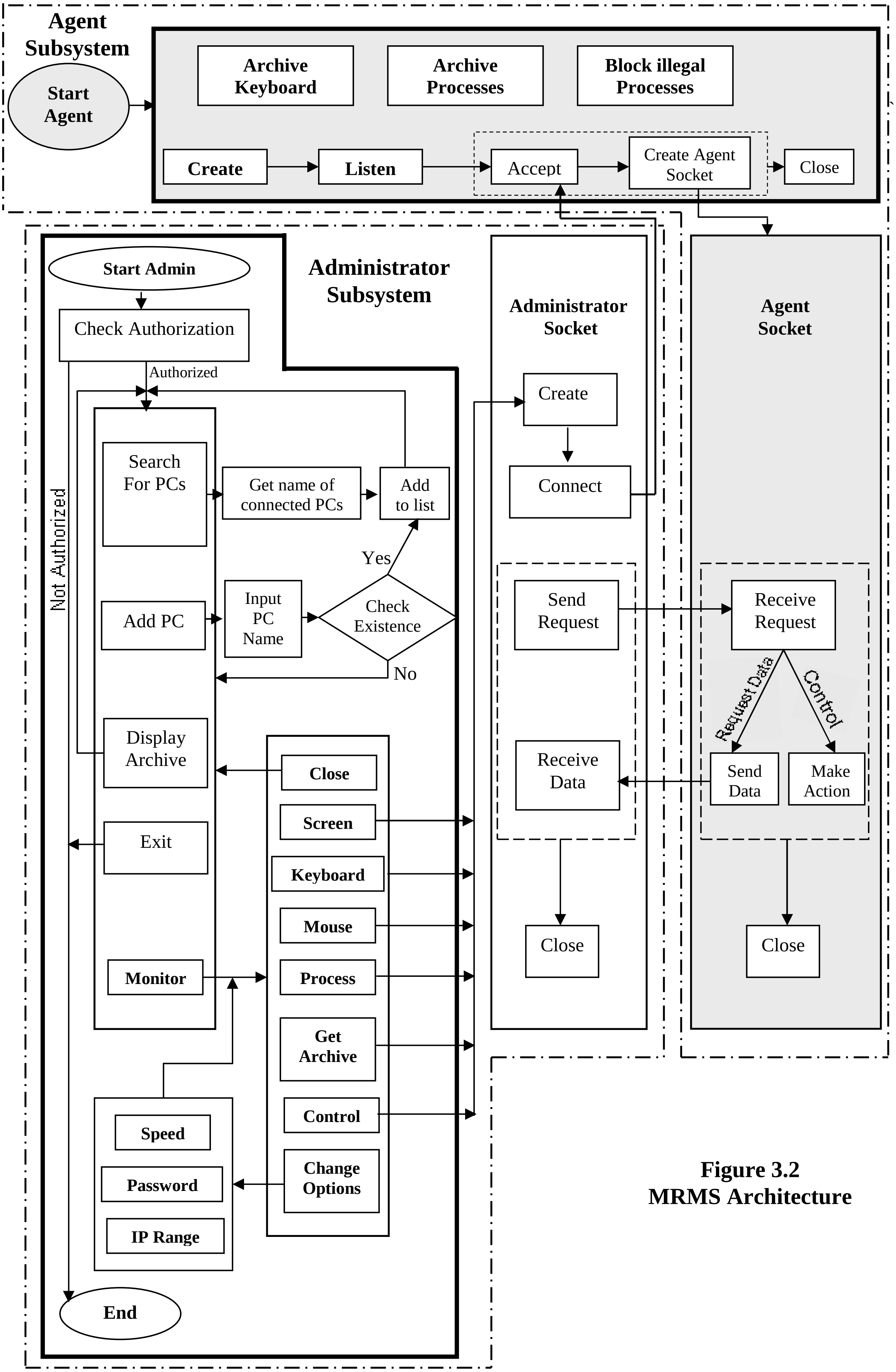


Figure 3.2
MRMS Architecture

The Agent subsystem in the remote computer has many functions as shown in figure 3.3.

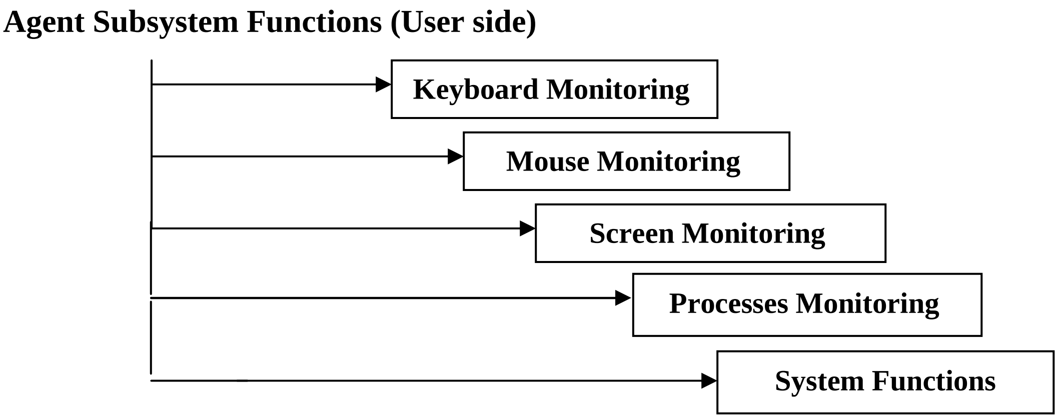


Figure 3.3 Agent subsystem functions on the remote Computer (User side)

The main function of the Administrator side (Watcher and Controller) is to receive information about the keyboard, mouse, desktop screen, and programs that are running by the users, and display this information to the Administrator. The Administrator has many activities and controls over the users PCs on the LAN as shown in figure (3.4).

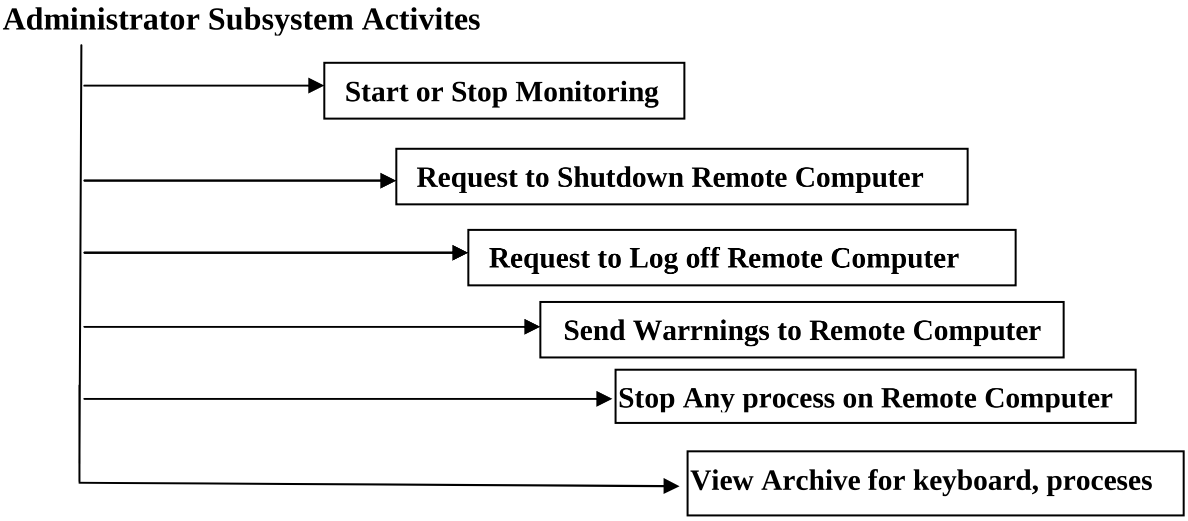

Administrator Subsystem Activites
Start or Stop Monitoring
Request to Shutdown Remote Computer
Request to Log off Remote Computer
Send Warrnings to Remote Computer
Stop Any process on Remote Computer
View Archive for keyboard, proceses

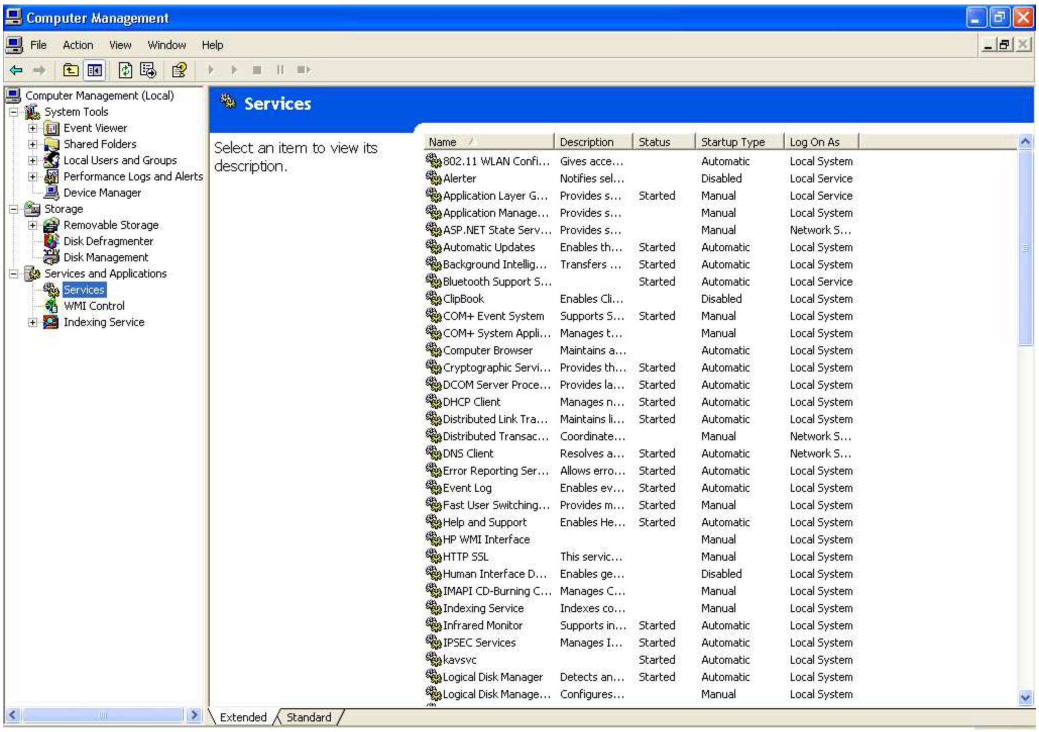
Figure 3.4 The Administrator Subsystem Activities

3.3 The Agent Subsystem Properties and Functions:

The Agent subsystem is built as a windows service application which enables the program to start at each windows startup, make it invisible in the task manager menu (Ctrl+Alt+Del), and did not interfere with other users who are working on the same computer.

Windows Service applications run in their own security context and are started before the user logs into the Windows computer on which they are installed.

A service goes through several internal states in its lifetime. First, the service is installed onto the system on which it will run. This process executes the installers for the service project and loads the service into the Services Control Manager for that computer. The Services Control Manager is the central utility provided by Windows to Administrator services as shown in fig 3.5.



Select an item to view its description.

Name	Description	Status	Startup Type	Log On As
802.11 WLAN Confi...	Gives acce...		Automatic	Local System
Alerter	Notifies sel...		Disabled	Local Service
Application Layer G...	Provides s...	Started	Manual	Local Service
Application Manage...	Provides s...		Manual	Local System
ASP.NET State Serv...	Provides s...		Manual	Network S...
Automatic Updates	Enables th...	Started	Automatic	Local System
Background Intellig...	Transfers ...	Started	Automatic	Local System
Bluetooth Support S...		Started	Automatic	Local Service
ClipBook	Enables Cli...		Disabled	Local System
COM+ Event System	Supports S...	Started	Manual	Local System
COM+ System Appli...	Manages t...		Manual	Local System
Computer Browser	Maintains a...		Automatic	Local System
Cryptographic Servi...	Provides th...	Started	Automatic	Local System
DCOM Server Proce...	Provides la...	Started	Automatic	Local System
DHCP Client	Manages n...	Started	Automatic	Local System
Distributed Link Tra...	Maintains li...	Started	Automatic	Local System
Distributed Transac...	Coordinate...		Manual	Network S...
DNS Client	Resolves a...	Started	Automatic	Network S...
Error Reporting Ser...	Allows erro...	Started	Automatic	Local System
Event Log	Enables ev...	Started	Automatic	Local System
Fast User Switching...	Provides m...	Started	Manual	Local System
Help and Support	Enables He...	Started	Automatic	Local System
HP WMI Interface			Manual	Local System
HTTP SSL	This servic...		Manual	Local System
Human Interface D...	Enables ge...		Disabled	Local System
IMAPI CD-Burning C...	Manages C...		Manual	Local System
Indexing Service	Indexes co...		Manual	Local System
Infrared Monitor	Supports in...	Started	Automatic	Local System
IPSEC Services	Manages I...	Started	Automatic	Local System
kavsvc		Started	Automatic	Local System
Logical Disk Manager	Detects an...	Started	Automatic	Local System
Logical Disk Manage...	Configures...		Manual	Local System

Figure 3.5 The Services Control Manager

After the service is installed in the computer it will be worked silently with each windows startup without any interface with the user and the user has no ability to delete it.

The Agent subsystem consists of six modules, the main modules are: keyboard module, mouse module, screen capture module, process monitoring module, log off module, and shutdown module.

1. **Keyboard Monitoring Module (**Keystroke Recorder):

Keystroke Recorder will record all keystrokes typed secretly, completely undetectable to the user; It not only captures standard alphanumeric keystrokes, it can also records "hidden" characters and keystroke combinations such as the Shift, Alt, Tab, Ctrl keys and many others, including functional keys such as F1 - F12, Print Screen, Num Lock and the rest, as well as "true" keystrokes which may otherwise appear differently on screen (such as passwords typed).

The Agent subsystem captures the keystroke at each time key pressed and saves the data on a hidden archive file. So when the Administrator requests for keyboard archive the Agent subsystem send this file to the Administrator. The Administrator has the ability to clear this archive file by sending a request to the Agent subsystem to clear the file.

Algorithm (3.1) represents the steps of capturing the information of the keyboard and store them on archive file.

Algorithm (3.1) Keyboard Monitoring

Method: this algorithm captures the keyboard pressed keys and save it in an archive file.

Input:

Pressed key.

Output:

Pressed keys on an archive file.

Algorithm steps:

<u>On Agent Program :</u>

While (key_pressed) Agent subsystem hook the keystroke and save it on a hidden text file.

<u>On Administrator Program :</u>

```
If Administrator requests keyboard Archive then
  Step1  : Administrator subsystem open connection with the specified computer .
  Step2  : Agent subsystem send data from keyboard archive file to the
           Administrator subsystem.
  Step3  : Displays the data on the Adminstrator's screen.
  Step4  : Disconnect the connection with Administrator subsystem.
```

2. Mouse Monitoring Module:

This module provides the Administrator with the motion of the mouse of the remote computer (the X, Y position) using the windows API (GetCursorPos). The module gets the position of the mouse on each time the desktop screen shot is transferred from the user to the Administrator since the mouse motion will appear on desktop screen image of the user on the Administrator screen. Algorithm (3.2) represents the steps of getting mouse position.

Algorithm (3.2) Mouse Monitoring

Method: this algorithm captures the Mouse Position and send it to the Administrator.

Input:

Request from Administrator to Agent subsystem to get mouse position.

Output:

Mouse position.

Algorithm steps:

Step 1 : Administrator subsystem open connection with the specified computer .

Step 2 : Agent subsystem get position of mouse (X,Y) and send data.

Step 3 : Sets the Mouse image of this Agent to the specified position.

Step 4 : Disconnect the Administrator subsystem connection.

3. Screen Monitoring Module:

This Module gets the screen image of the user and sends either the whole screen image or only the changed pixels to the Administrator. It takes a snapshot of the user’s screen and stores it in an array, this snapshot is taken using windows API (GetDesktopWindow). Agent subsystem sends the entire array to the Administrator when the request is coming for the first time, but in the next requests only the changed pixels are sent to the Administrator subsystem with their new positions. The Administrator subsystem receives the changed pixels and put them on the new positions on the Administrator screen. Algorithm (3.3) represents the steps of getting the screen image.

Algorithm (3.3) Screen Monitoring

Method: this algorithm captures the screen image and send it to the Administrator.

Input:

Request from Administrator to Agent subsystem to capture the screen.

Output:

Screen image.

Algorithm steps:

Step 1 : Administrator subsystem open connection with the specified computer .

Step 2 : Agent subsystem captures the desktop screen and save it as an image.

Step 3 : Agent subsystem compare the new screen image with the previous one.

Step 4 : Agent subsystem sends changed pixels and their positions only to the Adminstrator.

Step 5 : Administrator subsystem receives the changed pixels and their positions and use these information to update the displayed image.

4. Processes Monitoring Module:

This module gets the processes names from the Agent computer using the windows API (Getprocess) method and sends them to the Administrator The Administrator has the ability to stop any process running on the user’s computer using windows API (process.kill) method.

The following algorithms (3.4), (3.5) represent the steps of getting the running processes and stop any selected process.

Algorithm (3.4) Get Processes

Method: this algorithm gets the names of the running processes on the specified computer.

Input:

Request from Administrator to get processes.

Output:

List of processes.

Algorithm steps:

Step 1: Administrator subsystem open connection with the specified computer.

Step 2 : Administrator subsystem wait for a connection with the Agent subsystem of the specified computer.

Step 3 : Administrator subsystem sends a request to get names of the running processes

Algorithm (3.5) Kill Process

Method: this algorithm kills any selected process.

Input:

Request from Administrator to kill a process.

Output:

New list of processes.

Algorithm steps:

Step 1 : Administrator subsystem open connection with the specified computer.

Step 2 : Administrator subsystem wait for a connection with the Agent subsystem of the specified computer.

Step 3 : Administrator sends a request to kill a specified user process.

Step 4 : Agent subsystem killes the specified process.

Step 5 : Agent subsystem gets the new processes names and sends them to Administrator.

Step 6 : Administrator subsystem displays processes names on the screen.

Step 7 : Disconnect the connection with the Agent subsystem.

5. Block illegal Processes Module:

This module blocks any illegal process executed by the users by ending its work. The Administrator state list of illegal programs and send it to the Agent subsystem to save it on a database on Agent computer, so even if the Administrator subsystem is not running the blocking module still working on offline monitoring as shown in algorithm (3.6).

The Administrator can update on this database by adding or deleting programs from it.

Algorithm (3.6) Block Process

Method: this algorithm blocks any illegal process.

Input:
File of a list of illegal proceses.

Output:
List of processes.

Algorithm steps:

```
Step 1 : Agent subsystem checks if there is any illegal process.
Step 2 : If there is illegal process then kill this process.
Step 3 : if True then Goto Step1.
Step4  : Exit.
```

3.4 Administrator Subsystem Modules:

Administrator subsystem has control on the users’ PCs. It contains several modules, the main modules are:

1. Check for Administrator Authenticity:

This module checks Administrator Authenticity before displaying the main window of the monitoring program. Algorithm (3.7) is used to check the Administrator Authenticity.

Algorithm 3.7 Check Administrator Authenticity

```
Method: this algorithm checks Administrator authentication.
Input:
      Administator_password: the password of the adminstator.
      Administator_name: the name of the administrator.
Output:
      Either true or false.
Algorithm steps:
Step 1 : K=0     /* K represents number of tries when illegal user ID and/or
         Password entered*/
Step 2 : Input pass and ID.
Step 3 : Check the ID and password
           If (ID = Administator_name) and (Pass = Administator_password)
           then
               Return(true)
           Else
               Inc(K)
               If K<3 then goto step2
Step 4 : Return(false)
Step 5 : Exit
```

2. Search the network for computers :

This module searchs the network for the switched-on computers that are connected to the network and displays a list of their names. Administrator selects the computers to monitor and control from this list.

This module is achieved by sending a broadcast to all network computers and wait to receive their IP addresses and computer names as shown in algorithm (3.8).

Algorithm 3.8 Search for switched on computers on the network

Method: Returns the IP addresses and computers names of the switched on computers on the network.

Input:
IP address of adminstrator computer.

Output:
List of IP addresses and computers name of the switched on computers on the network.

Algorithm steps:
Step 1 : Send broadcast to all computers on network.
and ask for IP addresses and computers names.
Step 2 : Receive the IP address and computer name of each computer
and add it to the list
Step 3 : Display the list to Aadministrator.

3. Add computer name to the list of computers:

This module allows the Administrator to add a computer name to the list of computers on the network manually. The module checks for real existence of specific computer and adds its IP and computer name to the list as shown in algorithm (3.9).

Algorithm 3.9 Add computer name to the list of computers

Method: this algorithm adds a computer name to the list of the computers on the network manually by the Adminstrator.

Input:

IP address or computer name of target computer.

Output:

True and add computer name to list or false.

Algorithm steps:

Step 1 : Check the computer of this IP address, or computer name, is existed on the network and switched on.

Step 2 : If true **Then** add this IP address or computer name to the list.
Else return false.

4. Start monitoring the selected computer:

This module opens a connection between the Administrator and the selected computer. It sends a request to the Agent subsystem, which is already running on the selected computer, to start sending information to the Administrator about the mouse and the screen. The Agent subsystem uses algorithms (3.2) and (3.3) to collect and send these information.

When the Administrator starts monitoring a specific computer several subprograms are running to have control on this specific computer such as:

- **Show processes:** this subprogram build a list contains the programs that are running on the monitored computer (only user programs not system ones), by sending a request to the Agent subsystem to start algorithm (3.4).

- **Kill process:** this subprogram allows the Administrator to kill any program on the monitored computer.

- **Sending warning message**: This subprogram gives the Administrator the ability to display warning messages on the remote user computer. The Administrator sends a request to the Agent subsystem to display a warning message to the user.

- **Logging off the remote computer:** this subprogram logs off the remote user computer. The Administrator sends a request to the Agent subsystem to logs off the windows using the windows API function:

ExitWindowsEx (EWX_FORCE, 0)

- **Shutdown the remote computer:** this subprogram is used to shutdown the remote computer. The Administrator sends a request to the Agent subsystem to shutdown the computer using the windows API function:

ExitWindowsEx (EWX_SHUTDOWN, 0)

- **Pause / resume monitoring:** this subprogram is used if the Administrator wants to pause the monitoring of a specific computer. This function will stop all the operations of getting the screen and mouse motion until the Administrator sends resume command then the monitoring operations continues.

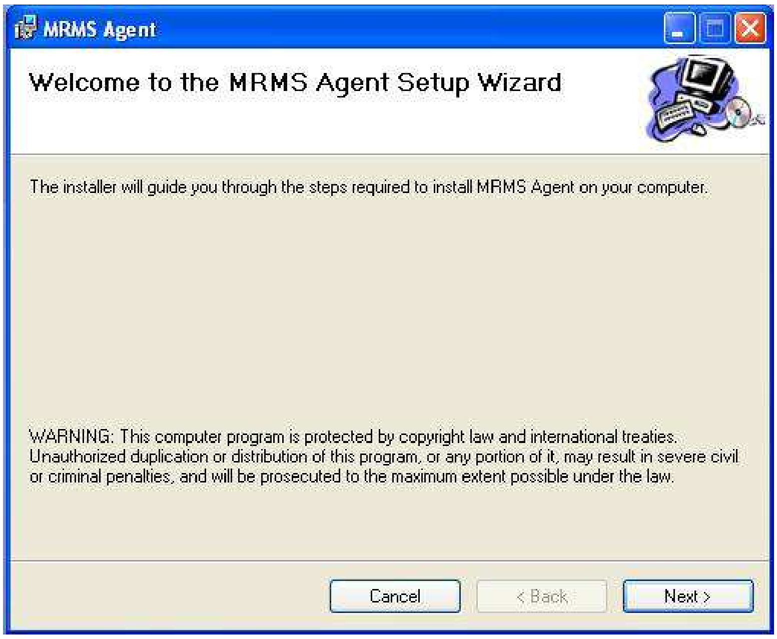
- **Stop monitoring a specific computer:** this subprogram stops all the monitoring operations and closes the connection with this computer.

Chapter Four

MRMS Implementation

4.1 Installing and Testing MRMS

MRMS system consists of two parts: Administrator part which installed on the Administrator's computer (watcher), and the Agent part which installed on the monitored computers as a windows service as shown in figure 4.1, 4.2, 4.3. (.Net) Framework must be installed on all the computers on the LAN so that no need to load Visual Basic on these computers, they works as a virtual machines.

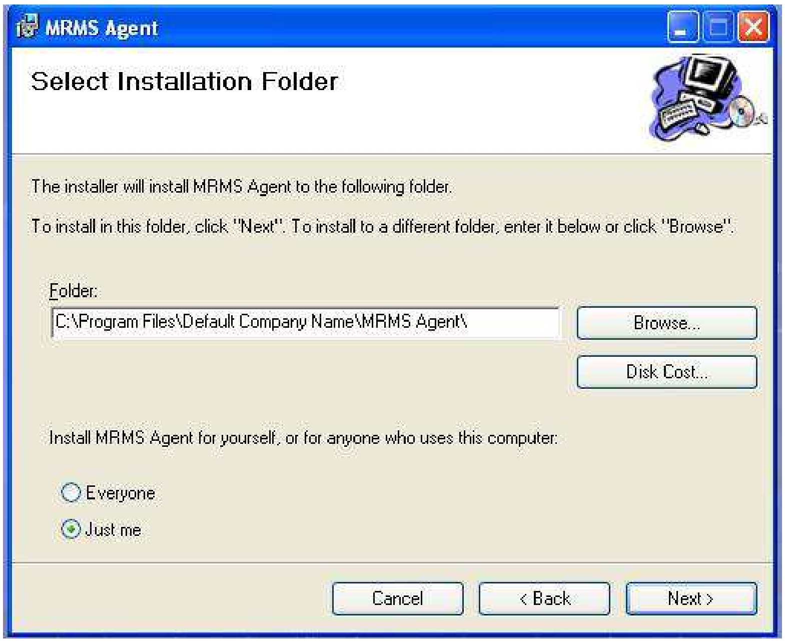


Welcome to the MRMS Agent Setup Wizard

The installer will guide you through the steps required to install MRMS Agent on your computer.

WARNING: This computer program is protected by copyright law and international treaties. Unauthorized duplication or distribution of this program, or any portion of it, may result in severe civil or criminal penalties, and will be prosecuted to the maximum extent possible under the law.

Figure (4.1) Installing Agent subsystem
Step1



Select Installation Folder

The installer will install MRMS Agent to the following folder.

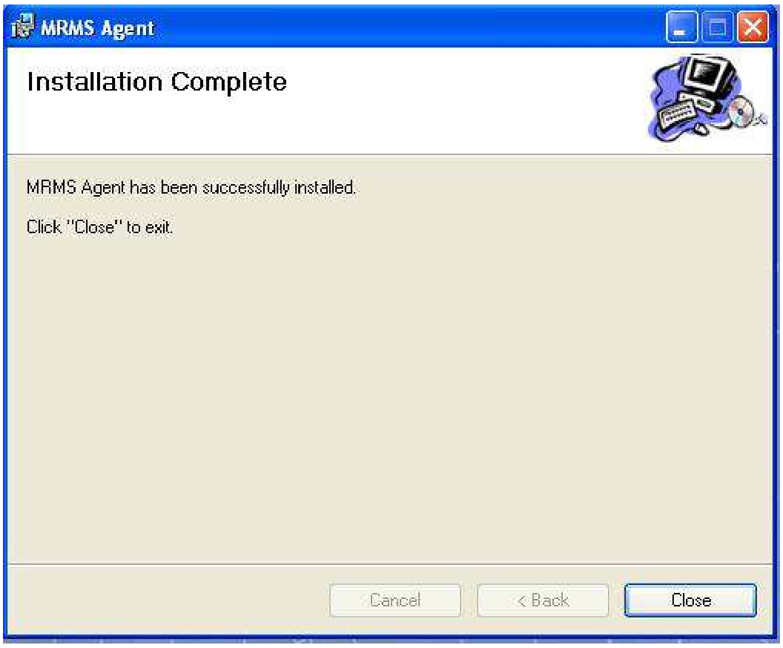
To install in this folder, click "Next". To install to a different folder, enter it below or click "Browse".

Folder: C:\Program Files\Default Company Name\MRMS Agent\

Install MRMS Agent for yourself, or for anyone who uses this computer:

○ Everyone
● Just me

Figure (4.2) Installing Agent subsystem
Step2



Installation Complete

MRMS Agent has been successfully installed.

Click "Close" to exit.

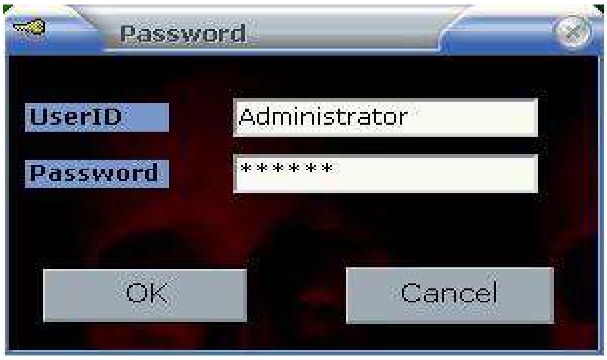
Figure (4.3) Installing Agent subsystem
Step3

The Administrator part is installed as an ordinary program on the Administrator computer. While the Agent part is installed as a windows service which has no interface with the user and invisible on task manager menu (Ctrl+Alt+Del) so that the user can not stop it or delete it.

The MRMS system designed with six main forms to enable the Administrator to monitor and control the user's computers. The forms are:

1. Password Form.
2. Main form.
3. Agent Form.
4. Options Form.
5. Monitoring Archive Form.
6. Keyboard and Programs Archive Form.

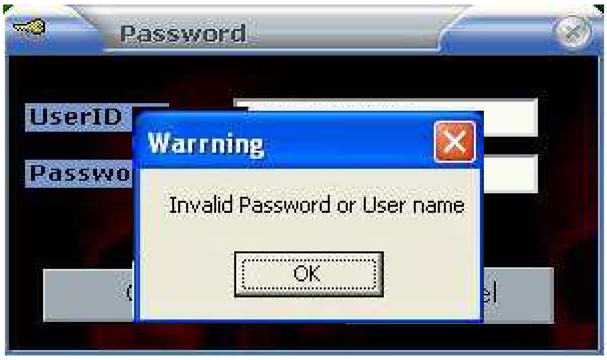
4.1.1 Password Form: First a password window is appearing to ensure the Authenticity of the Administrator as shown in figure 4.4, when the Administrator enters the correct user name and password the main window will be deployed on the screen. If Administrator enters incorrect user name or password a warning message is displayed as shown in figure (4.5). If Administrator enters wrong user name and password for three times the subsystem will closed. The Administrator has the ability to change the user name and password after entering to the main window using the options window.



UserID Administrator

Password ******

Figure (4.4) Password Form



Invalid Password or User name

Figure (4.5) Warning message after entering wrong password or user name

4.1.2 Main Form: This form consists of four options as shown in figure 4.6 these options are:

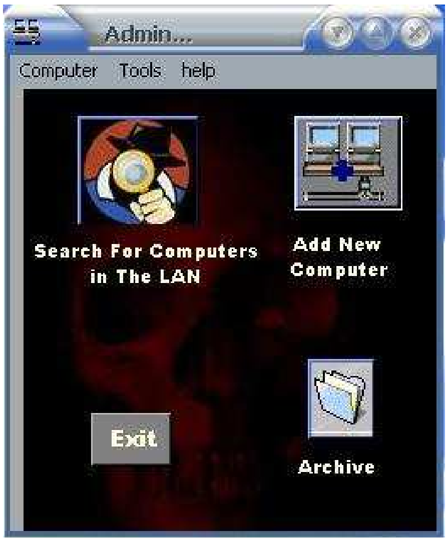
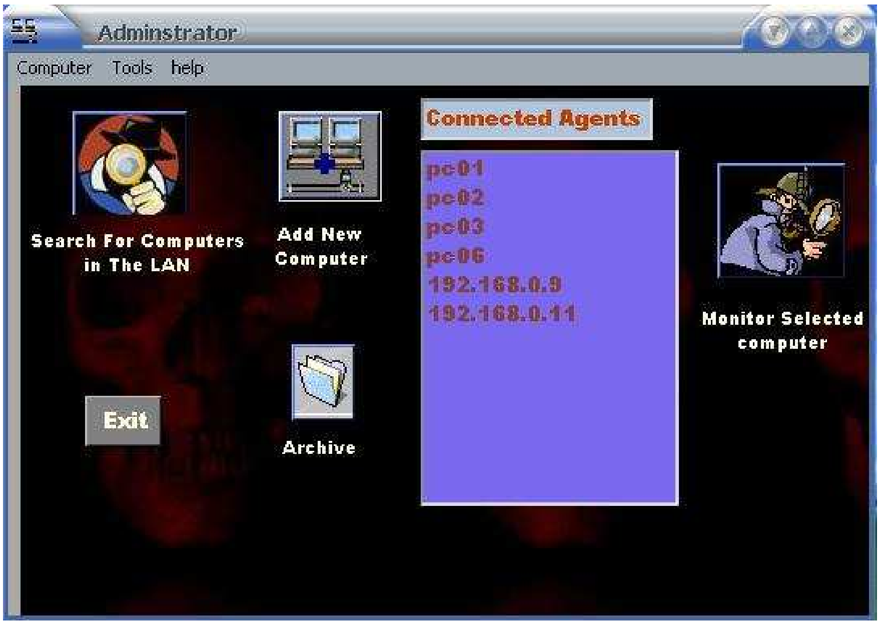


Figure (4.6) Main Form

a. **Search for computers in the LAN**: when choosing this option a list of all switched-on computers displayed on the screen. If the Agent program is existed on the target computers then their names will appear on the list otherwise, The IP of the computer is appeared. Also a new option is appeared on the screen (monitor selected computer) as shown in figure 4.7.

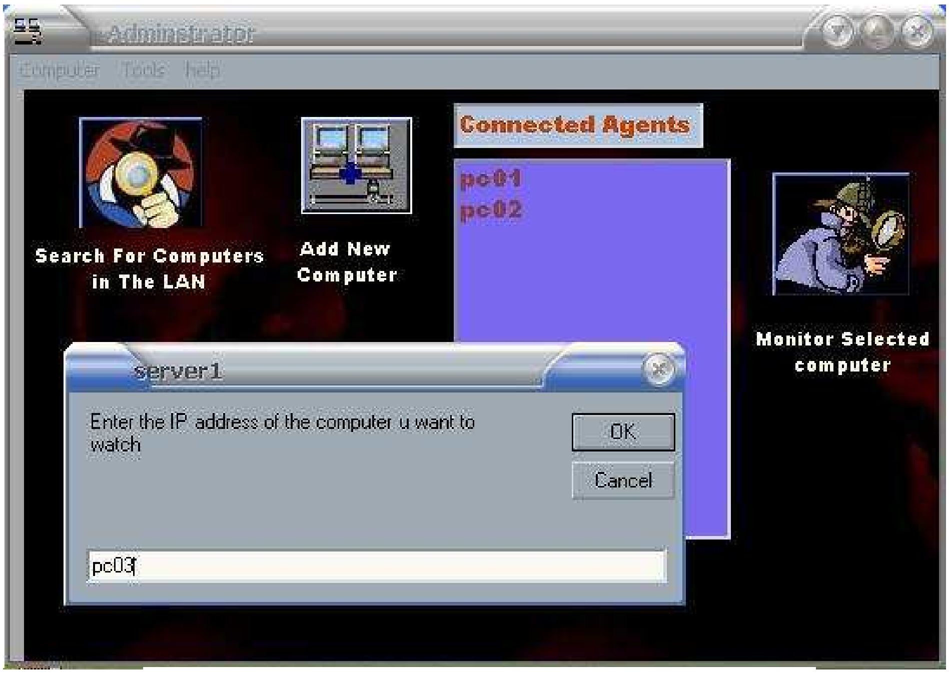


Connected Agents

pc01
pc02
pc03
pc06
192.168.0.9
192.168.0.11

Figure (4.7) Main Form with Agents list

b. **Add new Computer**: The Administrator may want to add a new computer IP or name which is not included in the list. This can be done using this option as shown in figure 4.8.



Connected Agents

pc01
pc02

Enter the IP address of the computer u want to watch

pc03

Figure (4.8) Add a new computer to Agents list

c. **Monitor selected computer:** When the Administrator select a computer name from the list and choose this option then the Agent form will deployed as shown in figure 4.9. This enables the Administrator to monitor and control the selected computer.

d. **Archive:** When choosing this option the archive form is deployed which contains the names of monitored computers and their monitored time as shown in figure 4.9.

e. **Exit:** Used to end the monitoring process (execution of the program).

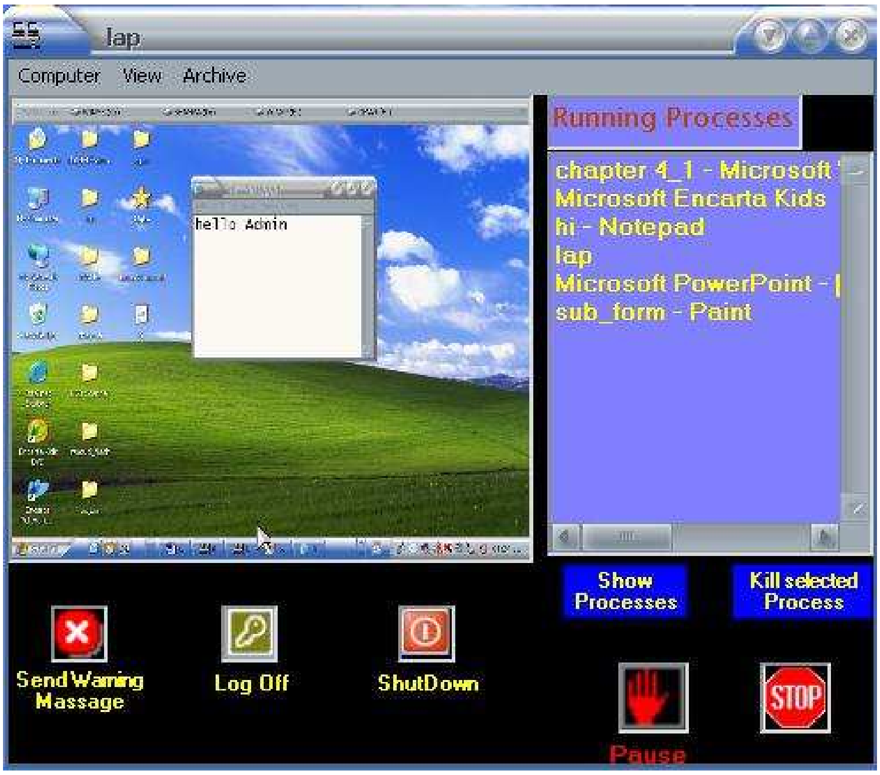
Also the main form contains a menu which contains:

a. **Options:** when it is selected the Options Form will appear as shown in figure 4.10.

b. **Help:** which contains two kinds of help

- Help about how to use the MRMS.
- Help about the product of the software.

4.1.3 Agent Form: This window will appear for each monitored Agent and contains the screen window of the target computer, a list of processes which are running in the target computer and a set of options as shown in figure 4.9.



hello Admin

chapter 4_1 - Microsoft
Microsoft Encarta Kids
hi - Notepad
lap
Microsoft PowerPoint - [
sub_form - Paint

Figure (4.9) Agent Form

The options are:

a. **Show Processes:** when pushing this button, all the processes which are running on the Agent computer will appear on the running processes list.

b. **Kill Process:** The Administrator has the ability to kill any process on the list by pushing on this button.

c. **Send Warning Massage:** To send any massage to the Agent.

d. **Log off:** to log off the Agent’s computer.

e. **Shutdown:** to shutdown the Agent’s computer.

f. **Pause / continue:** to pause and continue receiving information of this Agent.

g. **Stop:** to stop monitoring this Agent.

Also the Agent form contains a menu which contains:

a. **View:** to select the size of the displayed screen image of the Agent, two sizes are available, the small size which it is a part of the Agent window as shown in figure 4.9 and full window which full all the Agent window as shown in figure 4.10.

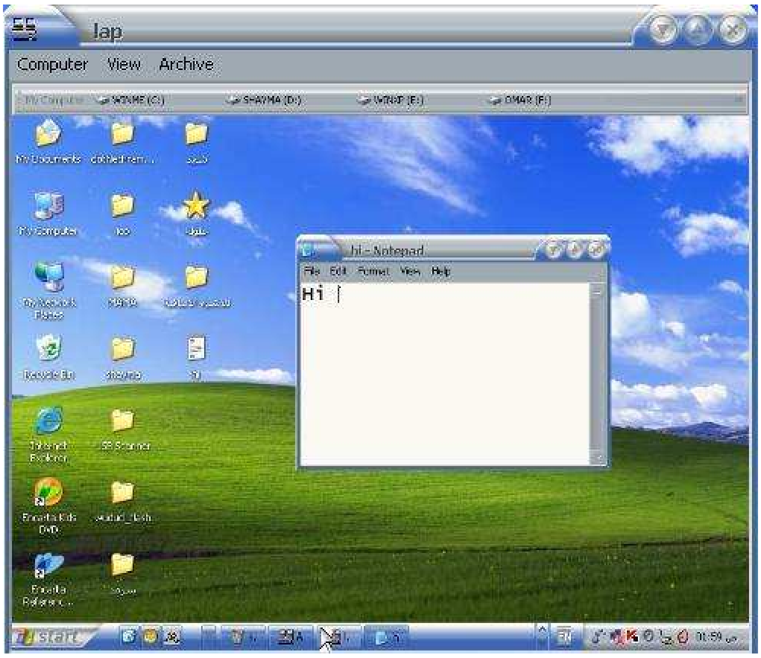
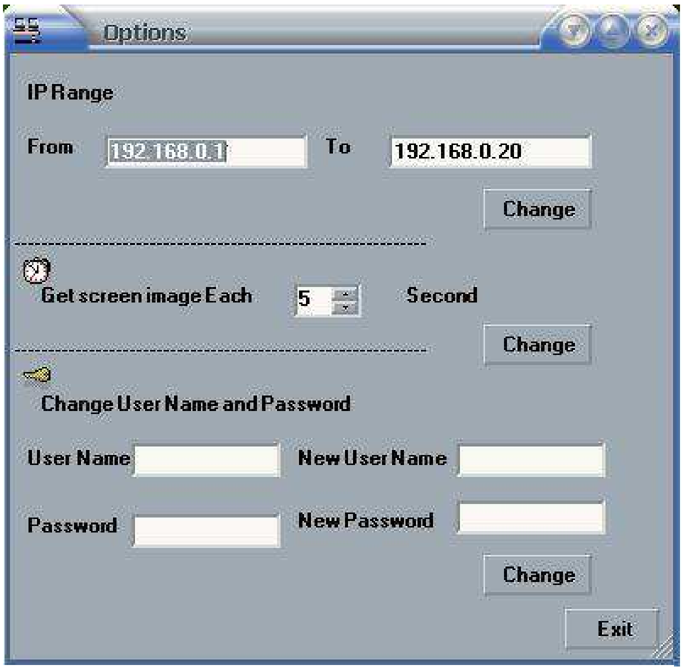


Figure (4.10) Agent Form in full window mode

b. **Archive:** contains four functions:

- **Keyboard Archive:** gets the keyboard archive file from the Agent’s computer as shown in figure 4.9.
- **Programs Archive:** gets the programs archive file from the Agent’s computer as shown in figure 4.10.
- **Clear Keyboard Archive:** Clears keyboard archive file which exists on the Agent’s computer.
- **Clear Programs Archive:** Clears Programs archive file which exists on the Agent’s computer.

4.1.4 Options Form: From this form the Administrator could changed some settings of the MRMS system as shown in figure 4.11 this settings are:



IP Range

From 192.168.0.1 To 192.168.0.20

Get screen image Each 5 Second

Change User Name and Password

User Name

New User Name

Password

New Password

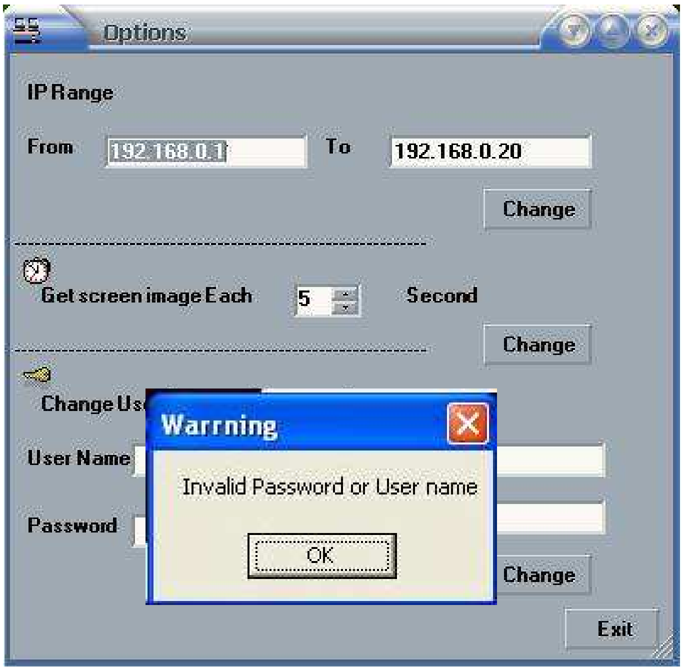
Figure (4.11) Options Form

a. **Change IP range:** this will change the range of IPs to search on for connected computers.

b. **Change the no. of received desktop images from the Agent:** this option will specify the no. of images to be received.

c. **Change the User name and Password:** this option allow the Administrator to change the user name and password after entering the correct old user name and password. If the Administrator enters incorrect user name or password a warning window will appear to him as shown in figure 4.12.

d. **Exit:** to return to main form.



IP Range

From 192.168.0.1 To 192.168.0.20

Get screen image Each 5 Second

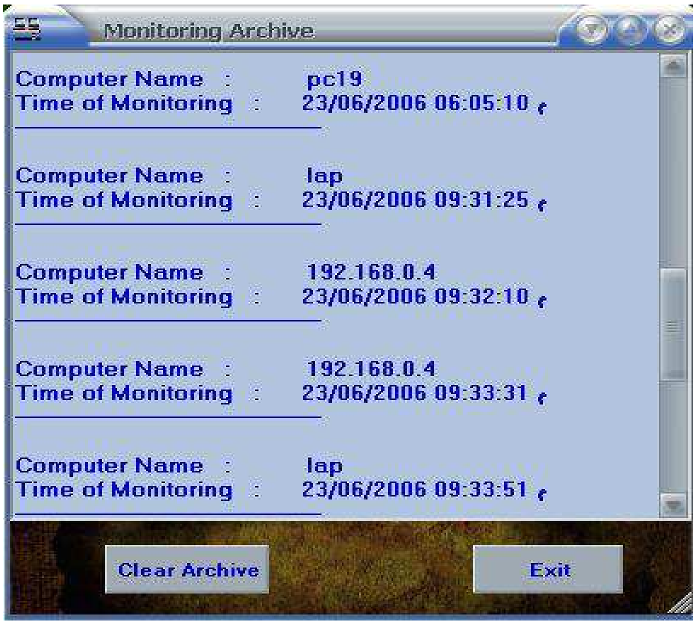
User Name

Password

Invalid Password or User name

Figure (4.12) Options Form

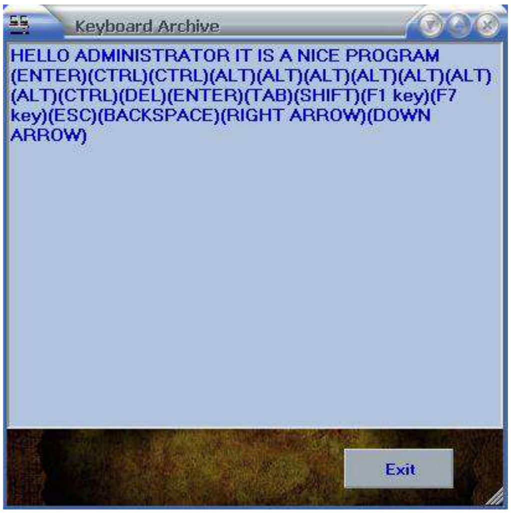
4.1.5 Monitoring Archive Form: This form is used to show the archive file which contains the monitored computers and monitoring time. The form contains also the Clear Archive button to clear the monitoring archive file from the Administrator computer, and exit button to return to main form, as shown in figure 4.13.



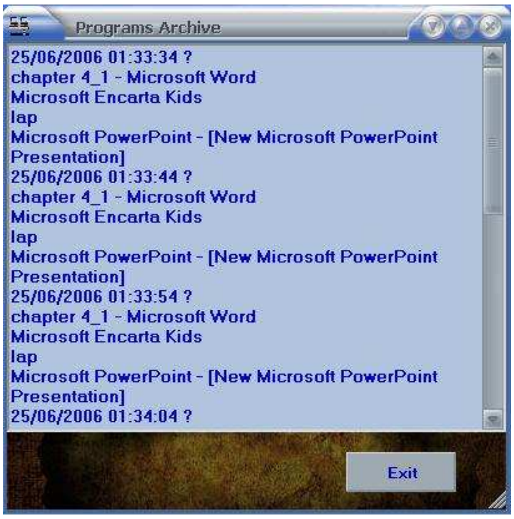
```
Computer Name      :     pc19
Time of Monitoring   :   23/06/2006 06:05:10 م

Computer Name      :     lap
Time of Monitoring   :   23/06/2006 09:31:25 م

Computer Name      :     192.168.0.4
Time of Monitoring   :   23/06/2006 09:32:10 م

Computer Name      :     192.168.0.4
Time of Monitoring   :   23/06/2006 09:33:31 م

Computer Name      :     lap
Time of Monitoring   :   23/06/2006 09:33:51 م
```

Figure (4.13) Monitoring Archive Form

4.1.6 Keyboard and Programs Archive Form: This form is used to display the keyboard archive file as shown in figure 4.14 and programs archive file as shown in figure 4.15.



```
HELLO ADMINISTRATOR IT IS A NICE PROGRAM
(ENTER)(CTRL)(CTRL)(ALT)(ALT)(ALT)(ALT)(ALT)(ALT)
(ALT)(CTRL)(DEL)(ENTER)(TAB)(SHIFT)(F1 key)(F7
key)(ESC)(BACKSPACE)(RIGHT ARROW)(DOWN
ARROW)
```



```
25/06/2006 01:33:34 ?
chapter 4_1 - Microsoft Word
Microsoft Encarta Kids
lap
Microsoft PowerPoint - [New Microsoft PowerPoint
Presentation]
25/06/2006 01:33:44 ?
chapter 4_1 - Microsoft Word
Microsoft Encarta Kids
lap
Microsoft PowerPoint - [New Microsoft PowerPoint
Presentation]
25/06/2006 01:33:54 ?
chapter 4_1 - Microsoft Word
Microsoft Encarta Kids
lap
Microsoft PowerPoint - [New Microsoft PowerPoint
Presentation]
25/06/2006 01:34:04 ?
```

Figure (4.14) Keyboard Archive Form

Figure (4.15) Programs Archive Form

4.2 Monitoring Example:

To monitor any computer on the LAN, these steps must be followed:

Step 1: The Adminstrator must enter the correct user name and pasword on the passowrd form as shown in figure 4.4.

Step 2: From the main form, when clicking on the search for computers on the LAN, a list of connected computers will appear as shown in figure 4.6.

Step 3: Or adding a computer name without searching by clicking on add new computer button as shown in figure 4.8.

Step 4: Adminstrator can select the computer name and click on monitor selected computer then the Agent form will appear as shown in figure 4.9. The administrartor can monitor as much as he wants from the computers on the list which the Agents' programs are installed on them.

Step 5: Adminstrator can display the processes that are executed on the Agent computer and stop any process by using **show processes** and **stop selected processes** buttons shown in figure 4.9.

Step 6: Adminstrator also can log off or shutdown the remote computer by using **log off** and **Shutdown** buttons shown in figure 4.9.

Step 7: Adminstrator also can send a warnning massage to the remote computer by using **Send warnning massage** button shown in figure 4.9.

Step 8: Adminstrator can display the keyboard and programs archive shown in figure 4.14 and 4.15, or clearing the archives from the main menu in the Agent form.

Step 9: Adminstrator can change some settings from the options on the tools menu like changing the IP range, changing the no. of received images from the Agent and changing the User name and Password as shown in figure (4.11).

Chapter Five

Conclusions and Suggestions Work

5.1 Conclusions

This work concerned with the implementation of a Multi-node Remote Monitoring system, which enables network Administrator to watch multiple computers connected through a LAN and has some control on them. Several problems are raised during the system development, The main problems are:

- How to get information from the remote user computer, because it is not possible to get information from remote computer unless there is an Agent subsystem installed on it. This Agent subsystem which is responsible of providing the Administrator with the needed information through the network connection.
- The big size of the desktop screen image takes time to be sent through connection, and sometimes only little change occur to the image. Therefore, only the changed pixels and their positions are sent through the connection.
- The mouse cursor does not appear when capturing the desktop image. Therefore the mouse position must be read and sent to Administrator to redraw it on the desktop image that appears on the Administrator screen, but reading and sending the mouse position is faster than reading and sending the desktop image (due to desktop image size). The solution to this problem is to read and send mouse position after sending the desktop image.

5.2 Suggestions Work

There is several ideas and suggestions for developing the MRMS system such as:

- **Full Monitoring and Controlling system:**

Add some other controlling tasks on the monitoring system (such as controlling the mouse and keyboard events). The Administrator can control both of them to change the user action, or prevent the user from some unallowable action.

- **Migrating the Agent subsystem into the user side computer**

To migrate the Agent subsystem into the user side computer, many methods could be used to do that. One of these methods is using another program and embedding the Agent subsystem inside it (Trojan horse). When the user run the program (Trojan horse), the user side program (Agent subsystem) will installed on the user computer and start running. Another way to migrate the Agent subsystem is through email by embedding the Agent subsystem on the email so the program will installed when the user open the email.

- **Warning messages from Agent subsystem to the Administrator**

The Agent subsystem sends warning messages to the Administrator if some error or illegal operations happened on the remote computer like running specified program which is not allowed.

- **Embading .Net framework:**

To avoid installing the .Net framework in each computer that will host the Agent subsystem, the .Net framework could be embedded inside the Agent subsystem.

- **Multimedia Monitoring:**

Monitoring all the multimedia devices, like sound system and video system. The video preview is not appearing when taking a snapshot to the desktop screen, therefore, the video image must be captured from the multimedia devices.

List of Abbreviations

API	Application Programming Interface.
ASP	Active Server Base Protocol.
DBMS	DataBase Management System.
FIFO	First In First Out
GDI	Graphics Devise Interface.
HTTP	Hyper Text Transmission Protocol
ICS	Internet Connection Sharing.
IP	Internet Protocol.
IPC	Inter-Process Communication.
ISO	International Standard Organization.
LAN	Local Area Network.
MAN	Metropolitan Area Network.
MRMS	Multi-node Remote Monitoring System.
OSI	Open System Interconnection.
PAN	Personal Area Network.
PC	Personal Computer.
PPP	Point - to - Point Protocol.
RAT	Remote Access Trojan.
SMTP	Simple Mail Transfer Protocol.
TCP	Transmission Control Protocol.
TCP/IP	Transmission Control Protocol / Internet Protocol.
UDP	User Datagram Protocol.
WAN	Wide Area Network.
XML	Extensible Markup Language.

References

- [**Abr85**] Abraham Silberschatz, James L. Peterson, "**Operating System Concepts**", 2nd edition, 1985.
- **[Ala02]** Alasadi H. Mohamed, Baghdad University, Computer science Dept., "**Design & Implement a remote PC performance monitoring",** M.Sc. thesis 2002.
- **[Alt02]** Altaie M. Saad, Al-Nahrain University, Computer Science Dept., "**Development of Windows Malicious Code for Remote Computers**", M.Sc. thesis 2002.
- **[Alz04]** Alzaidi D. Naeem, Al-Nahrain University, Computer Science Dept., **"Design & Implment Target Monitoring System**", M.Sc. thesis 2004.
- **[Ana02]** Anand Tripathi, Tanvir Ahmed Sumbed Pathak, and Megan Carneym "**Design of a Dynamiclly Extensible System for Network Monitoring using Mobile Agents**", 2002

  http://www.cs.umn.edu/Ajanta/papers/network-monitoring.pdf
- [**Nat04**] National Education Network Design, Inc., "**Network Management**", 2004.

  http://www.becta.org.uk
- [**Ben04**] Benjamin A. Kuperman," **A Categorization of Computer Security Monitoring Systems & The Impact on The design of Audit Sources** ", 2004.

  http://kb.indiana.edu/data/aehm.html
- [**Bra04**] Bradley Mitchell, "**LAN - Local Area Network**", 2004.

  http://www.compnetworking.about.com/library/glossary/bldef-lan.htm
- **[Cha01]** Chad Todd, "**Hack Proofing Windows 2000 Server**", Syngress, Inc., First Edition, 2001.

- [**CIS04**] CIS Issgy, "**Information Resource Security Policies**", 2004
  http://www.1_160_SecurityMonitoringPolicy.doc
- [**Cis06**] Cisco Systems,Inc , "**Network Management System**", 2006
  http://www.cisco.com/en/US/tech/tk869/tk769/technologies_white_paper 09186a00800aea9c.shtml
- **[Dav99]** Dave Marshall, "**IPC:Sockets**", 1999.
  http://www.cs.cf.ac.uk/Dave/C/node28.html
- **[Der03]** Deri L., Carbone R. and others, "**Monitoring Networks Using Ntop**"
  http://www.cs.plu.edu/courses/netsec/arts/im2001.pdf
- [**Enc05**] Encarta Reference Library, Inc., **"Network system**", 2005.
- [**Flo05**] Florida Center for Instructional Technology, "**Network Operating System**",2005
  http://www.fcit.usf.edu/network/chap6/chap6.htm
- [**Gor01**] Gorry Fairhurst, "**Metropolitan Area Network**", 2001.
  http://www.erg.abdn.ac.uk/users/gorry/eg3561/intro-pages/man.html
- [**Hel00**] P. T. Helton, "**Security in Computing**", Prentic Hall PTR, 2000.
- [**Hir04**] Hiran Ramankutty, "**Inter-Process Communication**", 2004
  http://linuxgazette.net/104/ramankutty.html
- [**Hop05**] Hope Computer ™, "**WAN**", 2005.
  http://www.computerhope.com/jargon/l/wan.htm
- **[INF02]** INFOSEC group**,** Computer Technology Associates, Inc., **"Computer Technology Associates (CTA)",** Paper 2002.
- **[Joh04]** John Wiley, R. T. Citron & Sons, Ltd.**,** "**Information Security",** 2004.
  http://www.it.uu.se/products/reports/2002-nc.pdf

- **[Jon06]** Jon Wojtowicz**,** "**Using .Net Remoting for Inter-Process Communication The Basics",** 2006.
  http://www.eggheadcafe.com/articles/20050831.asp
- [**Kel02**] Kelvinsky T.J, "**Hack I. T. Security Through Penetration Testing**", Person Education, 2002.
- [**Lam02**] Lam J. Kele, "**The Internet Challenges**", 2002.
  http://ihome.ust.hk/~lblkt/diploma/readings.html
- [**Law03**] Lawrence Harte, "**Introduction To Data Networks**", 2003.
  http://rapidshare.de/files/6373825/ALTHOS_Introduction_to_Data_Networks.rar
- **[MSD03]** MSDN library, Microsoft, Inc.,"**IPC in Visual Basic .Net**", 2003.
  ms-help://MS.MSDNQTR.2003APR.1033/cpguide/html
- **[Net04]** Netvizor Network Monitoring Software**, "Network monitoring spy software",** 2004**.**
  http://www. Network-monitoring.com/benefits.htm
- [**Net05**] NetScout, "**Network Performance Monitoring Systems**", 2005.
  http://www.netscout.com/solutions/it_initiatives.asp.
- [**Net06**] Network LookOut, "**Network LookOut Administrator**",2006.
  http://www.NetworkLookOut.com
- [**Nik04**] Nikolay k. Diakov, **"Monitoring distributed object and component communication"**, 2004.
- **[Off04]** Offical Spy Software Website, **"Computer Spy Software",**2004**.**
  http://www.SpySoftware.com
- [**Pet05**] Peter Burden, "**Selecting a Network Architecture**", 2005.
  http://www.scit.wlv.ac.uk/~jphb/cp2073/Lecture3.ppt

- [**Rus03**] Russ King and Tom Jelen, "**Tips and information for installing a dedicated server**", 2003.
- [**Spy04**] Spytech Software and design, "**Powerful Computer Spy Software**", inc., 2004

  http://www.spytech-web.com
- [**Tan03**] Tanenbaum S. Andrew, "**Computer Network"**, Prentice-Hall, Inc., Fourth Edition, 2003.
- [**Thi06**] ThinkQuest Technology, "**Three Tier Architecture**" ,2006

  http://www.tqnyc.org/tutorial/three_tier/index.php?s=.html
- [**Tom03**] Tom Jelen and Russ King, "**The basics of peer-to-peer networking**", 2003.

  http://www.techsoup.org/articlepage.cfm?ArticleId=210&topicid=3.
- **[USA96]** USAID Center for Development Information and Evaluation, "**Performance Monitoring and Evaluation**",

  http://pdf.dec.org/pdf_docs/pnaby215.pdf
- [**Van05**] Vangie Beal, "**client server architecture**", 2005.

  http://www.webopedia.com/TERM/C/client_server_architecture.html
- [**Wik05-a**] Wikipedia, the free encyclopedia, "**Local area network**", 2005.

  http://en.wikipedia.org/wiki/Computer_network.
- [**Wik05-b**] Wikipedia, the free encyclopedia, "**Metropolitan area network**", 2005.

  http://en.wikipedia.org/wiki/Metropolitan _area_network.htm
- [**Wil03**] William R. Cheswick and Steven M. Bellovion, "**Automated Network Monitoring**", 2003

  http://www.dcc.ichile/~rbaeza/i2p4.pdf
- [**Win02**] Windows XP professional service pack2 help and support center, Inc.,"**Peer-to-peer Architecture**", 2002.

Table of Contents

Chapter One : Introduction

Chapter Two : Network Monitoring Concepts

Chapter Three : Design of the Proposed System (MRMS)

Chapter Four : MRMS Implementation

Chapter Five: Conclusions and Future Work

جمهورية العراق
وزارة التعليم العالي والبحث العلمي
جامعة النهرين
كلية العلوم

بناء نظام مراقبة
متعدد المواقع

رسالة

مقدمة الى كلية العلوم في جامعة النهرين كجزء من متطلبات
نيل شهادة الماجستير في علوم الحاسوب

مقدمة من قبل

شيماء محمد رضا حمندي

(بكالوريوس علوم حاسوب ٢٠٠٢)

المشرفون

د. لمياء حافظ خالد **د. سوسن كمال ثامر**

٢٠٠٦ **١٤٢٧**

الخلاصة

لضمان استخدام الحاسبات ضمن الشبكة استخداماً جيداً في الجامعات و الشركات وغيرها، من المهم استخدام أنظمة المراقبة عن بعد (Remote Monitoring) لمراقبة المستخدمين و منع سوء أستخدام أي حاسبة. اضافة الى ذلك، عندما يعلم المستخدمون انهم مراقبون سوف يؤدي هذا الى تحسين ادائهم.

يهدف البحث الى تصميم و تنفيذ نظام مراقبة عن بعد لأكثر من حاسبة (MRMS)، (مراقبة آنية، وغير آنية) ضمن نطاق شبكة محلية. في المراقبة الآنية، يمكن مراقبة حاسبة او اكثر ضمن الشبكة مع السيطرة على بعض فعالياتها. النظام المقترح يزودنا بمعلومات عن فعاليات المستخدمين على الحاسبات البعيدة، بمراقبة اداء لوحة المفاتيح (keyboard)، مؤشر الفأرة (mouse)، ما يظهر على شاشة المستخدم و البرامج المنفذة عند المستخدم. يسمح النظام للمراقب (Administrator) التحكم بالحاسبات عن بعد عن طريق ارسال رسائل تحذير، غلق اي برنامج غير قانوني أو اطفاء حاسبة المستخدم.

في المراقبة غير الآنية، أي برنامج غير قانوني يغلق بواسطة النظام وفقاً لقاعدة بيانات يحددها المراقب. كذلك، كل المعلومات عن أداء لوحة المفاتيح و البرامج المنفذة للحاسبات البعيدة يتم خزنها في فايلات مخفية في تلك الحاسبات ويمكن للمراقب طلب أي معلومة من تلك الفايلات.