European Journal of Social Sciences
ISSN 1450-2267 Vol. 54 No 1 January, 2017, pp.79-91
http://www.europeanjournalofsocialsciences.com/

Network Traffic Monitoring Analysis System with Built-in Monitoring Data Gathering

Motaz Daadoo
Department of Computer Systems Engineering
Palestine Technical University - Kadoorie (PTUK), Tulkarm, Palestine
P.O. Box 7 Tulkarm City, Palestine
E-mail: m.daadoo@ptuk.edu.ps
Tel: +972-9-2688175; Fax: +972-2-2677922

Abstract

Networking, which is one of the most significant aspects of information technology revolution, is developing increasingly day after day. This is because it offers a huge amount of knowledge, resources and human experiences. On the one hand, it contains a considerable amount of harmful content, because of misusing. On the other hand, sitting for a long time in front of PC's or other network-based devices can affect body badly. As enterprise computing environments become more network-oriented, the importance of network traffic monitoring and analysis intensifies. Most existing traffic monitoring and analysis tools focus on measuring the traffic loads of individual network segments. Further, they typically have complicated user interfaces. This paper introduces and presents the design an application and implementation of an MS Windows-compatible software tool that is used to manage networks usage and keep track of every network user activity. An application consists of two parts client and server. The client side is a background-application runs whenever the PC is run, it turns off only when the PC is turned off and launched with its startup. The server side is more complex-GUI application that is responsible mainly for receiving data sent by clients group, managing and updating data to provide network owner up to date view. The effectiveness of an application has been verified by applying it to an enterprise network environment.

Keywords: Network traffic monitoring and analysis; Traffic management; Enterprise network management.

1. Introduction

As enterprise computing environments become more network-oriented, the importance of network traffic monitoring and analysis intensifies. Most existing traffic monitoring and analysis tools focus on measuring the traffic loads of individual network segments. Further, they typically have complicated user interfaces [Daadoo, M., Tarapiah, S., & Atalla, S. (2016), Evaluating]**,** [Daadoo, M., & Daraghmi, Y., (2015), Searching].

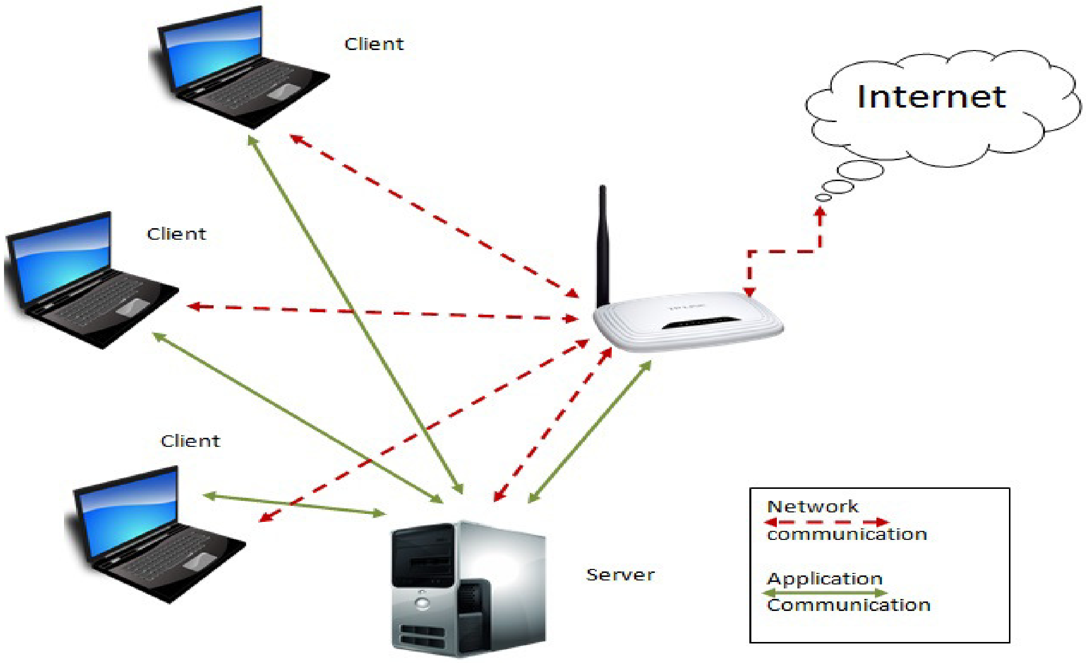
The objective of the proposed project here that it is designed an application to help networks owners to control their networks in a proper way by providing them necessary data and controlling permissions over their clients. If also aims to optimize accessibility and usability of control by providing a collection of variable demands. This work is done using three main well-known network protocols, which are TCP, RDP (Remote Desktop Connection Protocol) and HTTP. The first one is used to achieve reliable transmission of clients running apps, IPs, MACs and snapshots. The second one, which is RDP, is used to give server-part full control of wanted clients. The third one is used for communication with router to introduce more control privileges[Bär, A, et al., (2014)], [Fusco, F, et al., (2010)].

The proposed system consists of two parts: client and server. The client side is a background-application runs whenever the PC is run, it turns off only when the PC is turned off and launched with its startup. It contains set of classes built to provide information about client's status presented by: its basic access information (MAC and IP) addresses, a list of running applications and on-demand live snapshots. Firstly, the server side is more complex-GUI application that is responsible mainly for receiving data sent by clients group, managing and updating data to provide network owner up to date view. Secondly, it is responsible for sending control commands like logging off and shutting down to required station with ability to determine a period after which the command takes place. Thirdly, it provides facility of full control of the clients by using RDP protocol. Finally, it uses router information taken by http requests to show a list of possible illegal users, who have access to the network without having client program installed on their machines, and gives possibility of depriving them from accessing network anymore.

2. Proposed Model for Network Monitoring System

As shown in Figure 1, the proposed model is having bidirectional communication scheme, in which both server and client are working as sender and receiver simultaneously. The server receives a signal verify client state and bunch of basic information, and then it sends control commands. Besides, to the client does a reversed job by sending its information to the server and receiving commands from it. Moreover, network's router or access point plays an important role when it communicates with server over http requests giving data about network users and receiving commands of hosts filtering.

Figure 1: Proposed model



3. Software Design

Software can be mainly considered in two parts which are client and server parts. Each of them has its structure and its own device. The study is going to discuss client side and the server.

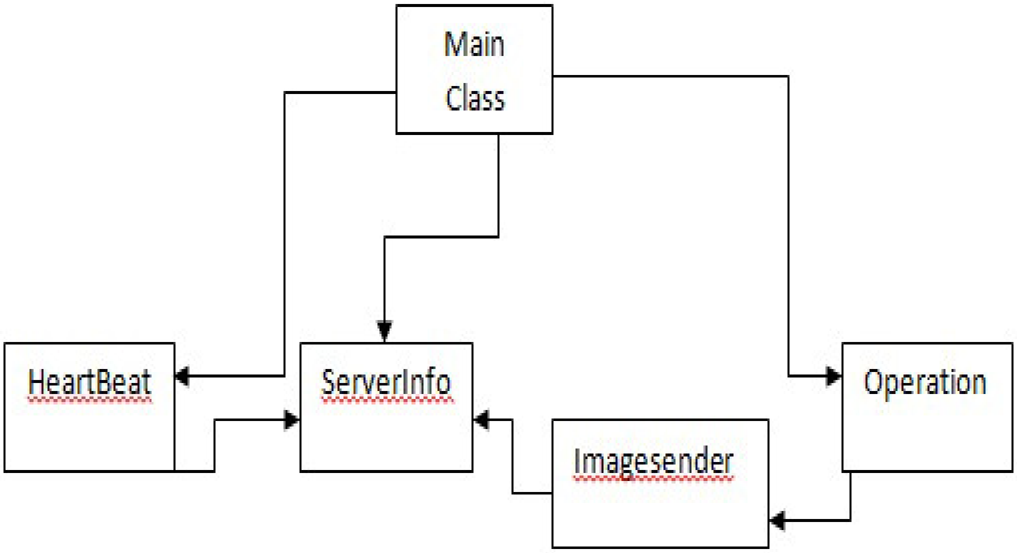
3.1 Client Side

The client part represents a non-interface background-running process that doesn't need any user interaction. Client software generally has to do three basic operations which are:

- Maintaining the server IP.
- Sending the heartbeat (a set of information makes server know that this client is living in addition to providing it client's access information and list of processes running in it).
- Sending a screenshots upon request so that server can track client's screen on demand.
- To perform these operations, client part consists of five classes as follows:
- **HeartBeat class:** This class is responsible for providing the server access information (IP and MAC addresses), and status information (List of running processes). This work is done continuously every 20 seconds to keep server up to date.
- **Imagesender class:** This class is responsible for sending an image to the server which is minimized first before being sent to get more performance. Image is provided when a command is received from the server. This operation is called from the server every half second to form like-video streaming.
- **ServerInfo class:** This class is responsible for providing server IP to the class that requires it. It first reads server IP from external file to maintain code flexibility and supply dynamic changes.
- **Operation class:** This class contains implementation to control commands that server sent. Operations supported are logoff, shutdown, mouse clicks, and take snap operation that take a snapshots and invoke image sender method to send it.
- **Main Class:** This is responsible for providing a receiver with commands sent to the client. It forms the registered lines of text coming in messages are interpreted and into handled commands.

The following Figure 2 is showing the class diagram of the client:

Figure 2: Client class diagram



As shown in the Figure above, the main class is maintaining either direct or indirect relationship with each of other classes. It initializes an object of HeartBeat class in the first place with associated server information taken from ServerInfo class. HeartBeat operation is a continuous operation that must be held all the time as an indicator of client process health, so every 20 seconds this operation is done continuously. Then it becomes ready to hear commands from the server in any time, in this case it is necessary to call the Operation class with the desired operation that is triggered by the server. Technical aspects of the main parts of the client side will be proposed in the part of proposed technique in the following pages.

Note: any client that accesses the network but not sending HeartBeat will appear in the server as a thief so client program should be run as soon as the client PC is running, so it is located in the startup programs in windows to be lunched when the windows boots up.

In the client part we use The TCP protocol because of its reliability features. It plays an important role in both sending and receiving data to and from the server. It provides a reliable information delivery for the commands sent from the server. However, it insures that the client heart beat is accessing the server without any problems.

3.2 Server Side

The server side is a bit complicated than client. This part is a GUI-based application that provides the user with the interface which concludes the state of network generally, and the client selected by admin controlling data. It consists of two main parts as it is responsible for doing two major jobs:

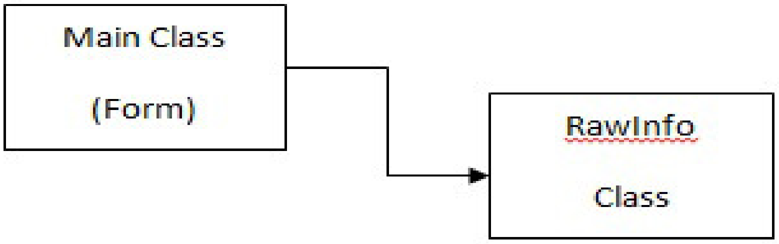
- Communicate with router to get network status.
- Monitor and send commands to the client.
- So it contains two classes which are:
- **Main class (Form):** This class contains functions that are responsible for sending commands to the clients and maintains a user view for what is happening on them.
- **RawInfo class:** This class is responsible for communication with router and it is the one that collects information from the router about the status of hosts connected currently to the router. It contains parts that handle router communication information and deal with basic authorization with http protocol.

Most of the server functionality, which is involved in the Form class in detailed description, does the following:

- Shutdown or logoff a client either soon or after a specific period.
- Show list of both valid and invalid hosts.
- Provide ability of MAC filtering to prevent illegal users.
- Take snapshots from a client.
- Control a client remotely.
- Show a list of running processes on a selected client.

The following Figure 3 is showing the class diagram of the server:

Figure 3: Server class diagram



Even it has less number of classes and simpler outer relations server classes are more complicated internally and are having more details compared with the client one's.

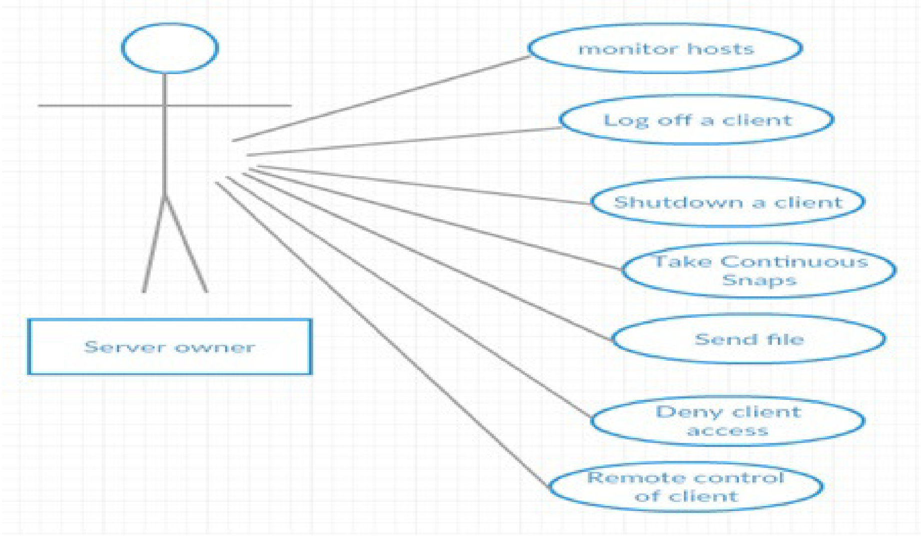
As class diagram illustrates the main class instantiates an instance of RawInfo class so that it will be able to get information from the router or access point about the network state and the active clients. And the rest of control and monitoring operations are encapsulated inside the form class.

In the server part three network protocols are used:

- **TCP (Transmission Control Protocol):** It provides reliable transmission of the server commands to the client. In addition to providing server connections information from the clients, the applications run on each client and on demand snapshots from the client's screen.
- **HTTP (Hyper Text Transfer Protocol):** This protocol is mainly used for communication with router. Because router provides its information in the form of web pages so http requests are the candidate tool to communicate with router. Doing this, authorization header of the http protocol should be manipulated because accessing to router needs to bypass router credentials which are implemented using basic access authentication. .net provides an HttpWebRequest class with necessary attributes to easily configure authorization part to deal with such cases.
- **RDP (Remote Desktop Protocol):** It is a secure network communication protocol for Windows-based applications. RDP allows network administrators to remotely diagnose and resolve problems encountered by individual subscribers. RDP is available for most versions of the Windows operating system as well as Mac OS X and open source version is available. Properties of RDP include encryption, smart card authentication, bandwidth reduction, resource sharing, the ability to use multiple displays and the ability to disconnect temporarily without logging off. RDP also allows redirection of functions such as audio and printing. This project uses the RDP protocol to provide admin to log in any of his clients and have a full control on it remotely which will greatly help when clients are facing some problems that need help to solve without efforts. Moreover, it provides an easy facility to transfer files any of the clients need from the server. However, .net especially C# doesn't fully support remote desktop visualization so the researcher uses ActiveX control to accomplish that part.

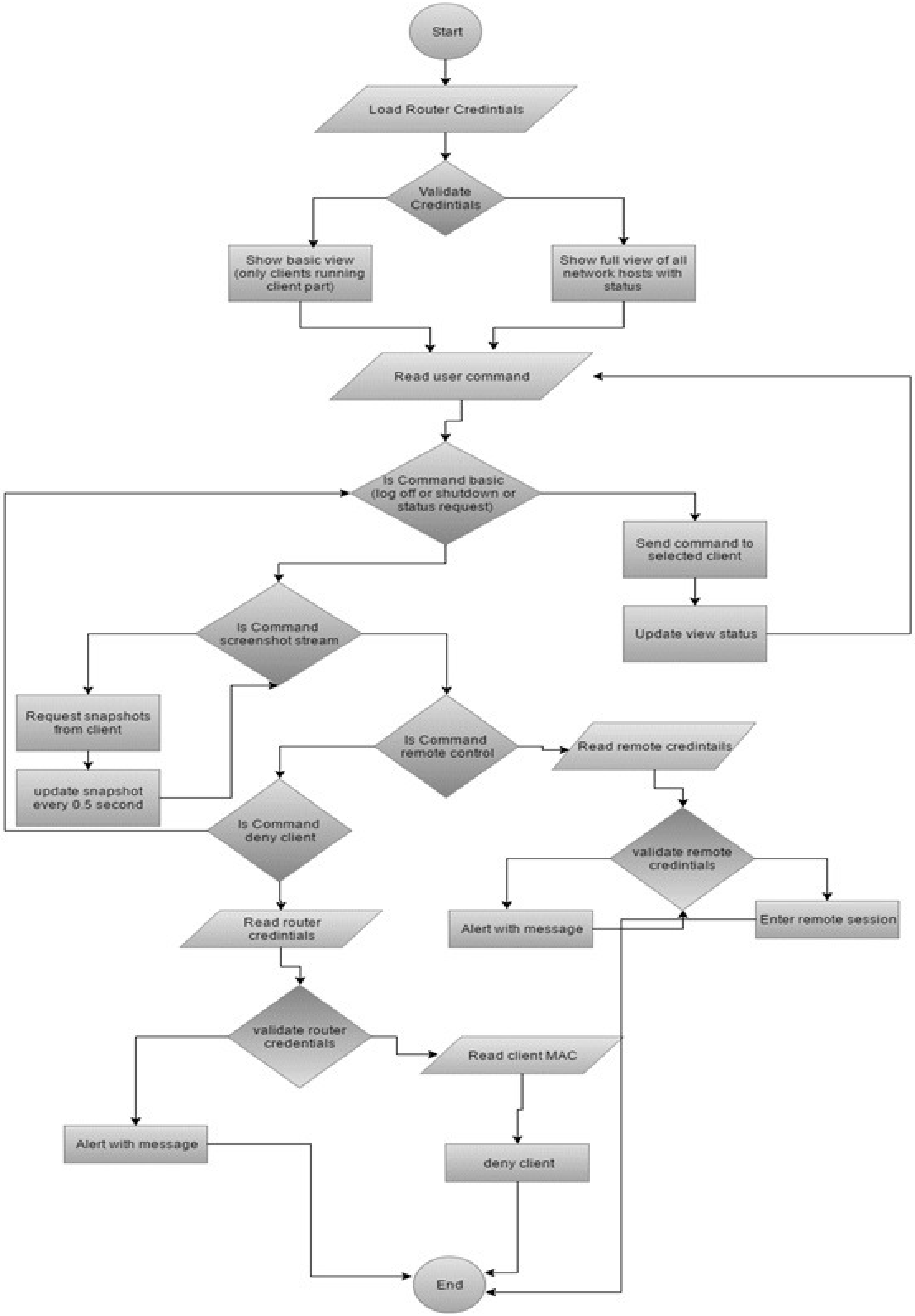
3.3 Use Case Diagram

Figure 4: Use case diagram



3.4 System Flow Chart

Figure 5: System flow chart


Start
Load Router Credintials
Validate Credintials
Show basic view (only clients running client part)
Show full view of all network hosts with status
Read user command
Is Command basic (log off or shutdown or status request)
Send command to selected client
Update view status
Is Command screenshot stream
Request snapshots from client
update snapshot every 0.5 second
Is Command remote control
Read remote credintails
Is Command deny client
validate remote credintials
Alert with message
Enter remote session
Read router credintials
validate router credentials
Read client MAC
Alert with message
deny client
End

3.5 Proposed Strategy

The proposed strategy is so effective and simple because it gives an easy and practical manner to monitor the network without any extra hardware or extra tools. The system uses exist networking protocols to communicate with router, and other devices and control them remotely. The pseudo code of the most important parts of the proposed model is given as the following:

Table 1: The pseudo code of the most important parts of the proposed model

```
Begin
If router credentials are incorrect
Show only clients which running service
End
Else
Show all active hosts and distinguish illegal ones
If server receives client beat
Add client, client Info and list of running programs on that client
Update clients' data every 20 seconds
Remove clients when they shutting down
Send command to the client and retrieve result
If client is illegal
Alert the user
If user wants to deny client
Access the router's privileges in router's control panel and deny wanted
If remote connection to a client is establishing
Check credentials if valid start session
Else
End session
End
```

3.6 Test Cases

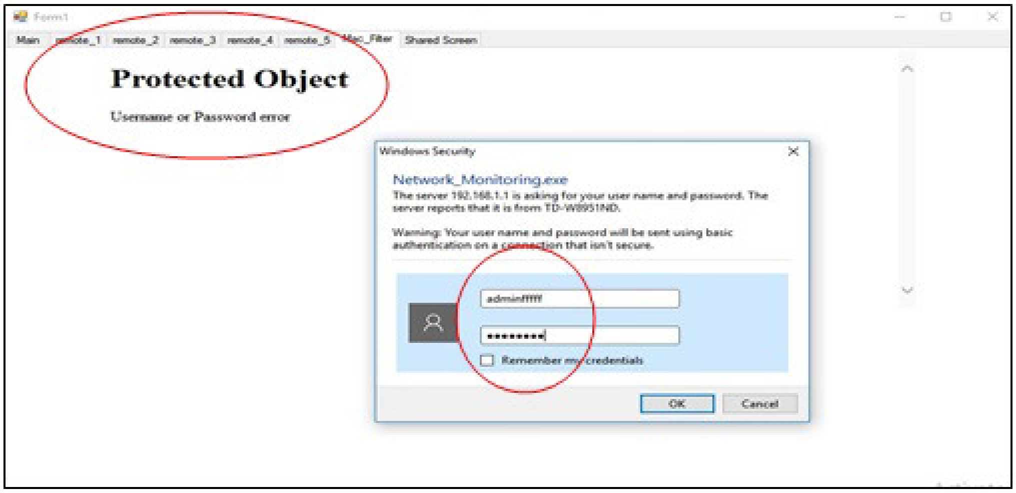
The system is tested with router of type TP-link but with various versions of windows. The client software is installed on windows 7, 8 and 10 versions which are versions used now.

3.6.1 Test Case 1

Title: Check router credentials.

System: server part (MAC filtering)
Input instructions: call filtering utility to deny a host with false authentication
Output: alert user and prevent access
Result: Test succeeded.

Figure 6: Test case 1



Protected Object

Username or Password error

Network_Monitoring.exe
The server 192.168.1.1 is asking for your user name and password. The server reports that it is from TD-W8951ND.

Warning: Your user name and password will be sent using basic authentication on a connection that isn't secure.

adminfffff

•••••••••

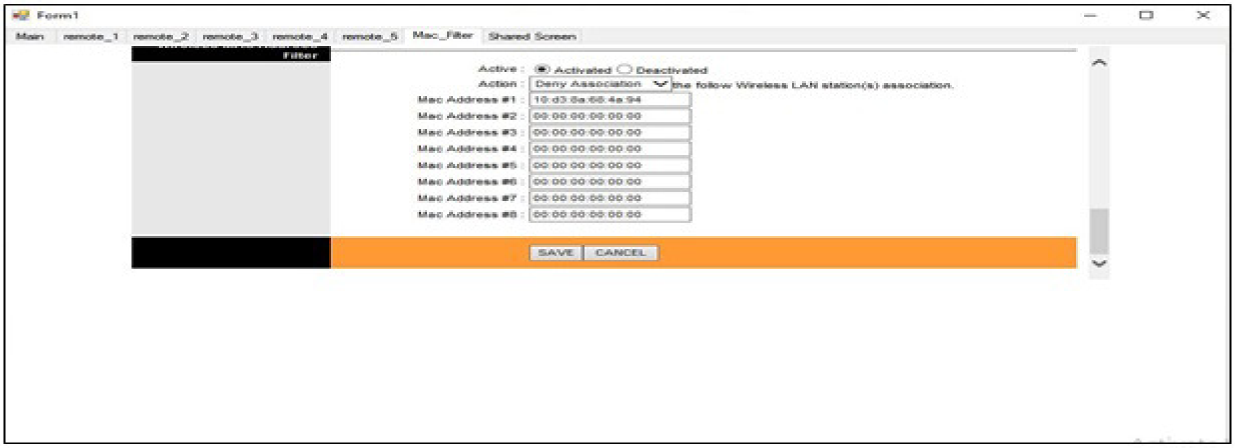
□ Remember my credentials

3.6.2 Test Case 2

Title: Check router credentials.

System: server part (MAC filtering)
Input instructions: call filtering utility to deny a host with true authentication
Output: deny access and disconnect selected client from the network
Result: Test succeeded.

Figure 7: Test case 2



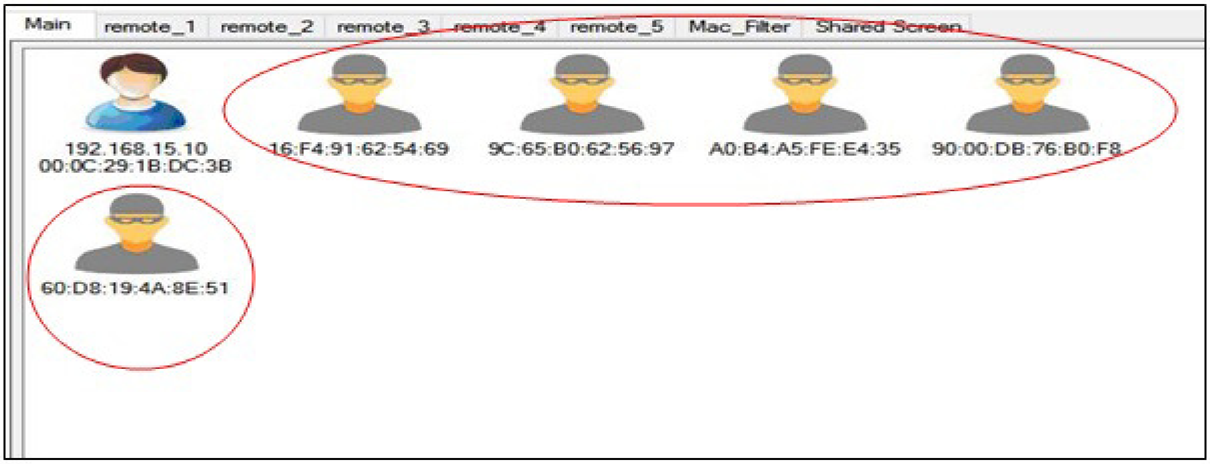
Active : ● Activated ○ Deactivated
Action : Deny Association the follow Wireless LAN station(s) association.
Mac Address #1 : 10:d3:8a:68:4a:94
Mac Address #2 : 00:00:00:00:00:00
Mac Address #3 : 00:00:00:00:00:00
Mac Address #4 : 00:00:00:00:00:00
Mac Address #5 : 00:00:00:00:00:00
Mac Address #6 : 00:00:00:00:00:00
Mac Address #7 : 00:00:00:00:00:00
Mac Address #8 : 00:00:00:00:00:00

3.6.3 Test Case 3

Title: Check client state

System: server part (main interface)
Input instructions: connect to the internet with a client without running client service
Output: alert user by showing MAC address and thief picture.
Result: Test succeeded.

Figure 8: Test case 3



192.168.15.10
00:0C:29:1B:DC:3B
16:F4:91:62:54:69
9C:65:B0:62:56:97
A0:B4:A5:FE:E4:35
90:00:DB:76:B0:F8
60:D8:19:4A:8E:51

3.6.4 Test Case 4

Title: Check server update on client shutting down

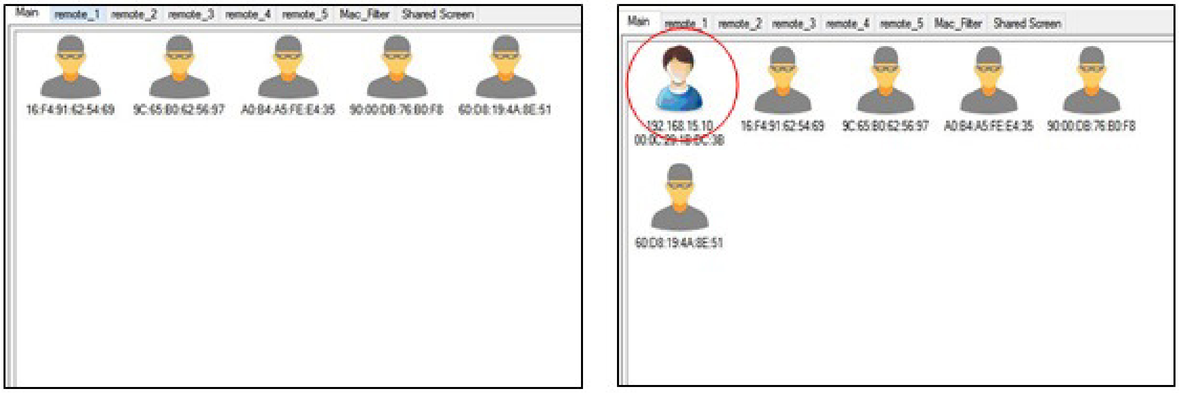
System: server part (main interface)
Input instructions: shut down a client
Output: after at most 20 seconds the server will be updated and client disappears.
Result: Test succeeded.

3.6.5 Test Case 5

Title: Check server update on client booting

System: server part (main interface)
Input instructions: lunch a client PC.
Output: after at most 20 seconds the server will be updated and client appears in list.
Result: Test succeeded.

Figure 9: Test case 4 & Test case 5



Test case 4 and 5 client disappears when shutting down and appears when boot up

3.6.6 Test Case 6

Title: Check operation (logoff, shutdown)

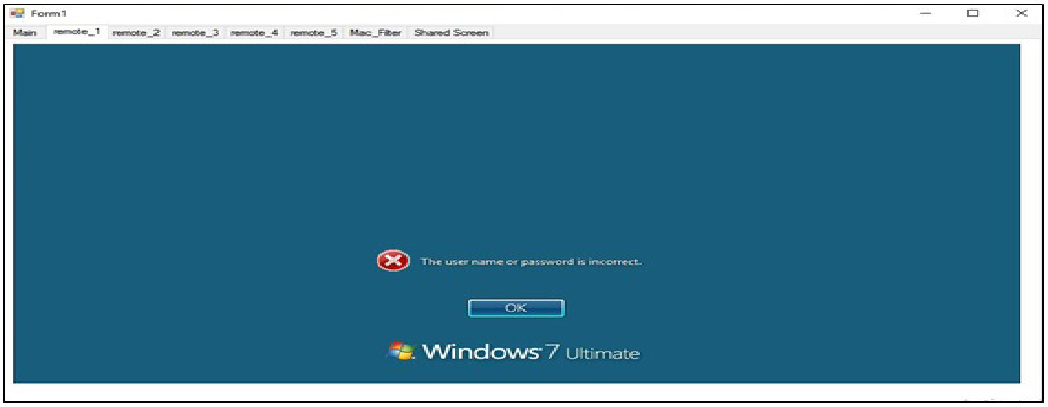
System: client part (OS).
Input instructions: send command to the client ex: shut down after 10 minutes
Output: after 10 minutes the client shuts down.
Result: Test succeeded.

3.6.7 Test Case 7

Title: Check remote control

System: server part (Remote client)
Input instructions: enter invalid client credentials.
Output: connection fails.
Result: Test succeeded.

Figure 10: Test case 6 & Test case 7



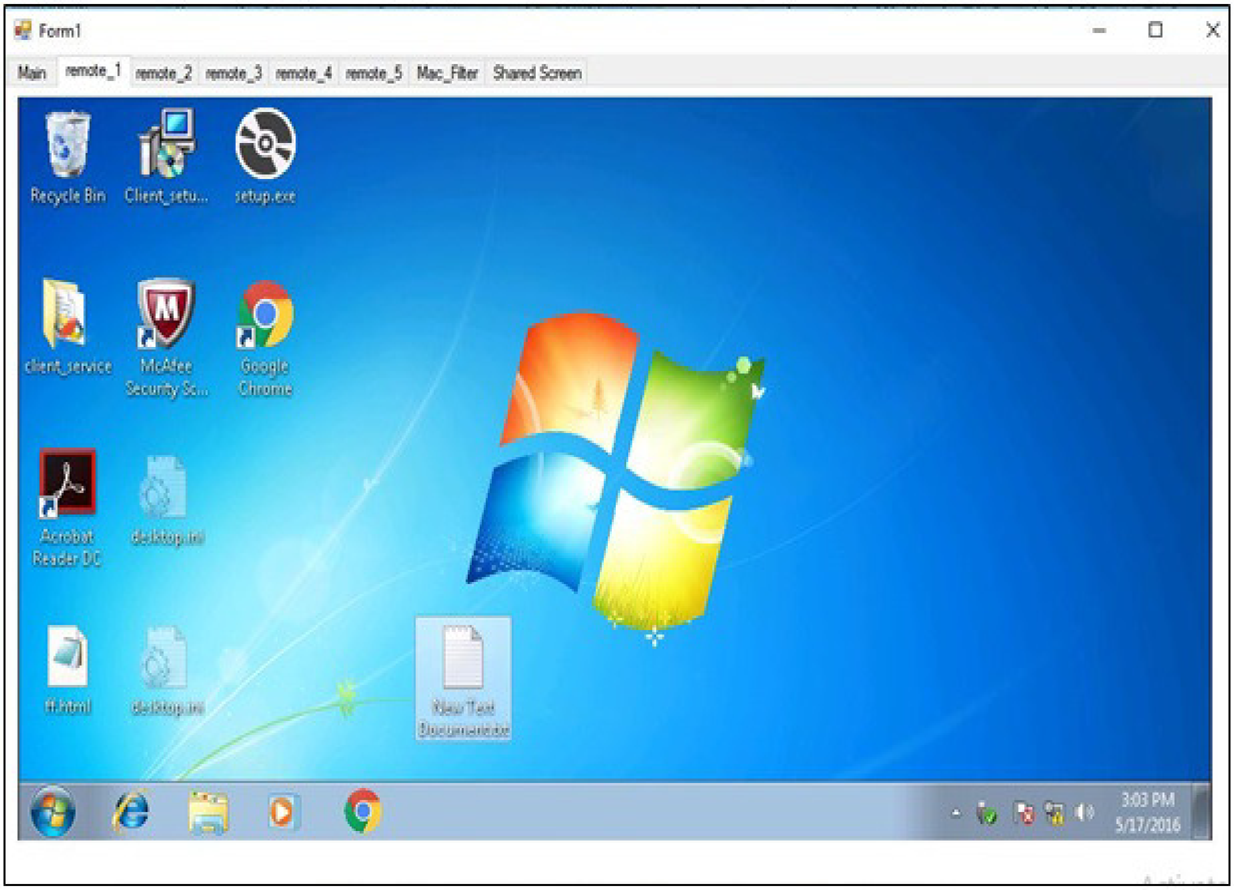
The user name or password is incorrect.

3.6.8 Test Case 8

Title: Check remote control

System: server part (Remote client)
Input instructions: enter valid client credentials.
Output: connection is established and the server takes full control.
Result: Test succeeded.

Figure 11: Test case 8



4. Results

Proposed project result is a complete tool that collects all information about your network and in a very-well organized scheme. This forms a friendly mechanism to monitor a network and manage its configurations.

Precisely using this system, you can:

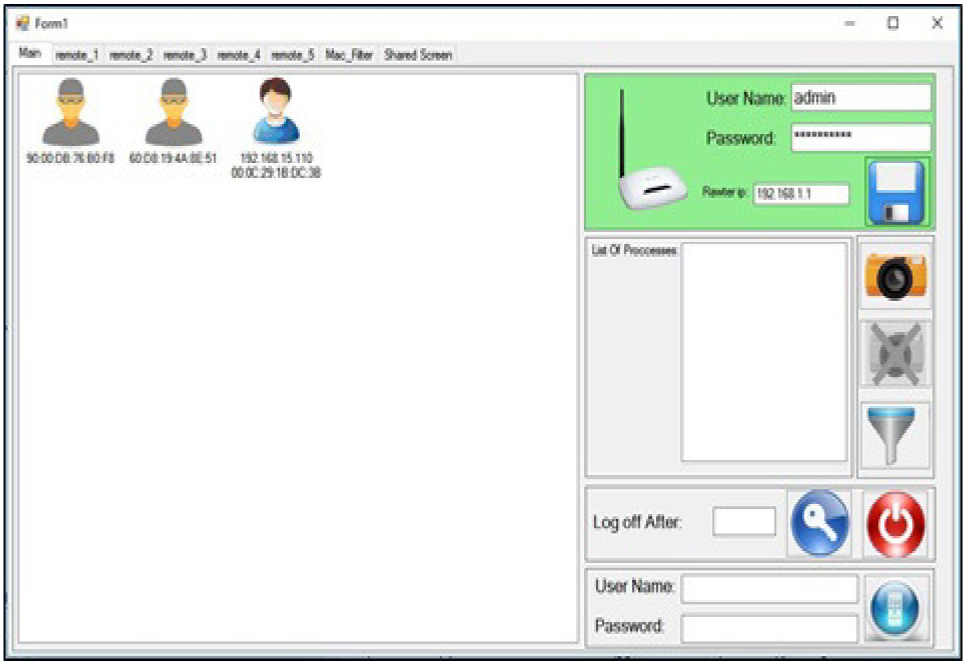
- Get helpful information about your network.
- Keep a track of clients' activities.
- Control clients operating periods.
- Organize access to the internet by editing permissions and denying illegal access.
- Share a client screen and monitor them.
- Send files to clients.
- Get a remote full control of clients concurrently.

All of this is given in one place and in a high performance and instantaneous results.

The following snapshots represent the main screens used to perform the application:

- This is the main interface it contains all control buttons in clients and router and displays network state and hosts information.(see figure 12).

Figure 12: Main Screen



User Name: admin

Password: **********

Router ip: 192.168.1.1

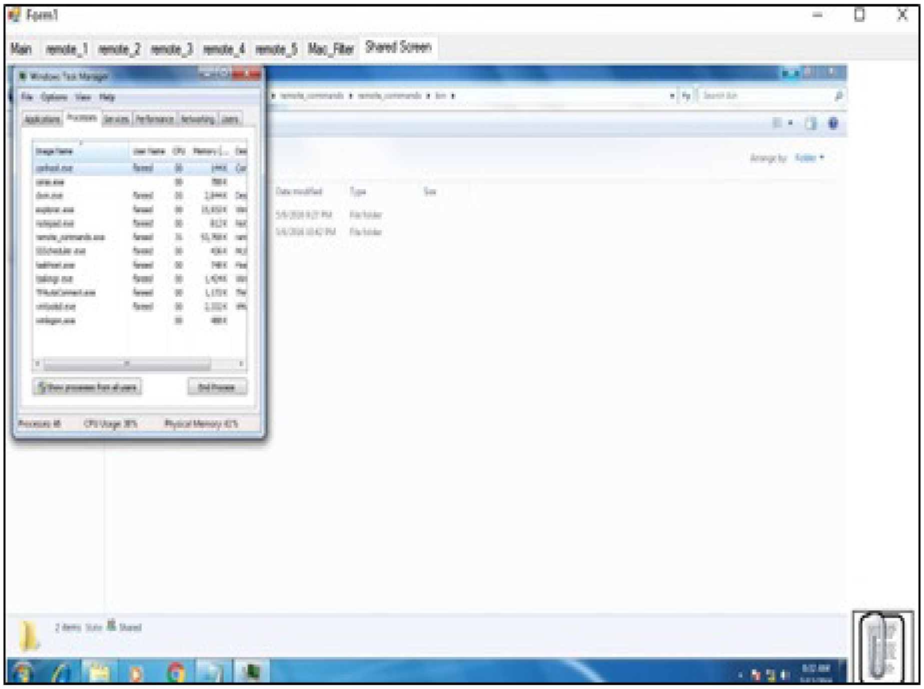
Log off After:

User Name:

Password:

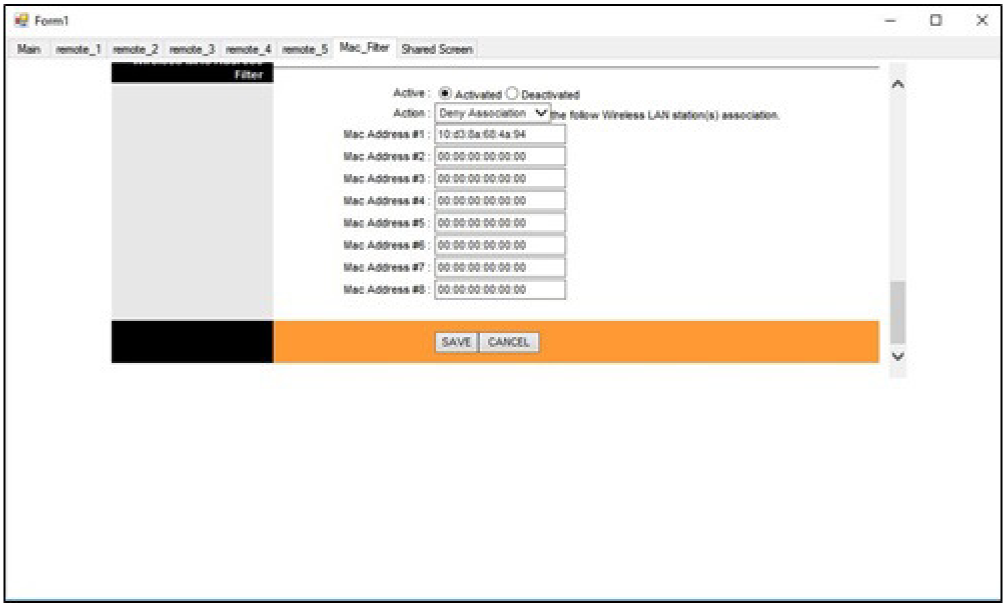
- Client appear inside the server program using screen sharing and file, so screen can be transferred to that client using file transfer button on the right. (see figure 13)

Figure 13: Shared screen and file transfer



- MAC filtering to deny access any client by adding its MAC address inside MAC filter fields and activate filtering mode. (see figure 14)

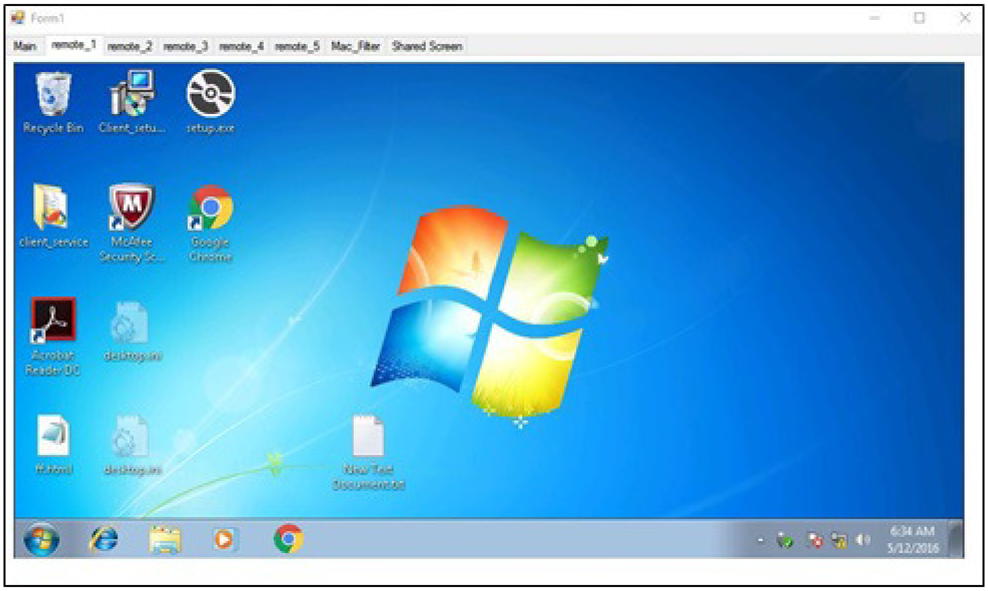
Figure 14: MAC Filter



Active : ● Activated ○ Deactivated
Action : Deny Association the follow Wireless LAN station(s) association.
Mac Address #1 : 10:d3:0a:60:4a:94
Mac Address #2 : 00:00:00:00:00:00
Mac Address #3 : 00:00:00:00:00:00
Mac Address #4 : 00:00:00:00:00:00
Mac Address #5 : 00:00:00:00:00:00
Mac Address #6 : 00:00:00:00:00:00
Mac Address #7 : 00:00:00:00:00:00
Mac Address #8 : 00:00:00:00:00:00

- Remote full control of another host. (see figure 15)

Figure 15: Remote each host and fully controlling



5. Conclusion

In this study of network management system, the proposed approach is implementing a set of well-known networking protocols to find everything about them and monitor the full state of the network. It is an efficient method that provides all needed knowledge about network suitable way that optimizes the needed owner interactions that is necessary to configure things as desired. The solution is programmatically modeled by using a set of simple and effective algorithmic techniques that manage client's communications do requests to the router pages, file transferring and screen sharing with basic controlling.

Moreover, the way in which application designed improves user control efficiency when using remote techniques, because of providing control of up to five devices concurrently adjacent tabs that control devices by visiting from remote sessions and windows remote connection utility, which provides each remote session in an independent window that causes difficulty in managing them by two remote sessions.

Acknowledgment

The author would like to thank Palestine Technical University – Kadoorie (PTUK) for supporting this research and allowing him to conduct this work in the university labs. The system is developed for educational purpose and is currently in use at Computer Networks Lab at University.

References

[1] Bär, A., Finamore, A., Casas, P., Golab, L., & Mellia, M. (2014, October). Large-scale network traffic monitoring with DBStream, a system for rolling big data analysis. In Big Data (Big Data), 2014 IEEE International Conference on (pp. 165-170). IEEE.

[2] Daadoo, M., Tarapiah, S., & Atalla, S. (2016). Evaluating Efficiency of Multi-Layered Switch Architecture in All-Optical Networks. International Journal of Applied Engineering Research, 11(22), 11030-11036.

[3] Daadoo, M., & Daraghmi, Y. (2015, August). Searching of optimum characteristics of multi-layer switching architecture in all-optical networks. In Heterogeneous Networking for Quality, Reliability, Security and Robustness (QSHINE), 2015 11th International Conference on (pp. 50-55). IEEE.

[4] Fusco, F., & Deri, L. (2010, November). High speed network traffic analysis with commodity multi-core systems. In Proceedings of the 10th ACM SIGCOMM conference on Internet measurement (pp. 218-224). ACM.

[5] Remote Desktop Protocol. Wikimedia Foundation, n.d. Web. 18 June 2016.