SILENT NETWORKING USING FUZZY LOGIC FOR POWER SAVING IN NETWORKED DEVICES

by

Prashant Singh

Submitted in partial fulfilment of the requirements
for the degree of Master of Computer Science

at

Dalhousie University
Halifax, Nova Scotia
March 2012

DALHOUSIE UNIVERSITY

FACULTY OF COMPUTER SCIENCE

The undersigned hereby certify that they have read and recommend to the Faculty of Graduate Studies for acceptance a thesis entitled "SILENT NETWORKING USING FUZZY LOGIC FOR POWER SAVING IN NETWORKED DEVICES" by Prashant Singh in partial fulfilment of the requirements for the degree of Master of Computer Science.

Dated: March 29th 2012

Supervisor: ______________________________

Co-supervisor: ______________________________

Reader: ______________________________

DALHOUSIE UNIVERSITY

DATE: March 29th 2012

AUTHOR: Prashant Singh

TITLE: SILENT NETWORKING USING FUZZY LOGIC FOR POWER SAVING IN NETWORKED DEVICES

DEPARTMENT OR SCHOOL: Faculty of Computer Science

DEGREE: MCSC CONVOCATION: May YEAR: 2012

____

Signature of Author

Table of Contents

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

A lot of work has been done in developing energy efficient network and user devices to reduce the power consumption of nodes and devices in networks. This thesis proposes an innovative approach using fuzzy logic for power saving and extending the life time of network nodes and user devices.

Using the concept of silent networking we will define an actionable silent period-this is the period during which the network node or the user device does not expect to originate, receive or relay any traffic. The decision of switching the network interface or user device in the silent mode depends on the history of the network activity. Secondly, if the actionable silent period is high enough, then we can switch the entire interface in power down mode leaving just the timer ON to wake up the interface at the end of silent period.

Fuzzy logic is used in mapping the history of the network interface and based on the fuzzy rules that we define, the actionable silent period for interfaces is formulated. Experimental analysis using simulations has been done to view the power saving that can be achieved using this method. Furthermore, a methodology for extending the lifetime of the networked devices is formulated. Using this innovative approach we can save a considerable amount of energy and proportionally increase the lifetime of the networked devices.

ACKNOWLEDEGEMENT

Foremost, I would like to express my sincere gratitude to my supervisor **Dr. Srinivas Sampalli** and co-supervisor **Dr. Denis Riordan** for their continuous support of my study and research, for their patience, motivation, enthusiasm, and immense knowledge. Their guidance helped me during all the time of my research and writing of this thesis. I could not have imagined having a better supervisor and co-supervisor for my Master's thesis study.

Besides my advisor, I would like to thank the rest of my thesis committee: Prof. **Dr. Nur Zincir-Heywood** for their encouragement, insightful comments, and hard questions.

Last but not least, I would like to thank my family: my parents **Chandra Bhushan Singh** and **Veena Devi**, for giving birth to me in the first place and supporting me spiritually throughout my life; also thanks the girl that has faith in me, thank you all!!!!

Chapter One: Introduction

1.1 Green Computing

To start this topic, first we need to know the concept of green computing and its origin. There are different meanings to both the terms 'green' and 'computing'. 'Green' is used in everyday language to refer to environmental sustainable activities. Green Computing encompasses policies, procedures, and personal computing practices associated with any use of information technology (IT). People employing sustainable or green computing practices strive to minimize greenhouse gases and waste, while increasing the cost effectiveness of IT, such as computers, local area networks and data centres [3]. Green computing can be described in another way as a process where design, manufacture, use and disposal involves as little environmental impact as possible. In other words, green computing is an initiative taken in consideration of all facets of a computer's life, from design to disposal [7].

1.2 Silent Networking

In the past decade or so, significant progress has been achieved in power saving using dynamic power management approaches for network nodes and mobile devices. A major progress can be seen in mobile devices radio transmission and sensors networks, because high energy consuming radio interfaces are turned off whenever possible as proposed in IEEE 802.11 [5, 6].

Silent networking is an innovative idea proposed by [3]; using this method one can significantly increase the life time of the energy constrained nodes, networks, and mobile user devices. The concept of silent networking comes into being when the network nodes or user devices power down some or all of their network interfaces when they do not expect to originate, receive or relay any traffic through their interfaces.

Silent networking [3] significantly increases the life time of the energy-constrained nodes, networks and user devices without proposing or making any changes to the current protocol, networks or to the user devices. Also, the nodes or user devices make their autonomous decisions about switching off network interfaces or user devices based on the previous records of the traffic.

The focus of this thesis is to incorporate silent networking to enable green computing by decreasing the power consumption and increasing the life time of the networked devices [3, 7].

1.3 How silent networking relates to green computing

The idea behind the concept of silent networking is to reduce the power consumption of the network interfaces and user devices as well as to extend the lifetime of the devices. This is in a direct way related to the environment, as to generate electricity we use various different types of natural resources such as coal, petroleum products, and nuclear energy.

In the same way by extending the lifetime of the devices we can reduce the manufacturing of network nodes and user devices compared to what it would be if we did not use the above-proposed approach. Here we can see that by using the concept of silent networking we avoid the usage of natural resources, i.e., used for manufacturing of devices to certain extent [1, 3].

1.4 Introduction to Paessler

Paessler Router Traffic Grapher (PRTG) is a software developed by Paessler, a network monitoring company [2]. PRTG is a very highly compact and easy to use software that monitors bandwidth usage and various other network parameters via SNMP (Simple Network Management protocol), packet sniffing and Cisco Net flow. Apart from monitoring the amount of data flowing through the router or device, it monitors CPU utilization, analyzes the traffic by type, or checks the disk space usage [2, 16].

Network monitoring is becoming the backbone of any company as today most businesses rely more or less on a computer and network infrastructure. Thus, the computer network's reliability and speed is an important tool in the company's success and avoiding losses.

PRTG [2] runs on a Windows platform in the network 24/7 and constantly records the network usage parameters. The recorded data is stored in the internal database for later reference [2].

1.5 Introduction to MATLAB

The name MATLAB [1] stands for Matrix laboratory. MATLAB was originally written to provide easy access to matrix software developed by the LINPACK and EISPACK project. MATLAB is a high-level technical computing language and interactive environment for algorithm development, data visualization, data analysis, and numerical computation. MATLAB [1], can solve technical computing problems faster than traditional programming languages, such as C, C++, and Fortran.

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. MATLAB has evolved over a period of years with input from many users. Currently MATLAB features a family of add-on application with specific solution called toolboxes. This is of great use for the user who wants to use a particular feature or specialized technology. Toolboxes are a comprehensive collection of MATLAB functions (M-files) that extend the MATLAB environment to solve particular classes of problems [1, 8].

Typical Uses include:

- Math and computation.
- Algorithm development.
- Data acquisition.
- Modeling, simulation, and prototyping.
- Data analysis, exploration, and visualization.
- Scientific and engineering graphics.
- Application development, including graphical user interface building.
- Solving logic problems with the help of toolboxes available.

One can also add on toolboxes such as

- Signal processing
- Neural networks
- Control system
- Wavelets
- Simulation
- Fuzzy logic and may others [1, 8].

1.6 Introduction to Fuzzy logic

The footprint of Fuzzy logic has been felt for a long time but its presence and real application started in the 1990's. This is the period during which we all started feeling the need for the various devices that can perform our tasks with just a tap or click of a button. This has led to a new trend in consumer products such as household appliances, camera and other such products. Fuzzy logic began with the 1965 proposal of fuzzy set theory by Lotfi Zadeh [9].

Fuzzy logic is much more general than traditional logical systems. The greater generality of fuzzy logic is needed to deal with complex problems in the realms of search, question-answering decision and control. Fuzzy logic provides a foundation for the development of new tools for dealing with natural languages and knowledge representation. Among these tools are: Precisiated Natural Language (PNL), Computing with Words (CW), Computational Theory of Perceptions (CTP), Protoform Theory (PT), Theory of Hierarchical Definability (THD), Unified Theory of Uncertainty (UTU); Perception-Based Probability Theory (PTp) [20].

To understand why fuzzy logic has gained popularity and made such a leap and bounce progress in the recent times. First, we need to understand the concept of fuzzy logic? "Fuzzy Logic is aimed at a formalization of modes of reasoning which are approximate rather than exact"[Lotfi A. Zadeh, 12]. Consider the following example,

1. Exact

All women are mortal

Kelsey is a woman

Kelsey is mortal

2. Approximate

Most Canadians are tall

Grosby is a Canadian

It is likely that Grosby is tall [17]

Fuzzy logic is all about the relative importance of precision: How important is it to be exactly right when a rough answer will do?

In this sense, we can say that fuzzy logic is both old and new. The reason behind this is, although the modern and methodical science of fuzzy logic is still young, the concept of fuzzy logic relies on age-old skills of human reasoning. Human reasoning is one of the fundamental tools to the concept of fuzzy logic and its development and this makes it different form other Artificial Intelligence logical concepts [17].

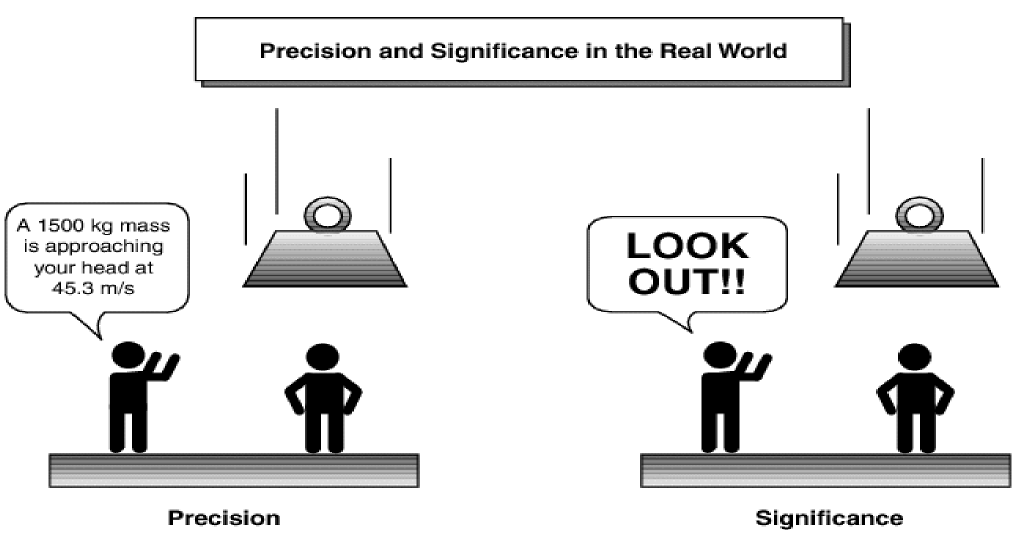


FIG. 1.1 PRECISION AND SIGNIFICANCE IN THE REAL WORLD [MATLAB, 17]

In the above image one can see that there are two ways to say the same thing "Falling of a weight on someone head".

a) Precise—"A 1500 kg mass is approaching your head at 45.3 m/s."

b) Approximate—Please "LOOK OUT". This is more significant than the previous one as the person on whom the object is falling doesn't need to know any information about the object, its mass or its velocity. The only thing that matters here is the safety of the person [10, 17].

Here, being approximate is many times better than being precise and this is where the concept of fuzzy logic comes into being.

Now the next question that arises is, why do we need to use fuzzy logic?

Below are some of the general observations about fuzzy logic:

1. Fuzzy logic provides flexibility

With any given system, if one wants to make it more robust and intelligent one can just need to add layers of functionality without starting from scratch. This is important as it saves the time and efforts to build the system again.

2. Fuzzy logic is easy to understand and explain

The mathematics calculation behind fuzzy logic is very simple as it all depends on the intelligence and reasoning level of the person. Fuzzy logic is a more intuitive approach without far-reaching complexity [8, 17].

3. Fuzzy logic has high level tolerance for imprecise data

Almost everything is imprecise if you look closely enough and here is the base for fuzzy logic. Fuzzy reasoning builds this understanding into the process rather than tacking it onto the end [1, 12].

4. Fuzzy logic is based on natural language

The basis for fuzzy logic is the basis for human communication. This observation underpins many of the other statements about fuzzy logic. Because fuzzy logic is built on the structures of qualitative description used in everyday language, fuzzy logic is easy to use [9, 17].

5. Fuzzy logic can be built on top of the experience of experts

In direct contrast to neural networks, which take training data and generate opaque, impenetrable models, fuzzy logic lets you rely on the experience of people who already understand your system.

1.7 Objective of thesis

The main objective of this thesis is reducing the power consumption of the network nodes (routers, switches etc.) and user devices. My thesis revolves around networked devices and it is basically designed for saving power, which is the need of the future and with the increase in the number of devices connected to the internet and the concepts like Wi-Fi, WiMax, IPv6 we should explore different ways of saving power as the number of devices is increasing tremendously.

Below are some of the points on which my thesis will concentrate,

1) How can we switch the network node and user device to silent mode, if it is not originating, receiving or relaying any packets?
2) How are the packets and power related to each other?
3) How significant is the power saving compared to without this proposed prototype model?
4) What is the effect of this power saving on the life time of the device?
5) Does it make or propose any changes to the current protocol?
6) Can we achieve higher power saving, if we make changes to the current protocol?

1.8 Outline of the thesis

The prototypes of the model have been evaluated with experimental analysis. Software PRTG [2] runs on the server which collects the required data and when this information about the packets is fed to the fuzzy logic tool box of Matlab [1] we can make the rules to decide when to switch the devices to different states. The experiment is explained in detail ahead.

Chapter 2: In this chapter we will discuss about the background of silent networking, operational concept of Paessler, MATLAB and fuzzy logic tool box.

Chapter 3: This chapter explains the proposed scheme and design overview.

Chapter4: This chapter explains prototype implementation and experimental results.

Chapter 5: This chapter contains the conclusion of the result and explains the further implementation of the proposed method.

1.9 Literature review

1. T. Zhang *, P. Gurung, E. Berg, S. Madhani, A. Muttreja, "**silent networking for energy-constrained nodes**" March 2006.

This paper is about silent networking for nodes using mathematical model based on the concept of mobile network and sensors.

2. A. Sinha, A. Chandrakasan,"**Dynamic power management in wireless sensor networks**" 2001.

Author propose an OS-directed power management technique to improve the energy efficiency of sensor nodes.

3. W. Akkari, A. Belghith, A. Mnaouer, "**Enhancing power saving mechanisms for ad hoc networks using neighborhood information**" 2008.

This papers explains how neighbouring information can be used in ad hoc network for enhancing the power saving.

4. C. Jones, K. Sivalingam, A. Agarwal, J. Chen, "**A Survey of Energy Efficient Network Protocols for Wireless Networks ∗**" 2001.

The authors in this paper has done a survey on the power consumption for various devices in the Wireless networks.

5. X. Liao, L. Hu, H. Jin, "**Energy optimization schemes in cluster with virtual machines**",Nov 2009.
The authors in this paper talks about the power saving in virtual cluster machines like cloud computing and data center.

6. R. Kravets, P. Krishnan, "**Power management techniques for mobile Communication**", 1998.
In this paper, the author presents a design and implementation of an innovative transport level protocol capable of significantly reducing the power usage of the communication device.

"Our proposed approach is a novel one because it incorporates fuzzy logic for power saving and increasing the lifetime of the device. Using this concept, we can achieve higher power saving and longer lifetime of the device compared to the mathematical method."

Chapter Two: Background

2.1 Background concept of silent networking

The term silent networking is new but the concept of silent networking is not. This concept of silent networking is used in mobile devices to turn off the radio interface when the device is not relaying any traffic [3, 13]. Below is the description of its working in mobile devices.

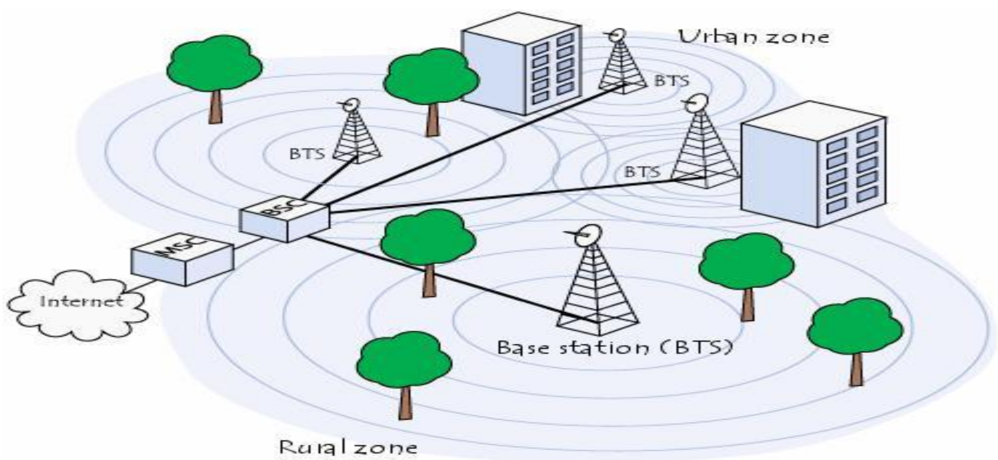


FIG. 2.1 MOBILE NETWORK CONNECTION PROTOTYPES

Every mobile device is connected to a base transceiver station (BTS) and receives signal (bandwidth) corresponding to the work the mobile device is performing. However, when the user is not using his mobile device the network connection is there but the bandwidth allotted to that particular mobile device is very low [3, 13, 18]. This method of allotting bandwidth to the mobile device which requires the connection for the various tasks it performs is called Frequency Division Multiple Access (FDMA) [13].

This method of silent networking in mobile devices is helpful in many ways. A few of them are,

1. It reduces the number of BTS in the area.

2. It reduces the transmission of unnecessary signal waves.
3. It reduces the power consumption of devices i.e. it increases the “STAND ON” time of a mobile device.
4. If the device is not used to send the signal continuously than its life time is bound to increase and this is one of the most important parameters as people cannot afford to frequently buy expensive mobile devices day in day out.
5. Because of the use of this idea in mobile devices and the increase in the life time of the device, the rate at which mobile devices are made is less compared to what it would have been if we had not used the above idea.
6. Hence, in a way we are reducing the use of natural resources and, disposing of the devices [3, 18].

Now, as we all know that, routers, switches etc. are the backbone of the Internet, and significant attention has been paid toward increasing the capacity, speed, range and connectivity of the devices compared to the amount of attention paid to making them environmentally friendly. We need to make them more environmentally friendly as we all know that our entire human race future depends on how we make use of our natural resources and how much less we pollute our environment [3, 17].

Using the concept of silent networking proposed in [3] we can significantly reduce the power consumption of the device by a considerable amount and due to this at the same time we can increase the life time of the device.

This concept is more useful when it is implemented on a large scale and practised by everyone as is the case with laptops. Individual efforts of saving energy might not have helped in reducing the power consumption of the laptop if it was not programmed to switch to power saving modes when it is not in use [3, 18]. So, one way of saving power is making the device intelligent enough to save power when it is not connected to any devices or when it sees that some of the devices are not using the connection and hence switching the devices to silent mode [14].

2.2 Background concept of Paessler

The PRTG Network Monitor [2] can support thousands of sensors and can optionally work with multiple remote probes to monitor multiple sites or network segments from one central core installation. You can also configure fail-safe monitoring using a cluster installation. PRTG Network monitoring software was introduced in 1997 and since then it has been constantly updated with new features and advancement in the networking field. Currently there are 150,000 users of the software. The best part of using this software is that you do not need to use any third party software.

What PRTG can be used for,

- Monitor and alert for uptimes/downtimes or slow servers.
- Monitor and account for bandwidth and network device usage.
- Monitor system usage (CPU loads, free memory, free disk space etc.).
- Classify network traffic by source/destination and content.
- Discover unusual, suspicious or malicious activity with devices or users.
- Measure QoS and VoIP parameters and control service level agreements (SLA).
- Discover and assess network devices.
- Monitor fail-safe using a failover cluster setup.[2]

Below is the homepage for the PRTG [2], from here it starts performing all the different operations required for monitoring the network and various sensors of the devices.

FIG. 2.2 IMAGE OF THE HOME PAGE OF PRTG NETWORK MONITOR [2]

One of the issues faced in the previous versions was of failure of the server node which has been taken care of in the latest release under "Clustering Mode". One can set up up to four additional nodes in cluster mode for fail-safe, gap-less monitoring. The best feature of the clustering mode is that all the fail-safe nodes monitor the data individually. This allows the user (administrator) to compare the data acquired from the different nodes. Fig. 2.3 illustrates the PRTG software in cluster mode,

FIG. 2.3 ILLUSTRATION OF A SINGLE FAILOVER CLUSTER [5]

Architecture of PRTG:

This is one of the important parts while setting up the PRTG software. One has to know about the architecture as this will increase the knowledge of the user and help him/her understand the different steps involved in setting up the PRTG software for monitoring different devices with different sensors.

The architecture is divided into different parts and all of these parts are categorized in three different categories.

Table 2.1 Architecture, parts and category of PRTG [2]

Types	PRTG parts and there functional explanation
System Parts	**Core Server:** This is the central part of a PRTG installation and includes data storage, web server, report engine, a notification system, and more.
	Probes: The part of PRTG on which the actual monitoring is performed. There are local probes, remote probes, and cluster probes available. All monitoring data is forwarded to the central core servers.
User Interfaces	**Ajax Web Interface:** The Ajax-based web interface is used for configuration of devices and sensors, as well as the review of monitoring results. Also system administration and user management are configured here.
	Enterprise Console: A native Windows application (former Windows GUI) as alternative to the web interface to manage your monitoring. With the Enterprise Console, you can connect to different independent PRTG core server installations and review data at a glance!

	Smart Phone Apps: Monitor your network on the go with PRTG and the smart phone apps for iOS and Android.
System Administration Programs	**PRTG Server Administrator:** Used to configure basic core server settings, such as administrator login, web server IPs and port, probe connection settings, cluster mode, system language, and more.
	PRTG Probe Administrator: Used to configure basic probe settings such as name of the probe, IP and server connection settings, and more.

2.3 Operational concept of MATLAB and Fuzzy logic tool box

a) MATLAB

As we have discussed above, "The MATLAB ® high-performance language for technical computing integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation" [1].

Basically MATLAB consists of the following six parts (Table 2.2 and Fig. 2.4),

Table 2.2 MATLAB parts [1]

Desktop Tools and Development Environment	This part of MATLAB is the set of tools and facilities that help you use and become more productive with MATLAB functions and files. Many of these tools are graphical user interfaces. It includes: the MATLAB desktop and Command Window, an editor and debugger, a code analyzer, and browsers for viewing help, the workspace, and folders

	[1].
Mathematical Function Library	This library is a vast collection of computational algorithms ranging from elementary functions, like sum, sine, cosine, and complex arithmetic, to more sophisticated functions like matrix inverse, matrix eigenvalues, Bessel functions, and fast Fourier transforms [1, 12].
The Language	The MATLAB language is a high-level matrix/array language with control flow statements, functions, data structures, input/output, and object-oriented programming features. It allows both "programming in the small" to rapidly create quick programs you do not intend to reuse. You can also do "programming in the large" to create complex application programs intended for reuse [1, 12].
Graphics	MATLAB has extensive facilities for displaying vectors and matrices as graphs, as well as annotating and printing these graphs. It includes high-level functions for two-dimensional and three-dimensional data visualization, image processing, animation, and presentation graphics. It also includes low-level functions that allow you to fully customize the appearance of graphics as well as to build complete graphical user interfaces on your MATLAB applications [9, 12].
External Interfaces	The external interfaces library allows you to write C and Fortran programs that interact with MATLAB. It includes facilities for calling routines from MATLAB (dynamic linking), for calling MATLAB as a computational engine, and for reading and writing MAT-files.
Tool Box	MATLAB consist of specialized tool for performing specialized task such as Fuzzy Logic tool box (specialized for working on Fuzzy method system and producing result for

	the analysis) [1, 9].

<u>MATLAB start-up window and the functions of different TABS</u>

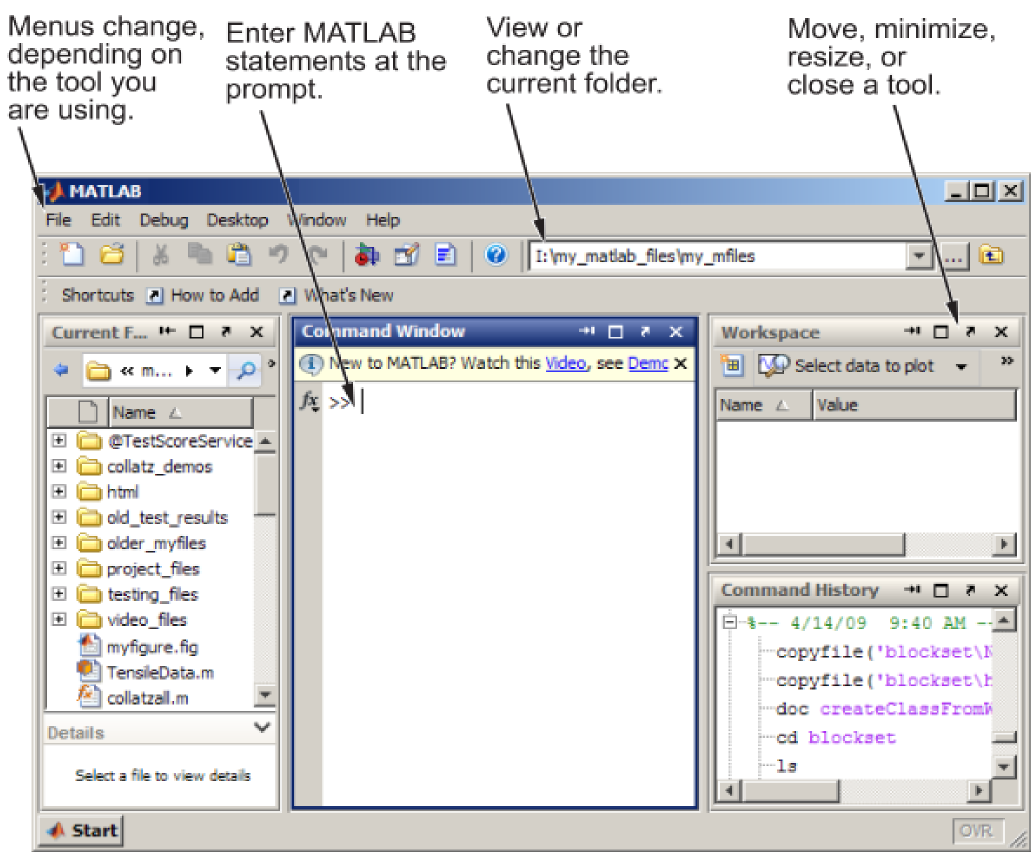


FIG. 2.4 MATLAB WINDOW, [MATLAB 7.0 [1]]

If someone wants to use a specialized tool box like the fuzzy logic tool box he needs to enter "fuzzy" at the prompt.

c) Fuzzy Logic Tool box:

Fuzzy Logic Toolbox™ provides MATLAB [1, 8]® functions, graphical tools, and a Simulink® block for analyzing, designing, and simulating systems based on fuzzy logic. The product guides you through the steps of designing fuzzy inference systems. Functions are provided for many common methods, including fuzzy clustering and adaptive neurofuzzy learning.

The toolbox lets you model complex system behaviors using simple logic rules, and then implements these rules in a fuzzy inference system (FIS). One can use it as a stand-alone fuzzy inference engine. Alternatively, one can use fuzzy inference blocks in Simulink and simulate the fuzzy systems within a comprehensive model of the entire dynamic system [1, 12].

Key features of the fuzzy logic tool box in Matlab are,

- Specialized GUIs for building fuzzy inference systems and viewing and analyzing results.
- Membership functions for creating fuzzy inference systems.
- Support for AND, OR, and NOT logic in user-defined rules.
- Standard Mamdani and Sugeno-type fuzzy inference systems.
- Automated membership function shaping through neuroadaptive and fuzzy clustering learning techniques.
- Ability to embed a fuzzy inference system in a Simulink model.
- Ability to generate embeddable C code or stand-alone executable fuzzy inference engines. [1, 12, 17]

Applications of MATLAB fuzzy logic tool box:

- One can create and edit fuzzy inference systems (FIS) with Fuzzy Logic Toolbox software.
- One can create these systems using graphical tools or command-line funtions, or can generate them automatically using either clustering or adaptive neuro-fuzzy techniques.
- If one has access to Simulink software, one can easily test your fuzzy system in a block diagram simulation environment.
- The toolbox also lets you run your own stand-alone C programs directly.

- One can define the rules in the rule editor and test the result output in the surface viewer.
- There is an option of selecting either of the two fuzzy methods Mamdani [14] or Sugeno [15].

Fuzzy logic Toolbox Graphical user interface tools:

There are five different tools in the Fuzzy logic toolbox, which are used to build, edit and view fuzzy inference system [8, 12, 17] (Table 2.3).

Table 2.3 MATLAB fuzzy logic toolbox GUI tools [1, 18]

Tool	Description
Fuzzy Inference System (FIS) Editor	This tool is to handle the high-level issues for the system—How many input and output variables? What are their names? Fuzzy Logic Toolbox software does not limit the number of inputs. However, the number of inputs may be limited by the available memory of your machine. If the number of inputs is too large, or the number of membership functions is too big, then it may also be difficult to analyze the FIS using the other GUI tools.
Membership Function Editor	This tool is used for defining the shapes of all the membership functions associated with each variable which can be changed as per the input.
Rule Editor	This tool is used for editing the list of rules that defines the behavior of the system.
Rule Viewer	This tool is used for viewing the fuzzy inference diagram. One can use this viewer as a diagnostic to see, for example, which rules are active or how individual membership function shapes influence the results and can be

	changed from here if required.
Surface Viewer	This tool is used for viewing the dependency of one of the outputs on any one or two of the inputs—that is, it generates and plots an output surface map for the system. These GUIs are dynamically linked, in that changes you make to the FIS using one of them, affect what you see on any of the other open GUIs. For example, if you change the names of the membership functions in the Membership Function Editor, the changes are reflected in the rules shown in the Rule Editor. You can use the GUIs to read and write variables both to the MATLAB workspace and to a file (the read-only viewers can still exchange plots with the workspace and save them to a file). You can have any or all of them open for any given system or have multiple editors open for any number of FIS systems.

Here are the images of the different tools of Fuzzy logic tool box in MATLAB [6] (Fig. 2.5)

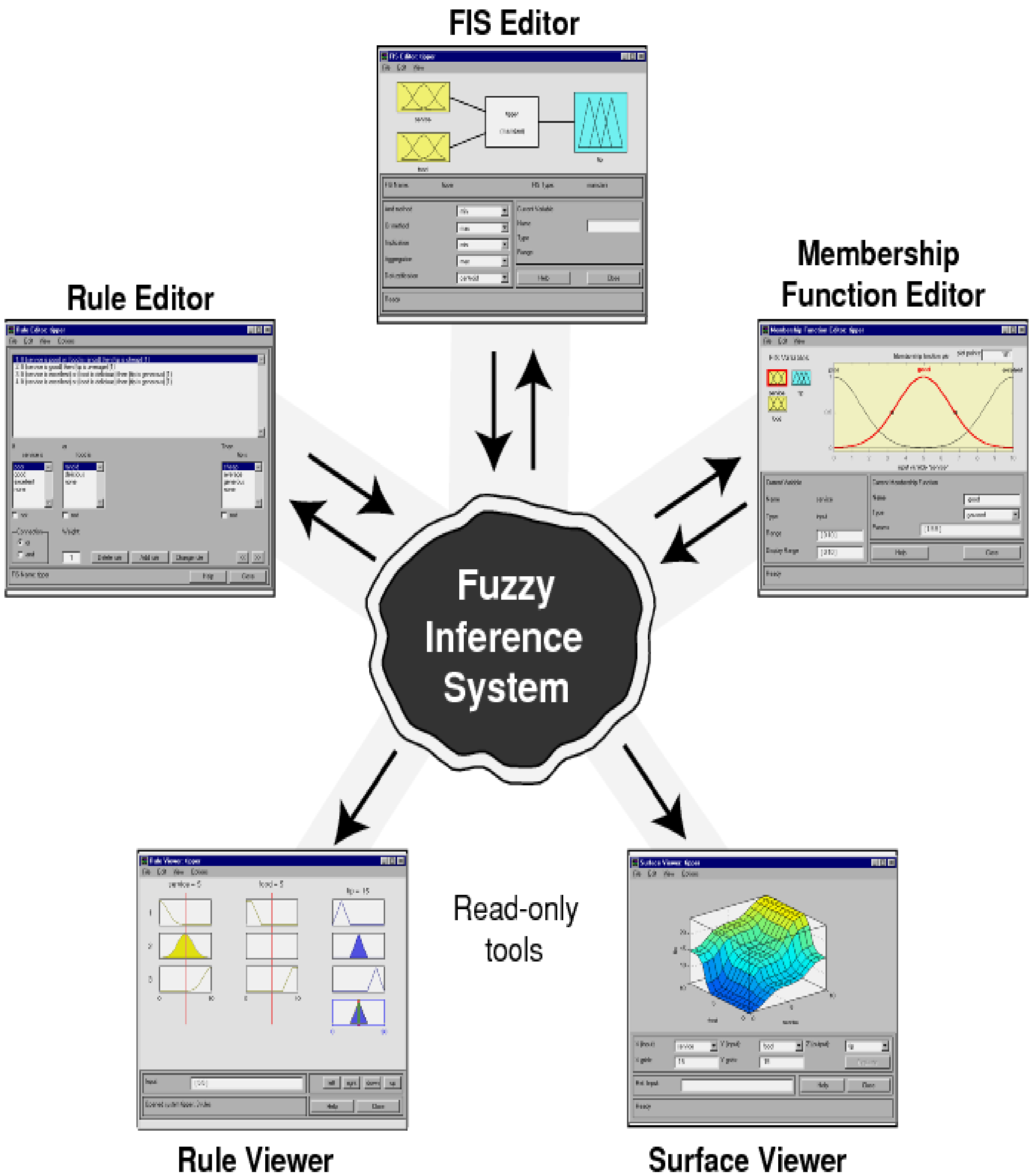


FIG. 2.5 MATLAB FUZZY LOGIC TOOL BOX GUI TOOLS [1, 12, 18]

Chapter Three: Proposed Scheme

3.1 Objective

The proposed innovative approach is to reduce the power consumption of the device when it is not originating, receiving or relaying any traffic through the network or the user devices. With this innovative approach we can increase the life time of the energy-constrained nodes and the end user devices considerably.

The proposed approach involves the monitoring of data (packets) through the network or the user device and using fuzzy logic to define the rules for obtaining actionable silent period (ASP1 and ASP). If we monitor the network or the user device, we can see that there is an irregular pattern in which the packets move depending on the time and work load during a day or certain period of time. In any network and user device there will always be a time frame during which you can see that there is no relaying of traffic but still the connection between the device and the network is at same level as it was when there was heavy traffic moving through the network or the user device.

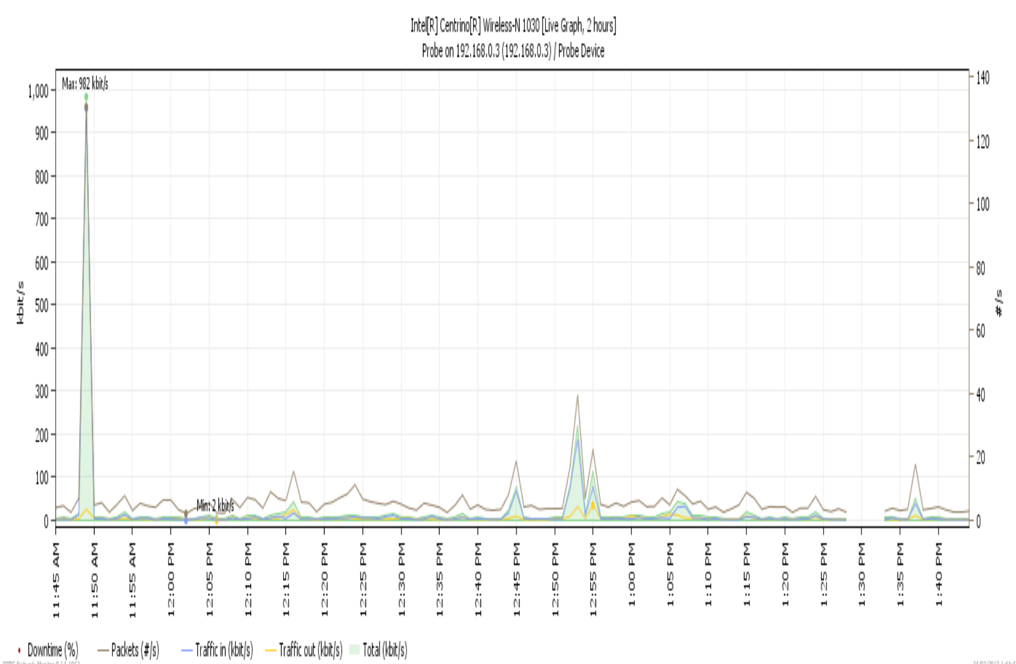


FIG. 3.1 ACTUAL TRAFFIC MONITORED BY PRTG [2] ON A NIC

In Fig. 3.1 and 3.2, we can see an irregular pattern of actual traffic flowing through the Network interface card of one of the laptops used for experiment. From this, we can see that the number of packets most of the time is less than eight packets/sec. These are packets which are just used for updating the address of the devices. Now, if we program and use some logical concepts we can switch these devices to silent mode without changing any current protocol.

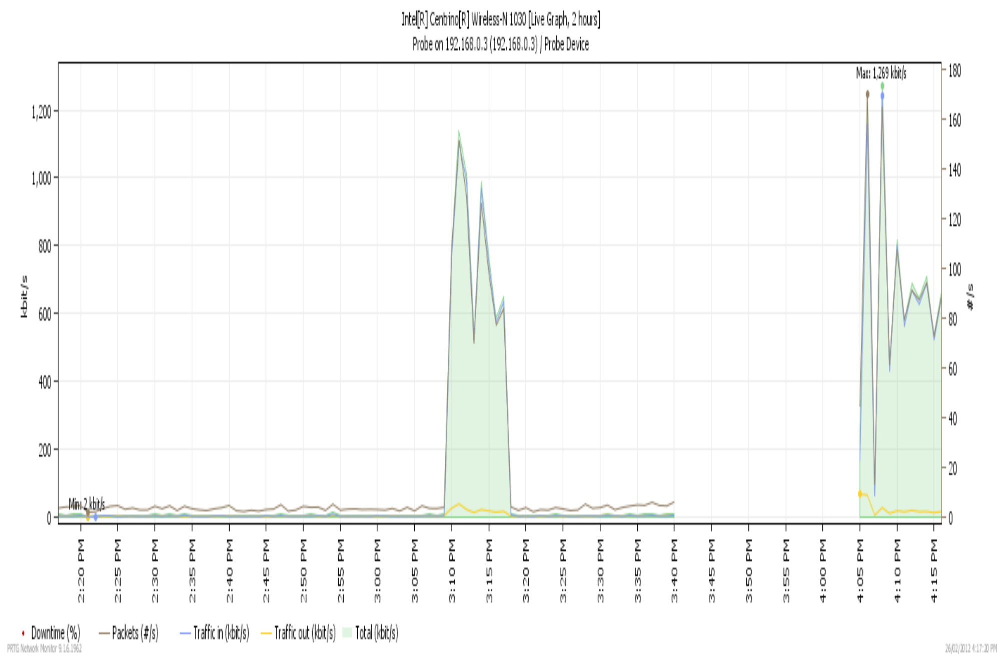


FIG. 3.2 ACTUAL TRAFFIC MONITORED BY PRTG [2]

From the point of view of understanding the concept, consider fig. 3.3 below.

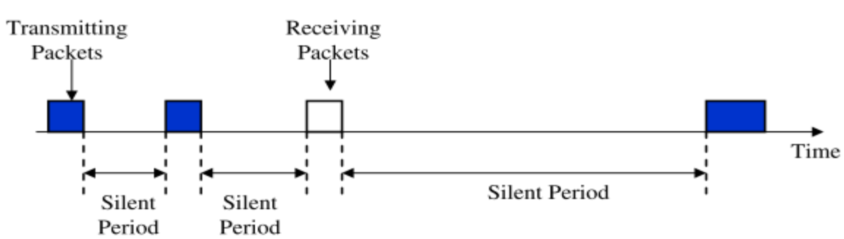


Fig. 3.3 Demonstrated Natural silent period of a network interface [3]

It gives a clear image of the relay of a single packet through a network interface or user device. We can see that when the packet is relayed through the network there is a silent period between the two transmitting packets and some of the time the silent period is large enough for switching the device in the silent mode. Obviously, as we have mentioned above, our goal is to achieve reducing the power consumption of the network and user device without changing any current protocol. Hence, the switching of the network and the user device will be in the way so that the devices have some buffer time to check whether there are some more packets on the way and if there are no packets and the buffer time has been reached than the device can be switched to the silent mode.

However, the issue of switching the network or the user device to silent mode may cause the loss of some packets, when a network or user device is receiving it; but in the case of the network or the device generating traffic (packets), it can switch the network or user device from silent mode to active mode without loss of any packets.

Some of the ideas to compensate the loss of packets which is very minimal compared to the amount of power we save and the increase in the life time of the network and user device are listed below and in the section on future work.

If we are able to predict the silent period correctly and program our network in a way that it will move the network interface and the user device to the silent mode during that time (when the network interface and user device do not expect to receive or transmit any data packets) regularly than we can achieve our goal of reducing the power consumption, increasing the life time of the network interfaces and user devices as well as reducing the packet losses.

Once we have the predicted data from the network and the user device we can set the rules in a way so that during the silent period the node will move from silent mode to complete power down mode leaving just a timer to wake it up once the natural silent period (predicted silent period when used to switch the device in the silent mode is called as natural silent period) is over. This is a very important concept this can save maximum power and if the flow of packets is not high as in the case of a personal network we can save maximum energy which can be nearly half of the amount of the power consumed in normal scenarios.

3.2 Actionable silent Period (ASPs)

From what we have discussed (above) in the objective we can say that the basic idea behind the motivation of the concept of silent networking is that we can monitor the traffic pattern and use this historical data to predict the naturally occurring silent periods for the nodes in the network or the user devices. Each node of the network interface or user device can be switched to the naturally occurring silent period if the silent period is expected to be an actionable silent period (ASPs)

"An ASP is defined as follows,

A silent period in which the interface does not need to originate, receive, or relay packets with a given probability, and satisfies the following conditions;

a) Longer than an ASP Threshold: An ASP threshold is a threshold such that when a silent period is shorter than the threshold, it will no longer make sense to power down a network interface. For example, the ASP threshold can be the sum of the times it takes the mobile device to shut down the interface and to later bring it back up. Essential to silent networking is a real-time ASP prediction method for determining when and for how long an interface should be powered off.
b) The sum of the time periods an interface is powered off should be as long as possible to conserve energy.
c) The incoming packet loss (or delay) caused by powering off the interface should be very small and controllable.
d) Note that the out-going traffic can trigger an interface to wake up if it is not already awake, hence is not likely to be lost." [3]

However, the proposed idea totally depends on the activity of the networks and user devices, but from the research work we can say that there is hardly any firm that has constant heavy traffic throughout any given period of time.

However, more dynamic and better power saving can be achieved using this concept if we are able to make some changes to the protocol. We will discuss this in the future work section.

3.3 ASP Model

Actionable silent period is the core of the proposed approach for saving power and extending the lifetime of the network nodes and user devices. During the initial period of data acquisition “learning phase” of the model it collects the information about the originating, receiving and relaying of the traffic (network activity) through the network nodes and the user devices under consideration [3, 5].

With the monitoring of the data, ASP model constructs a histogram which contains the information about the packets (traffic). This histogram contains the length of the Actionable silent period for each of the network nodes and the user devices; based on the activity of the network nodes and the user devices the ASP switches the network nodes and user device interface to the silent mode and if the ASP threshold is greater than the set value (depends on the administrator) then all the interface is powered off except the timer which switches the nodes and the user device from power down mode to the active mode at the end of the Actionable silent period (ASP 1) [3, 4].

Once the initial histogram is prepared by the ASP model, it keeps on updating itself (Histogram) by continuously monitoring the networked devices and updating the events in its algorithm. In our prototype experiment we have used Paessler monitoring tool PRTG [2] for monitoring the traffic (packets) through the network nodes and user devices.

3.4 Flow chart for the experiment

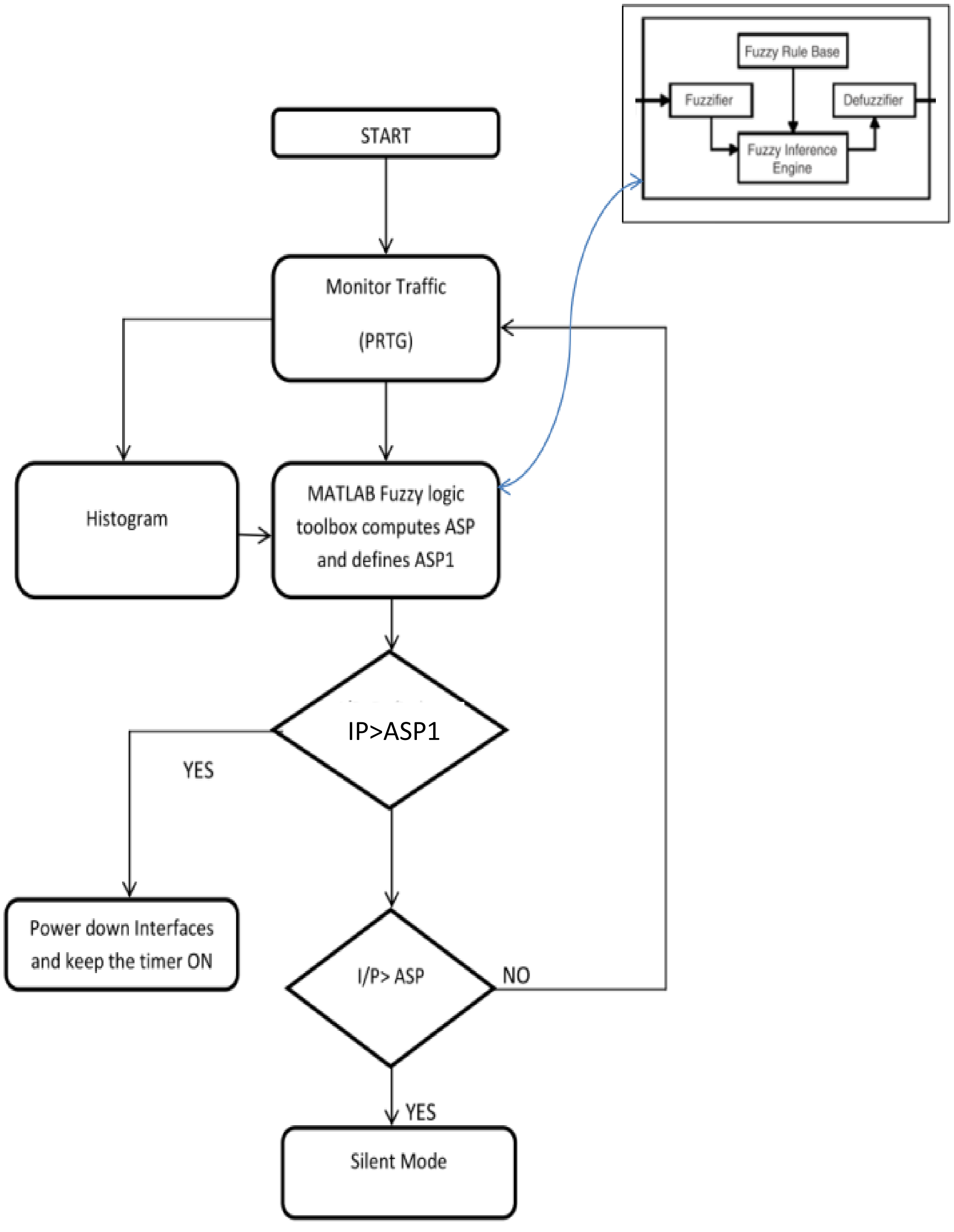


FIG. 3.4 FLOW CHART FOR THE DESIGN OVERVIEW AND FUZZY PROCESS [8]

In the flow chart above, the procedure (flow) and the design of the experiment is mapped. Once, the experiment is started, the flow goes to the Paessler monitoring tool PRTG [2], how PRTG monitors and what kind of information we can receive from it is explained later.

Once we have the data containing the information about the traffic of each node of the network and the user device, it is updated in the histogram (saves the information) and is provided to the Fuzzy logic tool box of MATLAB [1, 12]. Now, MATLAB [1] computes the silence period (input for the next level in the flow chart) by observing the flow of the traffic (activity of the network) and records it in the histogram for further analysis. For further reference, let keep the current input to ‘S’ [3].

Now, once we have the current silent period input S, we compare it to the stored (from his previous observation) or defined actionable silent period ASP [3].

If the comparison of the result is positive i.e. S>ASP1, the network nodes or the devices switches to the power down mode. ASP1 can be a defined value or threshold value.

If the comparison of the result is negative, the flow goes to the next level and it is compared again to the ASP value which will be the recently calculated value and will have a lower threshold value than the ASP1.

If the result of the second comparison is negative the flow returns back for monitoring the network activity. If the result of the comparison is positive, than the network node or the user device is switched in the silent mode.

Chapter Four: Prototype Implementation and Experimental results

4.1 Paessler Monitoring Tool PRTG implementation

As we have discussed above, Paessler is a network monitoring tool commonly known as a Paessler Router Traffic Grapher (PRTG). In this section we will answer the following questions.

a) How does PRTG [2] capture data on a network?

b) What kind of information can it provide?

c) What are the things we need to take care of to make sure that we are obtaining correct data?

d) Where can we see the output?

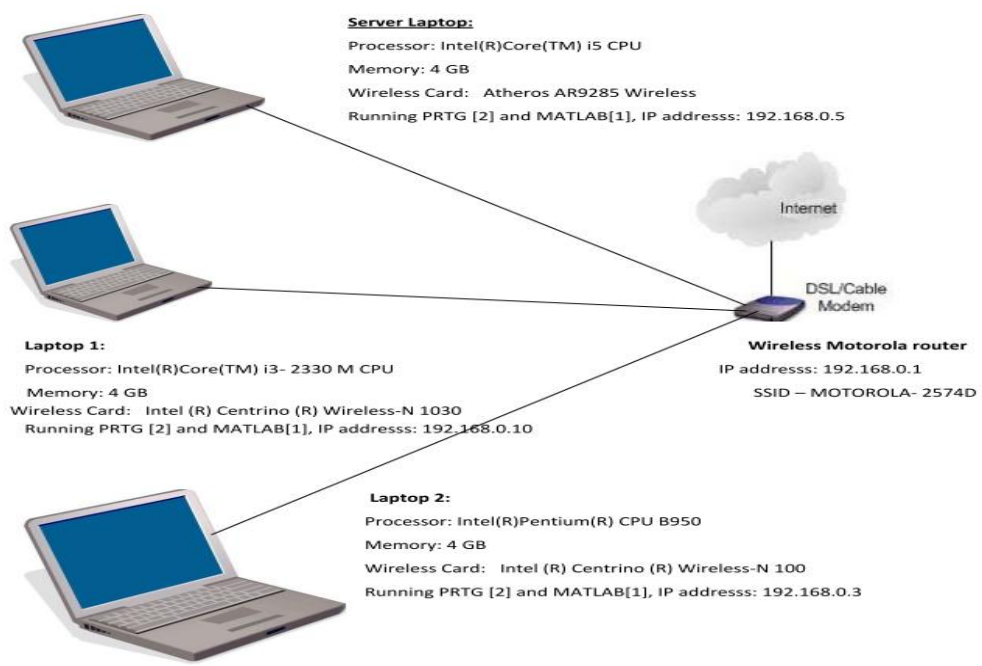


FIG. 4.1 ARCHITECTURE AND EXPERIMENTAL SETUP OVERVIEW

Fig. 4.1 explains us the experimental setup for our experiment. In the experiment we have used three wireless enabled laptop and one wireless router. The configuration of the laptop is given in the figure as well as in the appendix. On the server laptop, PRTG [2] and MATLAB [1] are running. This is the hardware configuration for our design.

Now, from the software end, one needs to have a copy of the PRTG on the server (on the device from where we are going to monitor the network activity). In the PRTG Server Administrator program one can define different system-oriented settings that affect the PRTG installation, as well as restart services and view log information. With the installation of the PRTG on the server complete it will look like the figure below.

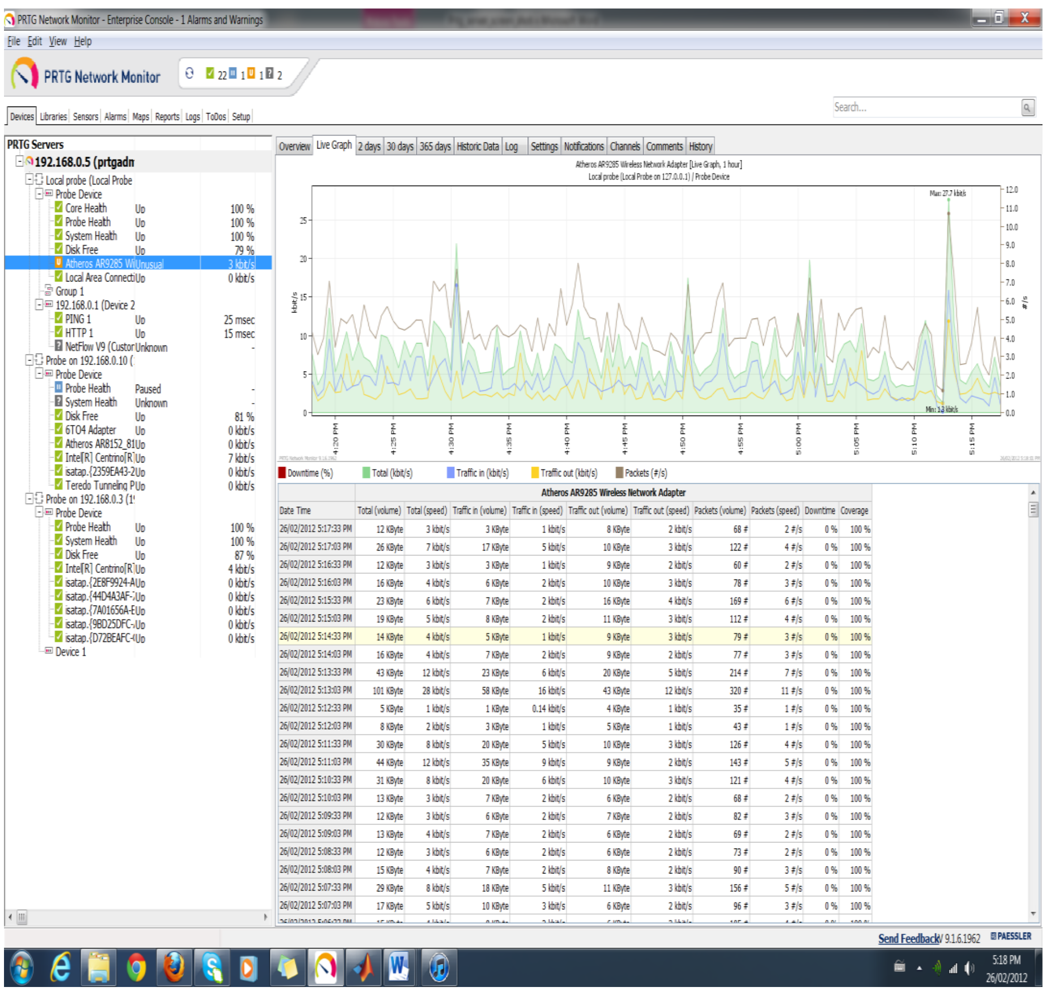


FIG. 4.2 PRTG [2] ENTERPRISE CONSOLE, SERVER

From the server window one can find all the information that is required while monitoring. One can get live coverage of the data traffic, and a recorded coverage of traffic up to a period of one year.

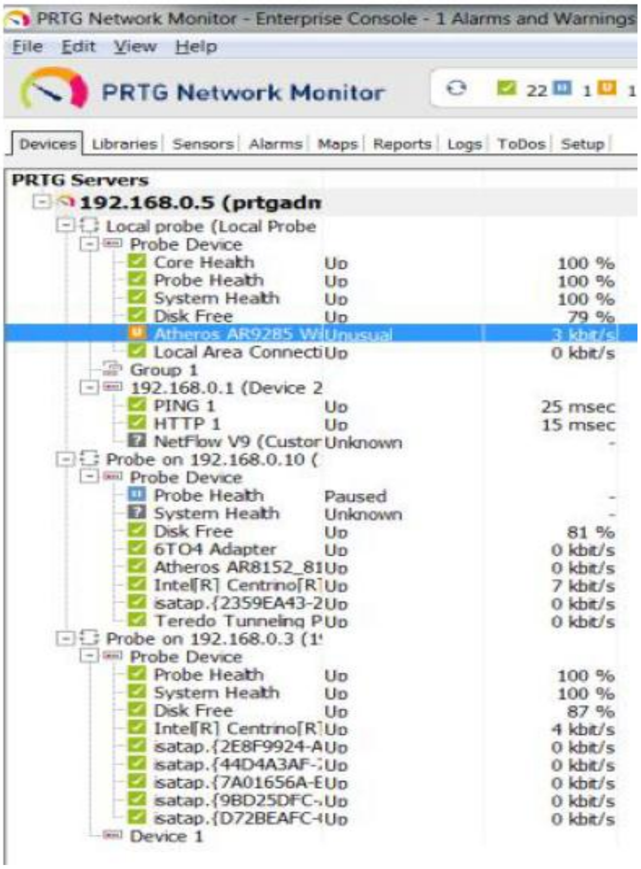


FIG. 4.3 SERVER SECTION INFORMATION ABOUT DEVICES [2]

Fig. 4.2 shows information about the network activity of different network nodes and user devices. A user can set different parameters he/she wants to monitor.

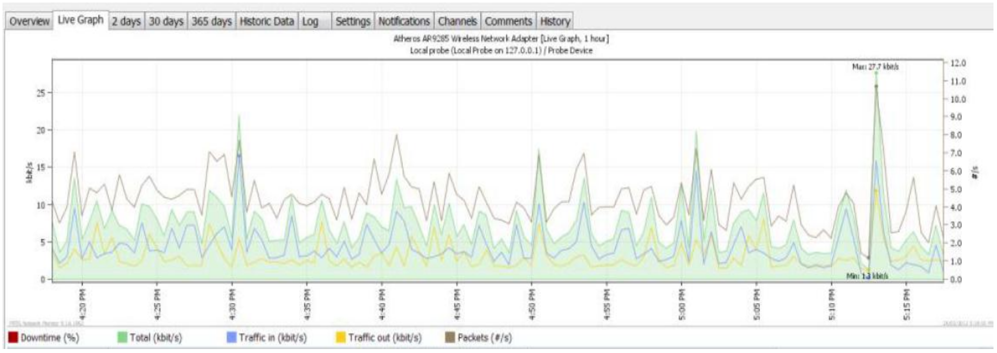


FIG. 4.4 SERVER SECTION INFORMATION CONTAINING TRAFFIC GRAPHS [2]

In this area (fig. 4.4) of the server console one can view the different traffic pattern for individual network nodes and user devices.

	Atheros AR9285 Wireless Network Adapter									
Date Time	Total (volume)	Total (speed)	Traffic in (volume)	Traffic in (speed)	Traffic out (volume)	Traffic out (speed)	Packets (volume)	Packets (speed)	Downtime	Coverage
26/02/2012 5:17:33 PM	12 KByte	3 kbit/s	3 KByte	1 kbit/s	8 KByte	2 kbit/s	68 #	2 #/s	0 %	100 %
26/02/2012 5:17:03 PM	26 KByte	7 kbit/s	17 KByte	5 kbit/s	10 KByte	3 kbit/s	122 #	4 #/s	0 %	100 %
26/02/2012 5:16:33 PM	12 KByte	3 kbit/s	3 KByte	1 kbit/s	9 KByte	2 kbit/s	60 #	2 #/s	0 %	100 %
26/02/2012 5:16:03 PM	16 KByte	4 kbit/s	6 KByte	2 kbit/s	10 KByte	3 kbit/s	78 #	3 #/s	0 %	100 %
26/02/2012 5:15:33 PM	23 KByte	6 kbit/s	7 KByte	2 kbit/s	16 KByte	4 kbit/s	169 #	6 #/s	0 %	100 %
26/02/2012 5:15:03 PM	19 KByte	5 kbit/s	8 KByte	2 kbit/s	11 KByte	3 kbit/s	112 #	4 #/s	0 %	100 %
26/02/2012 5:14:33 PM	14 KByte	4 kbit/s	5 KByte	1 kbit/s	9 KByte	3 kbit/s	79 #	3 #/s	0 %	100 %
26/02/2012 5:14:03 PM	16 KByte	4 kbit/s	7 KByte	2 kbit/s	9 KByte	2 kbit/s	77 #	3 #/s	0 %	100 %
26/02/2012 5:13:33 PM	43 KByte	12 kbit/s	23 KByte	6 kbit/s	20 KByte	5 kbit/s	214 #	7 #/s	0 %	100 %
26/02/2012 5:13:03 PM	101 KByte	28 kbit/s	58 KByte	16 kbit/s	43 KByte	12 kbit/s	320 #	11 #/s	0 %	100 %
26/02/2012 5:12:33 PM	5 KByte	1 kbit/s	1 KByte	0.14 kbit/s	4 KByte	1 kbit/s	35 #	1 #/s	0 %	100 %
26/02/2012 5:12:03 PM	8 KByte	2 kbit/s	3 KByte	1 kbit/s	5 KByte	1 kbit/s	43 #	1 #/s	0 %	100 %
26/02/2012 5:11:33 PM	30 KByte	8 kbit/s	20 KByte	5 kbit/s	10 KByte	3 kbit/s	126 #	4 #/s	0 %	100 %
26/02/2012 5:11:03 PM	44 KByte	12 kbit/s	35 KByte	9 kbit/s	9 KByte	2 kbit/s	143 #	5 #/s	0 %	100 %
26/02/2012 5:10:33 PM	31 KByte	8 kbit/s	20 KByte	6 kbit/s	10 KByte	3 kbit/s	121 #	4 #/s	0 %	100 %
26/02/2012 5:10:03 PM	13 KByte	3 kbit/s	7 KByte	2 kbit/s	6 KByte	2 kbit/s	68 #	2 #/s	0 %	100 %
26/02/2012 5:09:33 PM	12 KByte	3 kbit/s	6 KByte	2 kbit/s	7 KByte	2 kbit/s	82 #	3 #/s	0 %	100 %
26/02/2012 5:09:03 PM	13 KByte	4 kbit/s	7 KByte	2 kbit/s	6 KByte	2 kbit/s	69 #	2 #/s	0 %	100 %
26/02/2012 5:08:33 PM	12 KByte	3 kbit/s	6 KByte	2 kbit/s	6 KByte	2 kbit/s	71 #	2 #/s	0 %	100 %

FIG. 4.5 SERVER SECTION INFORMATION ABOUT TRAFFIC [2]

From Fig 4.5 one can find out how many packets are following or what is the speed of the network at a particular instance in numerical format.

All the information can also be accessed from the website,

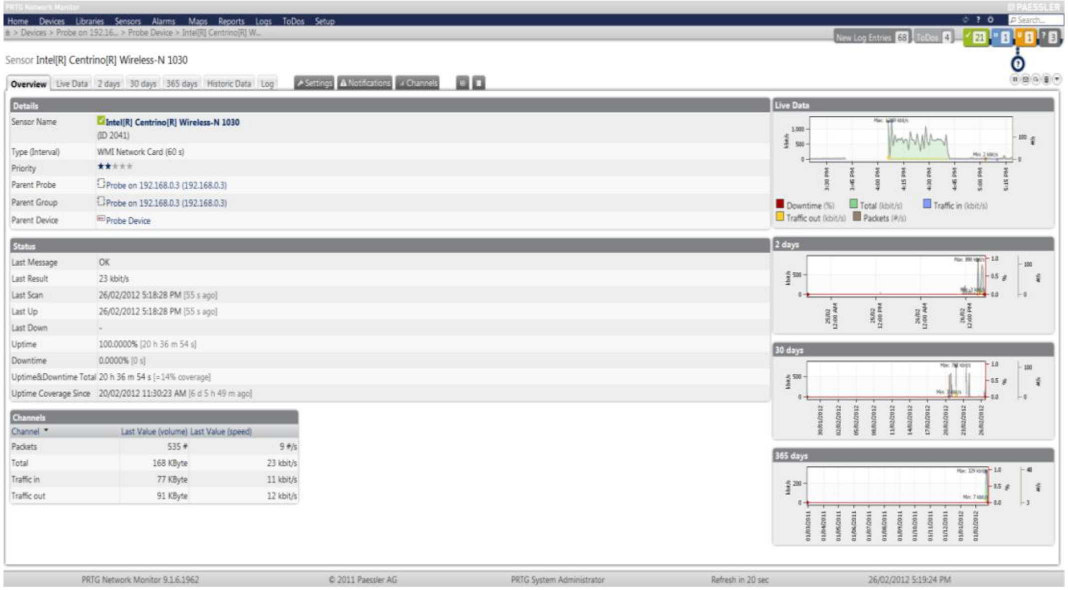


FIG. 4.6 INFORMATION ABOUT THE DEVICE ON PRTG WEBSITE

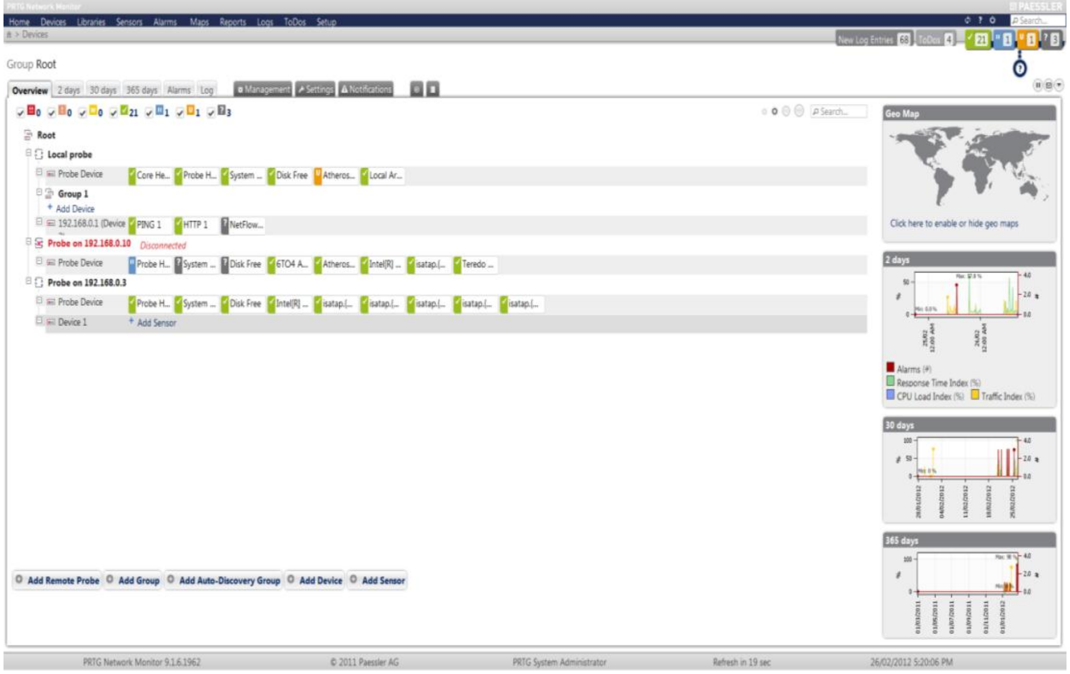


FIG. 4.7 INFORMATION ABOUT THE PARAMETERS FOR WHICH DEVICE IS MONITORED

From Fig. 4.7, we can see the network nodes and the user devices that are considered for monitoring. One can monitor various parameters like health of the device; disk free, and traffic moving through the user device.

Below are some of the parameters one can monitor,

- Common Sensors
- Bandwidth Monitoring Sensors
- Web Servers (HTTP) Sensors
- SNMP Sensors
- Windows/WMI Sensors
- Linux/Unix/OS X Sensors
- Virtual Servers Sensors
- Mail Servers Sensors
- SQL Database Servers Sensors
- File Servers Sensors
- Various Servers Sensors
- VoIP and QoS Sensors
- Hardware Sensors
- Custom Sensors

Above are the group of sensors; each group can have many different types of sensors within itself. Consider the case of SNMP sensors, in this group there are various types of sensors depending on the device you want to monitor [2].

- SNMP APC Hardware Sensor
- SNMP Custom Sensor
- SNMP Custom String Sensor
- SNMP Dell Hardware Sensor
- SNMP HP LaserJet Hardware Sensor
- SNMP Library Sensor
- SNMP Linux Disk Free Sensor

- SNMP Linux Load Average Sensor
- SNMP Linux Meminfo Sensor
- SNMP System Uptime Sensor
- SNMP Traffic Sensor
- SNMP Trap Receiver [2]

Remote Probe:

Remote probe is used to monitor and sharing the work load from the system where core installation has been done. Remote probe is also installed to extend the monitoring of distributed network. In my experiment we have installed remote probe on all the device we are monitoring from the server. [2]

4.2 Matlab Fuzzy logic tool box implementation

On the MATLAB [1] prompt when we provide input as fuzzy, the fuzzy logic tool box FIS editor opens.

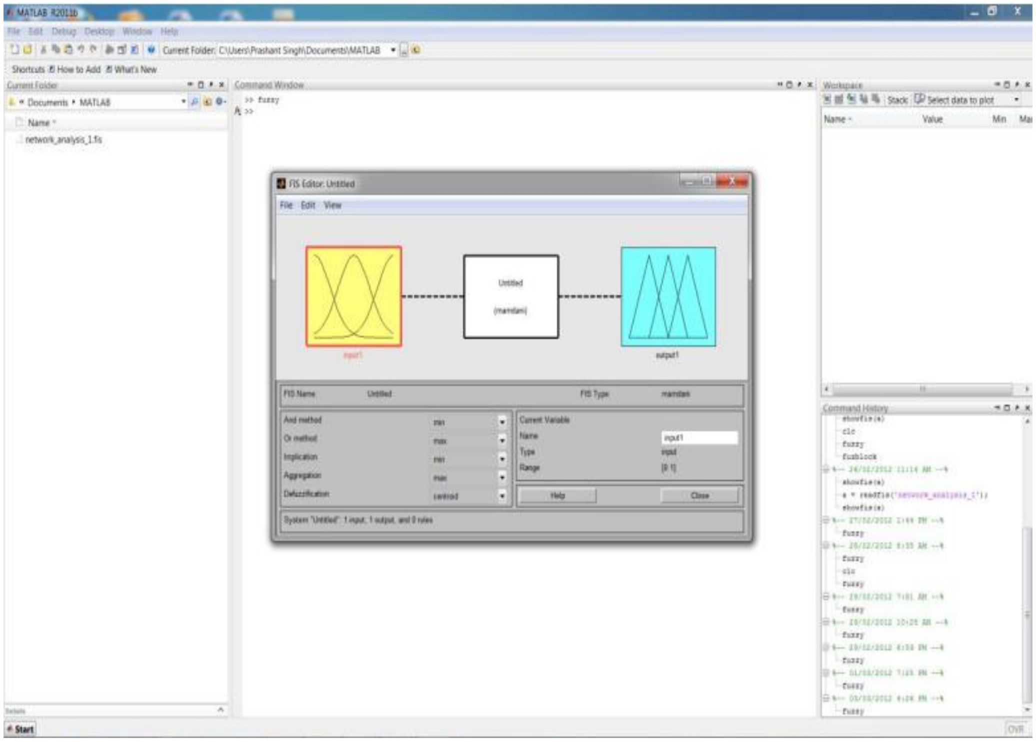


FIG. 4.8 MATLAB FUZZY LOGIC FIS EDITOR

In the Fuzzy logic FIS editor we have to initialize the number of inputs, the fuzzy type, and the number of outputs.

In our experiment, we have considered three laptops (including one server laptop) and one Motorola wireless router. Based on this we have considered five inputs and two outputs. We are also considering Mamdani fuzzy logic as the fuzzy rule generator (As it is widely used). Out the two outputs, one explains the nature of the output packets and the other provides information about the power behaviour (how much power do we save).

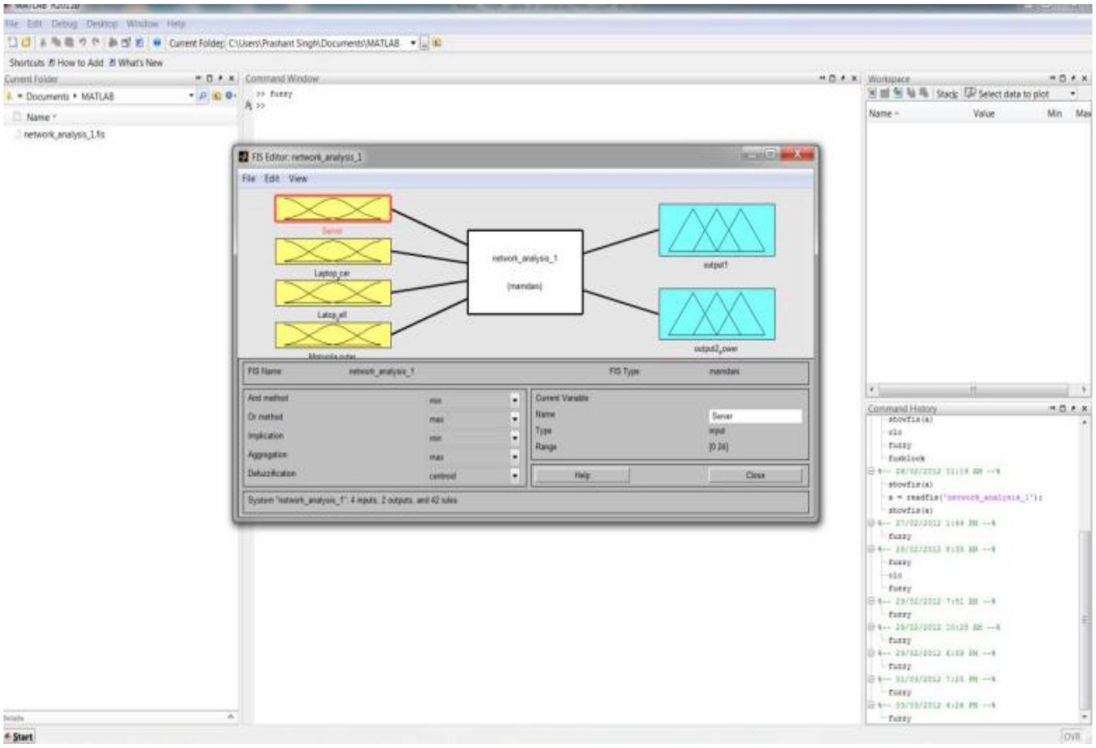


Fig. 4.9 Fuzzy Logic tool box with input and output

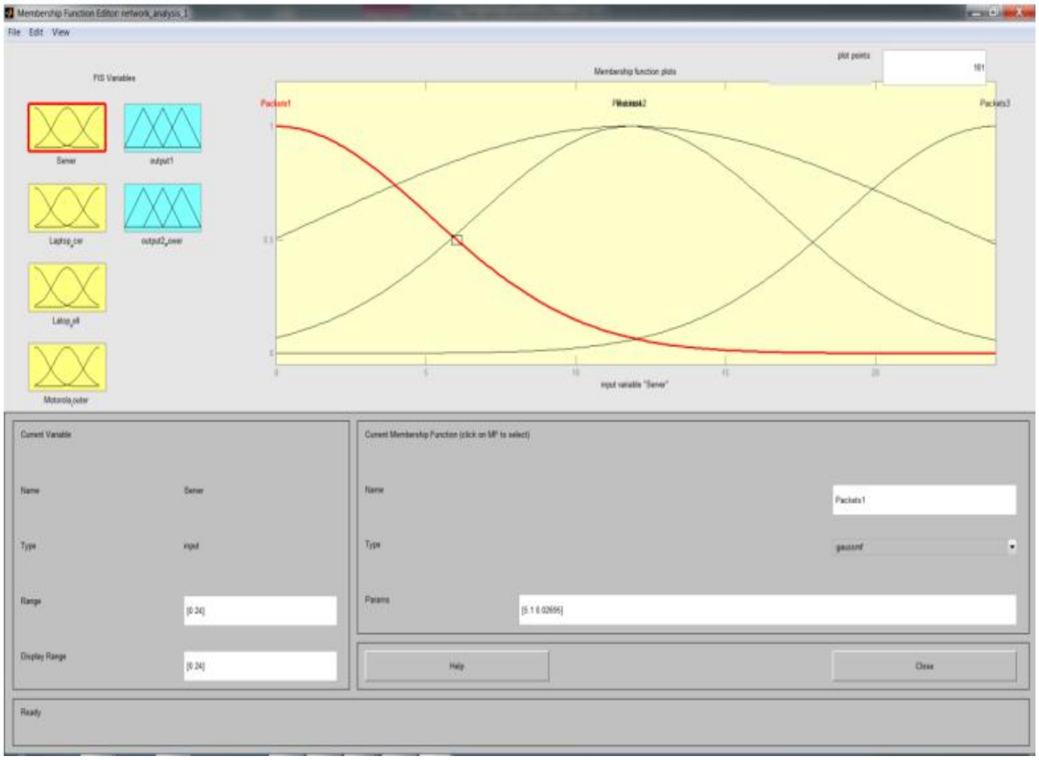


Fig. 4.10 Fuzzy Logic tool Membership Editor

In fuzzy logic tool Membership editor we define the nature of the input for each device as well as the range (our range is of 24 hours).

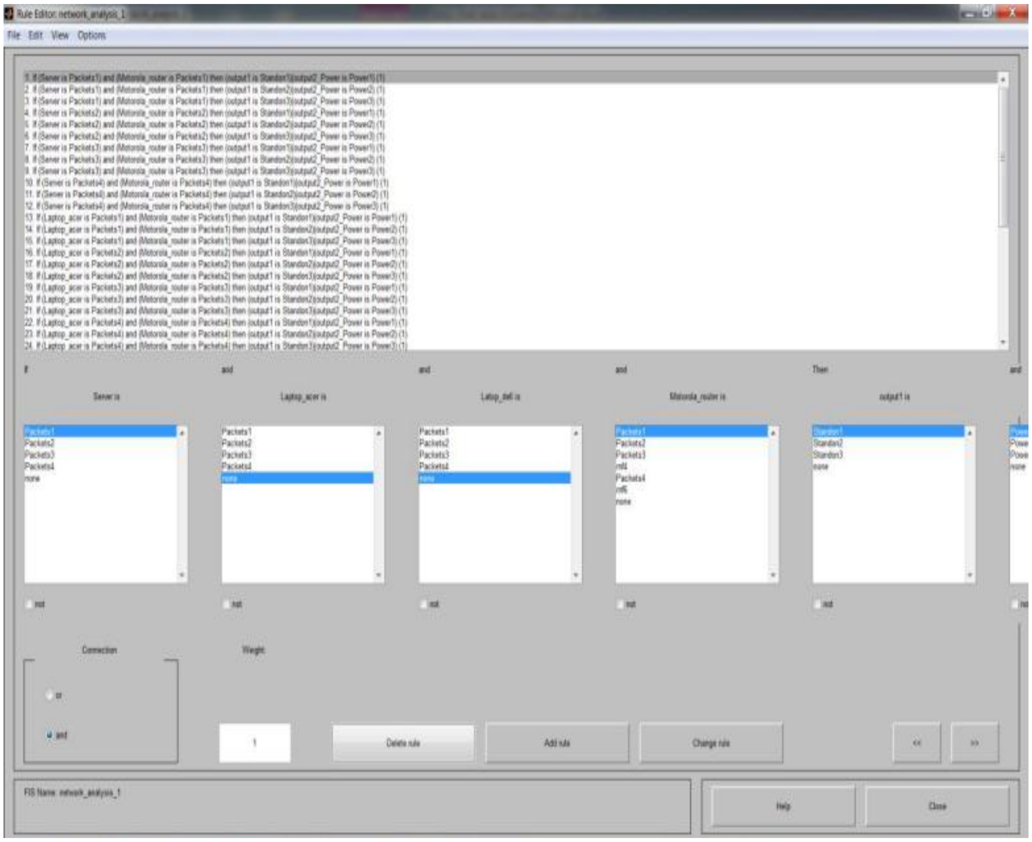


FIG. 4.11 FUZZY LOGIC TOOL RULER EDITOR

Fig. 4.11 shows the fuzzy logic tool box editor [1, 12]. Here, we define rules for our system. Using this rule we define inputs from the membership editor function, and the Mamdani fuzzy logic generates the output for the analysis.

Sample rules for laptop 1 in the experiment

- If (laptop_1) or (packet_1) and (R1) or (R2) output power
- If (laptop_1) or (packet_2) and (R1) or (R2) output power
- If (laptop_1) or (packet_3) and (R1) or (R2) output power
- If (laptop_1) or (packet_4) and (R1) or (R2) output power
- If (laptop_1) or (packet_1) and (R1) or (R2) output traffic
- If (laptop_1) or (packet_2) and (R1) or (R2) output traffic

- If (laptop_1) or (packet_3) and (R1) or (R2) output traffic
- If (laptop_1) or (packet_4) and (R1) or (R2) output traffic
- The same pattern is followed for all the devices in the experiment.
- However, if one want to change certain rules or add some more rules it can be easily done and this is one of the advantages of fuzzy logic.

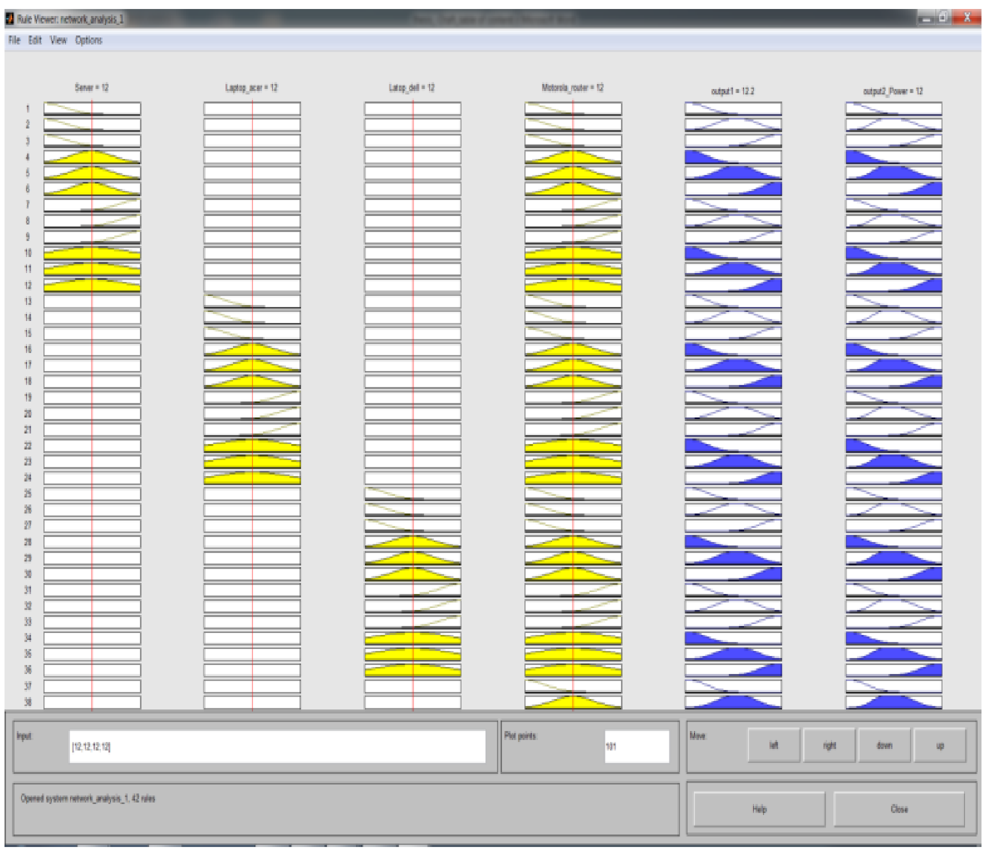


FIG. 4.12 FUZZY LOGIC RULE VIEWER FOR INDIVIDUAL NODE AND USER DEVICE

Fig. 4.12 of the fuzzy logic tool box in Matlab [1] is the rule viewer [12]; here one can make changes for individual devices at once rather changing and editing each rule from Rule editor.

4.3 Paessler Monitoring tool result (output)

In this section we will be analyzing the output of the Paessler Monitor tool. We have monitored the traffic for three laptops and one router over a period of a day.

1. PRTG Graph Server: Laptop 1

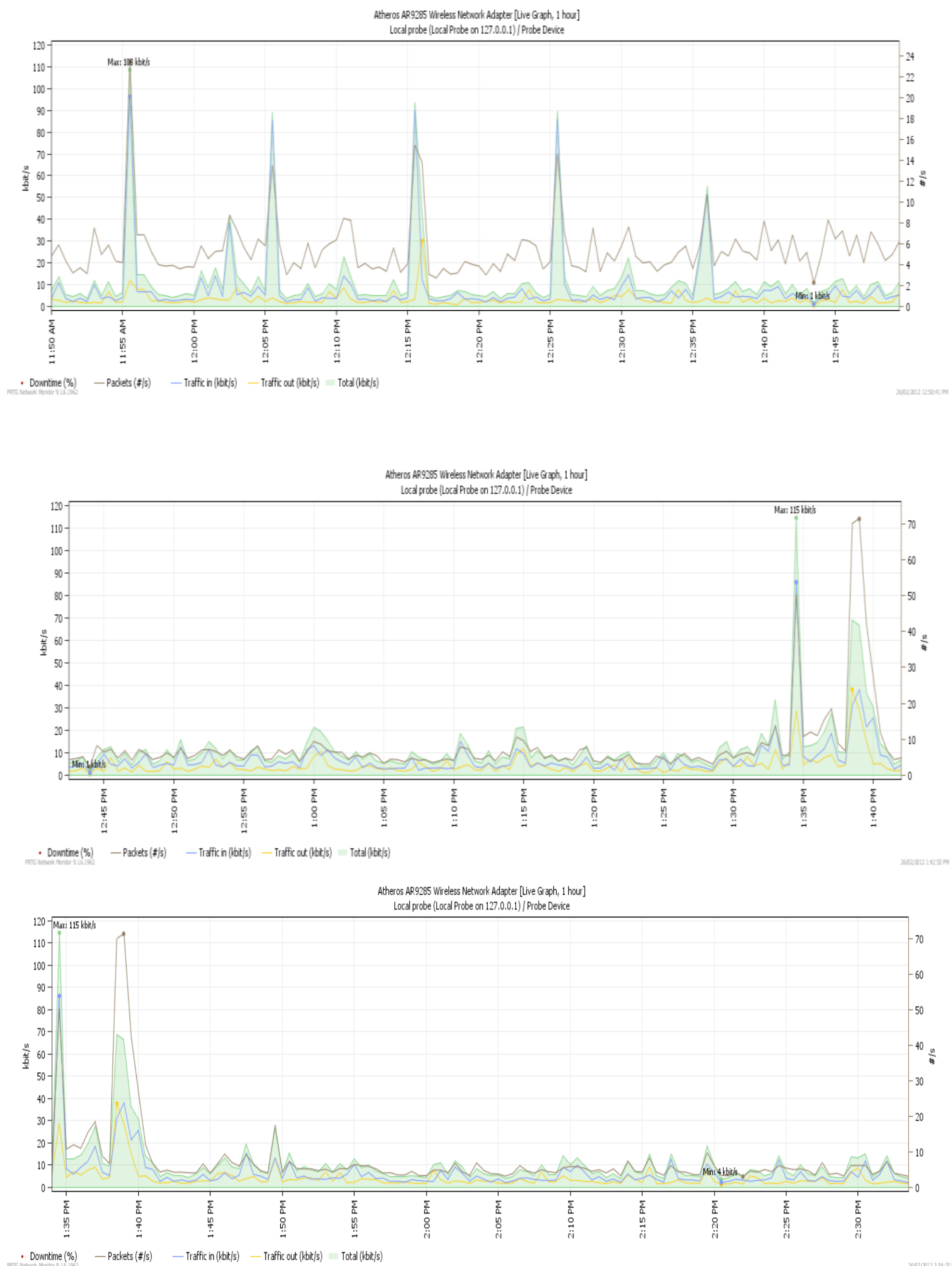


FIG. 4.13 NETWORK ACTIVITY GRAPH OF LAPTOP 1

2. Network activity of Laptop 2:

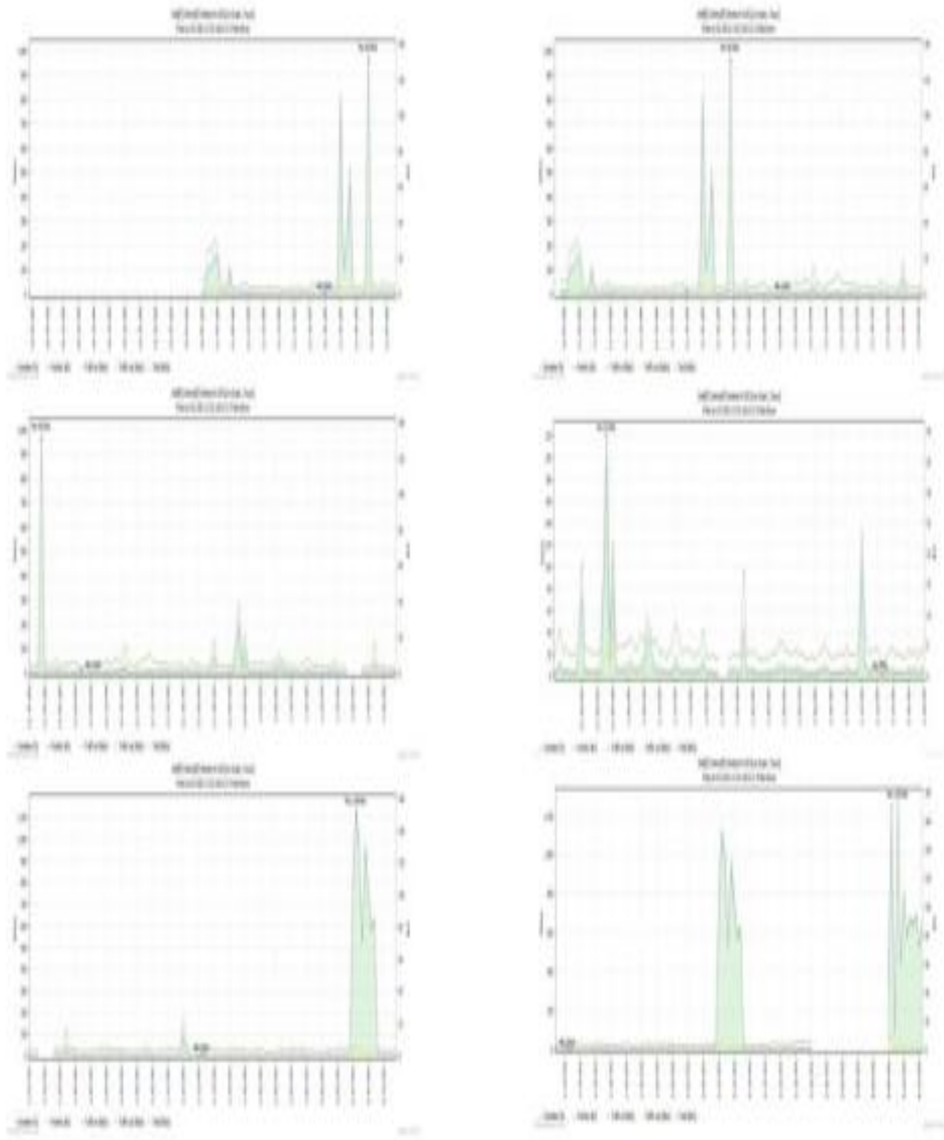


FIG. 4.14 NETWORK ACTIVITY GRAPH OF LAPTOP 2

3. **Network Activity of Router:**

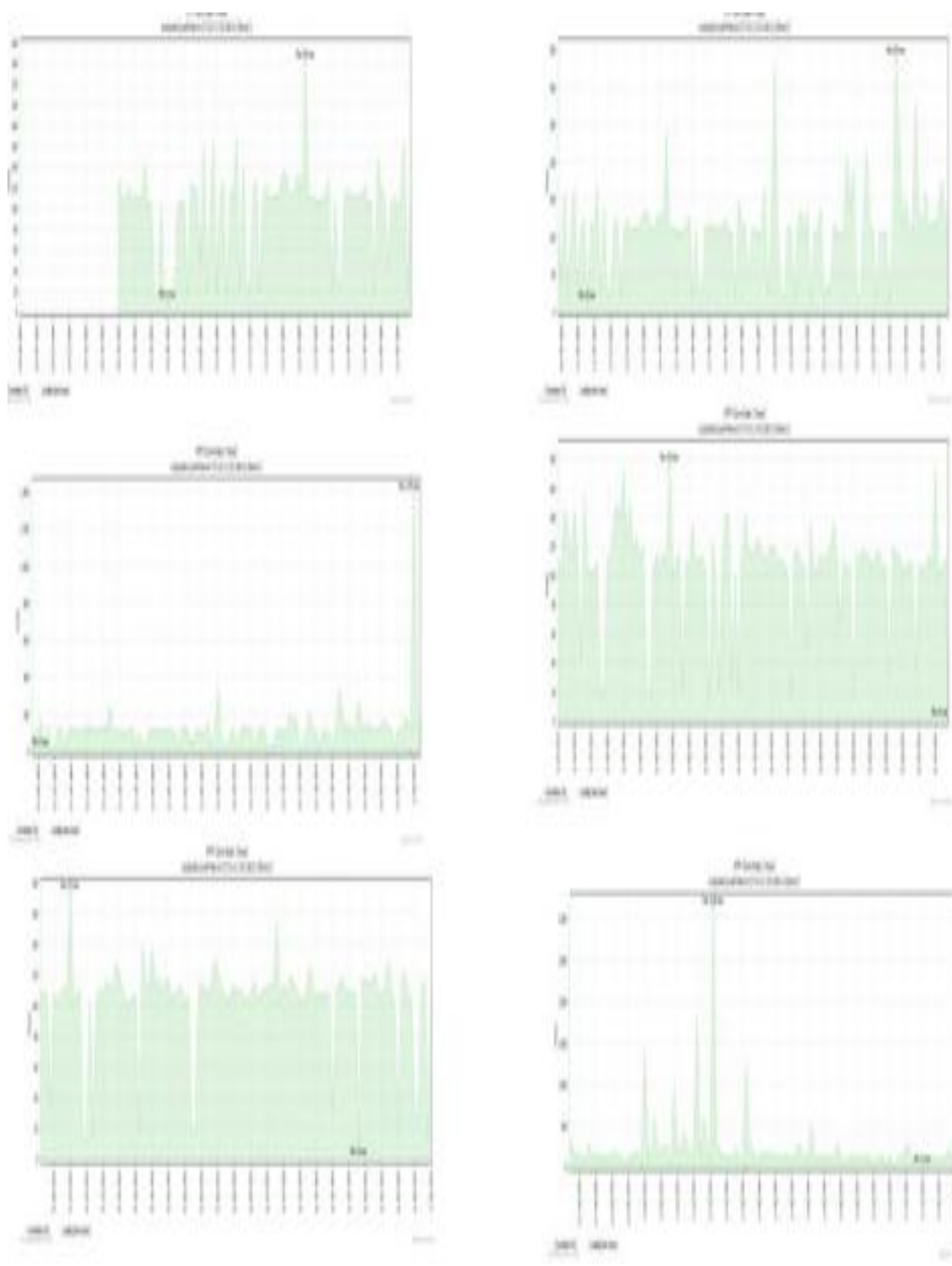


FIG. 4.15 NETWORK ACTIVITY GRAPH OF ROUTER

4. Network Activity of Laptop 3:

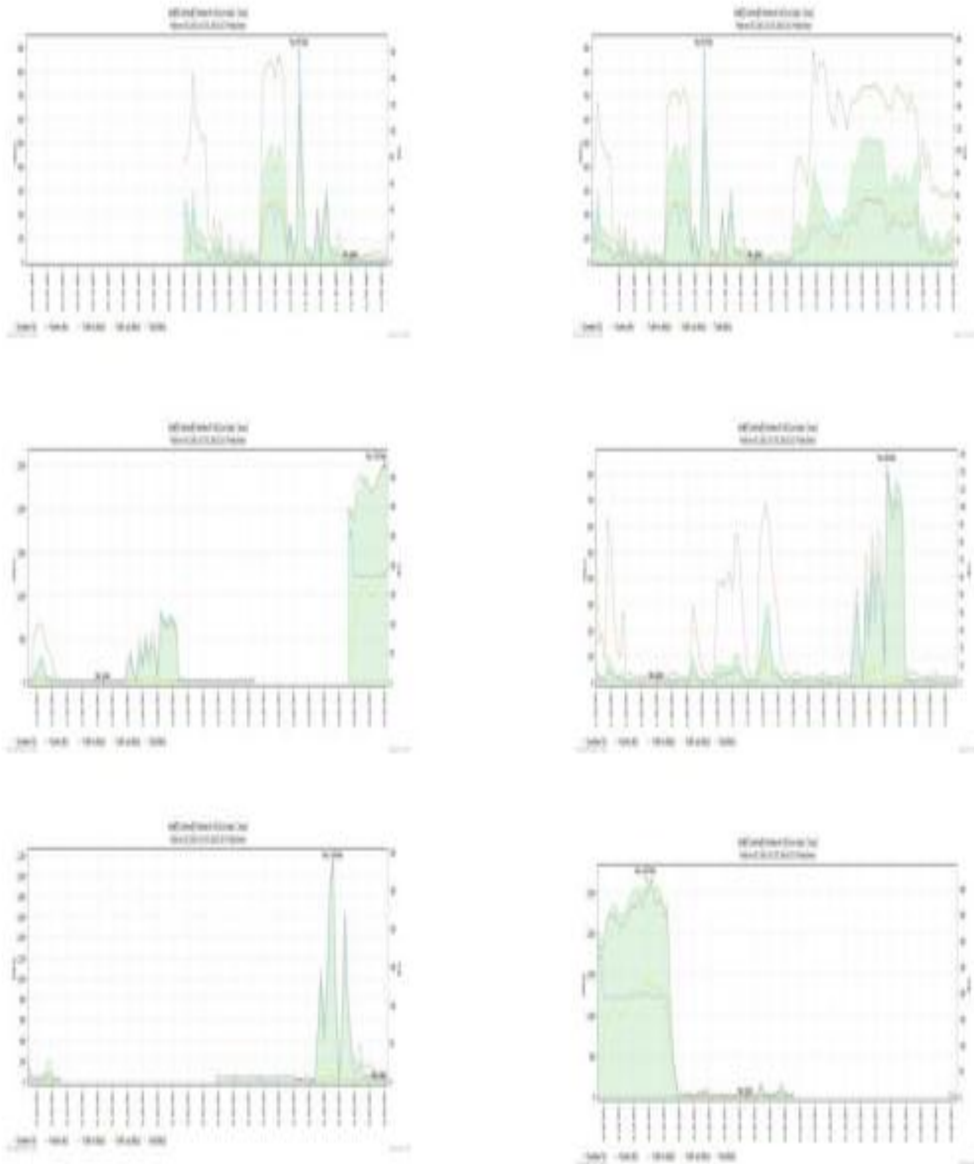


FIG. 4.16 NETWORK ACTIVITY GRAPH OF LAPTOP 3

4.4 MATLAB Fuzzy Logic Tool Box Result

a) **Network Activity and Power flow output of Server:**

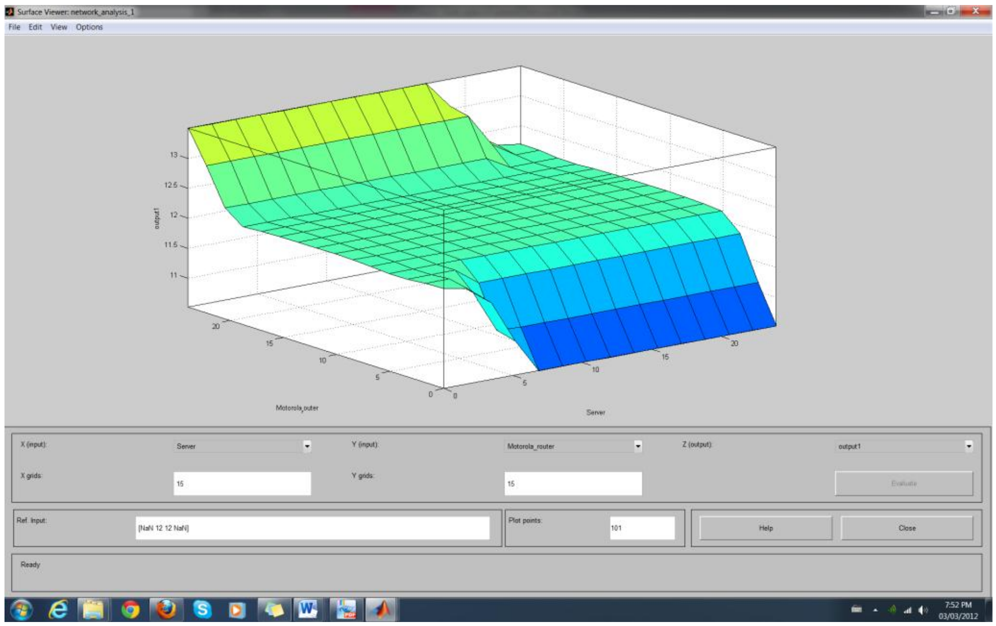


FIG. 4.17.1 NETWORK ACTIVITY (TRAFFIC AT SERVER)

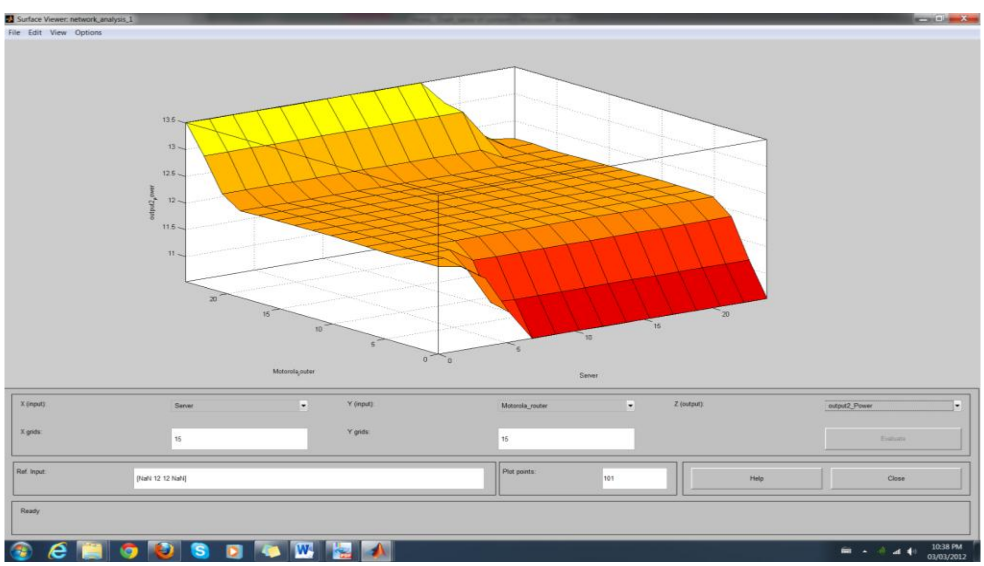


FIG. 4.17.2 POWER O/P DEPENDING ON SEVER AND ROUTER

b) Network Activity and Power flow output of laptop 1:

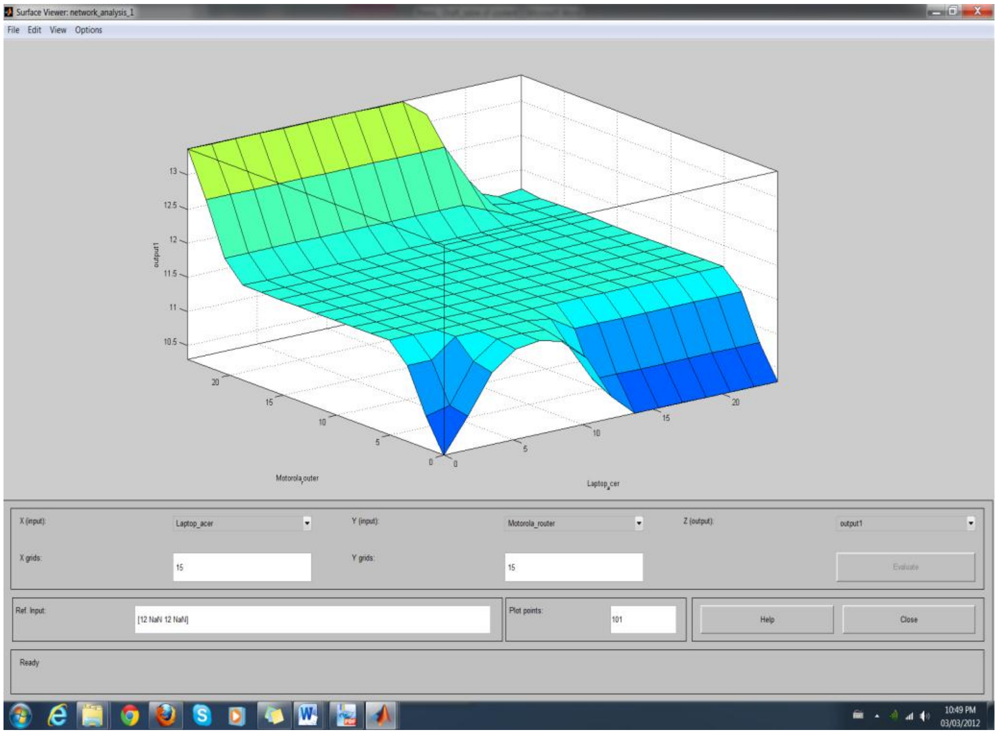


FIG. 4.18.1 NETWORK ACTIVITY AT LAPTOP1 (TRAFFIC AT LAPTOP 1)

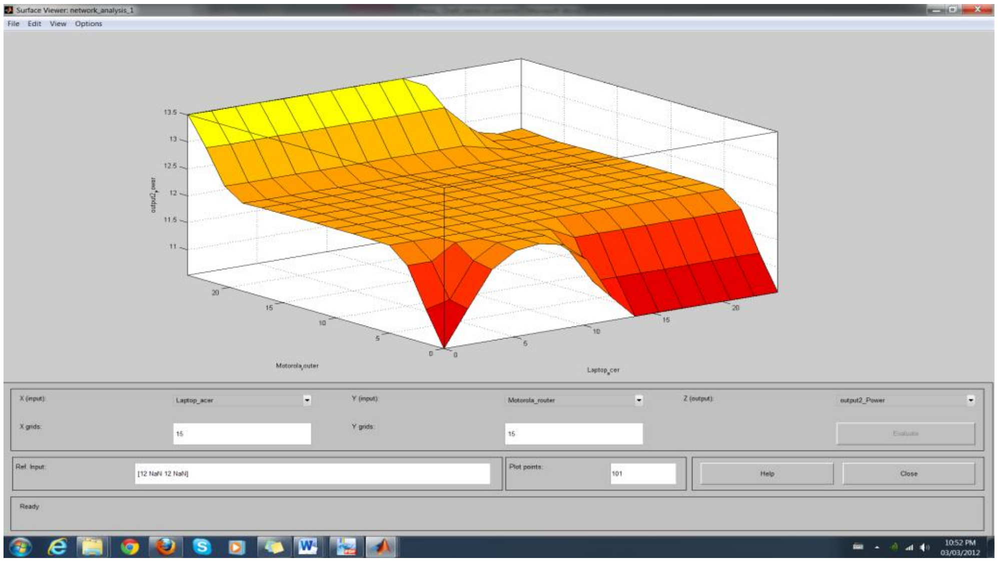


FIG. 4.18.2 POWER O/P DEPENDING ON LAPTOP 1 AND ROUTER

c) Network Activity and Power flow output of laptop 3:

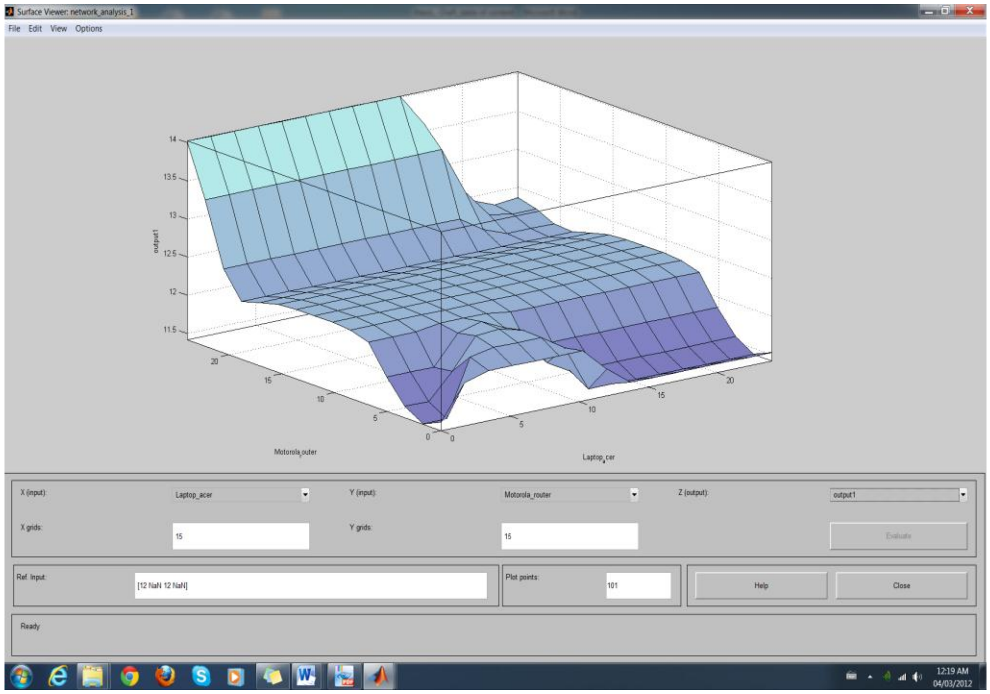


FIG. 4.19.1 NETWORK ACTIVITY AT LAPTOP2 (TRAFFIC AT LAPTOP 2)

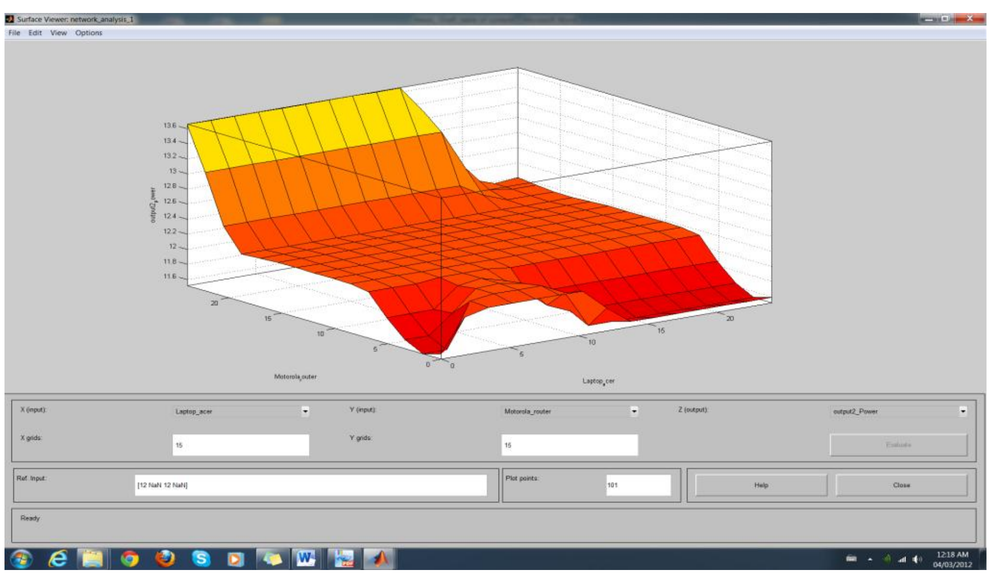


FIG. 4.19.2 POWER O/P DEPENDING ON LAPTOP 2 AND ROUTER

4.5 Result Analysis

From the results that we have obtained from MATLAB [1] we can see that power saving is directly proportional to the network activity. However, if we manage our network efficiently we can save a considerable amount of energy and extend the life time of the network nodes and user devices.

From Fig. 4.16.1 and Fig. 4.16.2, we can observe that there is a constant high network activity compared to other user devices or network node at the server end. This is acceptable as the server is the main stay of the network. However the consumption of the power of the server can be reduced depending on the flow of traffic. From the graph we have obtained from the PRTG [2] we can see that there is always high traffic at the server end in our experiment. Whereas, if we consider a real scenario, where there are multiple servers, one can implement the proposed idea to conserve energy and extend the life time of the server.

Consider the remaining two outputs from MATLAB [1] and PRTG [2], we can see that for the major part of the experiment there was low network activity. Our proposed method is more suitable in such a situation. Here we have two options as we have discussed above and show in the flow chart (Fig. 3.4); we can define ASP1 or compute ASP, based on the history of the network activity. If we have defined actionable silent period 1 based on the history of the network, the entire network interface and the user device can switch to the power down mode leaving the timer ON, so that the network interface and the user devices can be switched to active mode at the end of the actionable silent period 1. If the duration of the silent period is less than ASP1 but more than computed ASP, then the device can turn off some of the interfaces to save the power, for reference please see the PRTG (Fig. 4.17, 4.18) output graph of Laptop 2 or 3. Fuzzy logic plays a vital role in computing the duration of the ASP. However, one more application of the proposed approach can be the switching of the network node to the silent mode if the flow of traffic is not high through all the interfaces in an organization and just keeping up a high priority link.

Now, the main question that arises is about the computation of the amount of power saved and by how long we can extend the life time of the device?

Let consider our experiment which contains three laptops and a router.

Each laptop consumes 10 units of power and router consumes 5 units of power per hour.

We run the experiment for a period of 24 hours. We have also defined both ASP 1 and ASP based on the history of the network activity.

Let say that ASP1 starts at 10 PM and ends at 7 AM in the morning and ASP is of 30 minutes.

If all the devices run for the entire duration of the experiment the total power consumed will be

Total power consumed = (n*p*d) + (t*p*d) which is 840 units

Number of user device = n

Number of network nodes = t

Power required per hour = p

Total duration in hour = d

If we consider our proposed approach, than the power consumption will be,

Power saved = ± [Total power consumed without using proposed approach- (ASP 1+ASP)]

The formula used above will give the amount of power consumed by the devices which is mathematically negative of what we want; hence we have used ± sign, to negate the effect of negative value.

In our experiment, the network node and the device were switched to power down mode during the ASP 1. For ASP, the server and router were kept on for all the duration; however the laptop 1 was shut down for 3 hours and laptop 2 was shut down for 4 hours.

So the total power saved using ASP 1 + ASP is 385 units which is approximately 46 % in our experiment. However, I completely agree with the fact that the experiment was conducted in a simulated and controlled environment with the help of a minimum of devices. The result can be more realistic or better if it was conducted in a real time environment. We would also like to add that the flow of packets was controlled by keeping the individual laptop in the ideal state (laptop 1 and 2). Hence, I would like to keep a tolerance level of ±10 % for the energy efficiency using the proposed approach.

Extending Lifetime:

Predicting the lifetime of a device is not practically possible as we cannot run a device till it dies in an experiment. So, we have proposed a hypothesis for determining the time by which the life of the network nodes increases.

Consider a big firm having 100 network nodes and all working 24 hours a day, we will consider the same value of ASP1 and ASP.

ASP 1 = 9 hours and ASP = 30 min interval i.e. when node doesn't expect to originate or receive any traffic.

Considering the ASP 30 min interval, 25 nodes switch to silent mode for 1 hour, 25 nodes for 2 hours and 50 nodes for 2.5 hours during a period of 24 hours plus a defined power down mode (ASP 1 of 9 hours).

Calculating the result, we can see that using the proposed approach we can save 1100 hours out of the 2400 hours of service (if we do not use the proposed method) which is approximately 45% of the lifetime of the network nodes.

Again, I would like to add that this is a hypothetical scenario and we have considered the optimal value to enhance our result. Nevertheless, we can keep a tolerance level of ± 10 %, but in any case we are extending the lifetime of the device by considerable amount using this proposed approach.

This approach can be more beneficial in large firm, Wi-Fi or WiMax zone where there are large number of network nodes and user devices and extending life by even a (46± 10) % saves a huge amount of expense.

Advantages of silent networking using fuzzy logic;

1. Fuzzy logic is simple to understand and implement.
2. There is advancement in manufacture of consumer goods using fuzzy logic.
3. The mathematical calculation reduces significantly which reduces the complexity of the design.
4. Using fuzzy logic, the device can be switched to the silent mode even for a small duration based on the previous network activity.
5. If implemented at the hardware level, it can perform the task of awaking the neighbouring network node in case of congestion and help in avoiding the packet loss.

Chapter Five: Conclusion and Future work

5.1 Conclusion

From the experiment we have conducted and the results that we have obtained we can say that if the concept of silent networking is used we can save a considerable amount of energy and at the same time can also extend the lifetime of the network nodes and the user devices. The result we have obtained in the case of power saving is very high (46±10 %) as the conditions in which the experiment was performed was optimal for conducting the experiment. However, even if we consider the tolerance level (i.e. 10%), we are almost saving 36% of the power that would be used if we are not using the proposed method. With the rapid increase in the number of devices and the network nodes in today's world, the amount of energy that we would save using the concept of silent networking is relatively good.

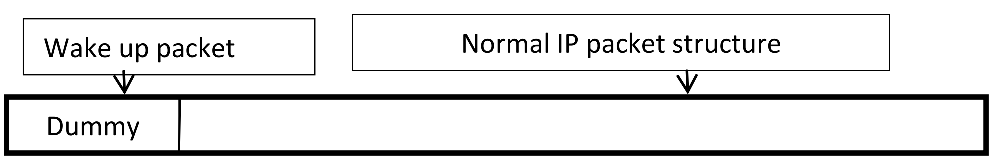
From the point of view of extending the lifetime of network nodes and user devices, as we have discussed in the analysis of the result, it is very difficult to run an experiment so long that the network node or the device dies. However, we have assumed some conditions and derived a conclusion that using the proposed method we can extend the lifetime of the device by approximately by 45 % with a tolerance level of 10 %. We have kept the tolerance level of 10 % as the assumption that we have made while calculating the increase in the lifetime of the networked devices where same to the parameters used for deriving the result of power saving (optimal condition for the experiment).

One question that comes at the end is of packet loss if the experiment is conducted in a real time environment. We agree that if this approach is implemented in a real time environment there will be packet loss; however, the amount of packet loss will be negligible compared to the amount of power we are saving and time by which we extend the lifetime of the network nodes and user devices.

However, we all know that packets traffic moves from one node to another only when the node receiving the packet receives all of the packets (including the header and trailer), if the node doesn't receive all the packets it requests the sender to send it again. This might cause another issue, i.e. delay in transmission of the packets and also slowing down of the network. However, delay should not be factor in today's world and in future as the devices are comparatively faster enough to make sure that user cannot feel the delay. However, this case might arise only when a node or device is in silent mode and it receives data packets. Even this case can be avoided if we make some changes to the structure of packets. This is discussed in future work.

5.2 Future Work

a) The issue of packet loss arises when a network node or user device is in silent mode and it receives the data; during the transition phase from silent to active mode there might be some packet loss. However, this issue can be avoided by using a dummy (Wake up packet) at the start of the packet.



b) We can achieve a better power saving and proportional increase the lifetime of the device if we can make some improvement (concept of silent networking) in the networking protocol such as IEEE 802.11 and the others.

c) The third option to increase the power saving mechanism is by enabling the device to handle enough packets before it can send i.e. having a buffer limit to handle a particular number of packets. This idea needs extensive research as currently we cannot hold a data packets if all the packets have arrived at a particular node and have a path to reach the destination.

d) One of the unexplored areas using the concept of silent networking is cloud computing. In cloud computing a lot of information is stored at a particular place (cloud) and all the users will retrieve the information from here. However, we all know that each and every user will not use the information stored using cloud computing around the clock. So, we can devise an approach that helps us in turning off some of the interface, when we do not expect relay of traffic through network devices. However, the concept of cloud computing itself is in a nascent stage, so to implement the idea might take a longer period of time [20].

References

[1]The MathWorks, Inc.; MATLAB®7 Getting Started Guide; The MathWorks, Inc., 3 Apple Hill Drive Natick, MA 01760-2098; 1984-2008; http://www.mathworks.com/

[2]Paessler the network monitoring company; Paessler Network Monitoring 9 User Manual; © 2012 Paessler AG; Printed: January 2012 in Nuremberg; http://www.paessler.com/

[3] T. Zhang *, P. Gurung, E. Berg, S. Madhani, A. Muttreja, "Silent networking for energy-constrained nodes"; Telcordia Technologies, Piscataway, NJ 08854, USA Princeton University, NJ, USA; Available online 20 March 2006.
http://www.sciencedirect.com/science/article/pii/S0140366406000946

[5] E. Shih, P. Bahl, M. Sinclair, "Wake on wireless: an event driven energy saving strategy for battery operated devices", in: Proceedings Mobicom 2002, September 23–28, 2002, Atlanta, GA.

http://research.microsoft.com/en-us/um/people/bahl/papers/pdf/mobicom02.pdf

[6] M. Stemm, R.H. Katz, Measuring and reducing energy consumption of network interfaces in hand-held devices, IEICE (Institute of Electronics, Information and Communication Engineers) Transactions on Communications, Special Issue on Mobile Computing, E 80B(8), August 1997, pp. 1125–1131.
http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.47.1888

[7] L. Curtis, Green Computing, The architecture Journal; Microsoft volume # 18; Copyright © 2008 Microsoft Corporation. All rights reserved; www.architecturejournal.net

[8] Dr. S.N. Sivanandam, S. N. Deepa, Dr. S. Sumathi; Introduction to Fuzzy Logic using MATLAB; © Springer-Verlag Berlin Heidelberg 2007; ISBN-10 3-540-35780-7 Springer Berlin Heidelberg New York; ISBN-13 978-3-540-35780-3 Springer Berlin Heidelberg New York.

[9] Lofti A. Zadeh; "Fuzzy Logic, Neural Networks, and Soft Computing"; Communication of the ACM, March 1994/ vo. 37, No. 3.
http://www.cs.berkeley.edu/~zadeh/papers/Fuzzy%20Logic,%20Neural%20Networks,%20and%20Soft%20Computing-1994.pdf

[10] A. Rajagopalan, G. Washington, G. Rizzoni, and Y. Guezennec; "Development of Fuzzy Logic and Neural Network Control and Advanced Emissions Modeling for Parallel Hybrid Vehicles"; Center for Automotive Research Intelligent Structures and Systems Laboratory The Ohio State University Columbus, Ohio; December 2003; NREL/SR-540-32919.

[11] A. Sinha, A. Chandrakasan,"Dynamic power management in wireless sensor networks", IEEE Design and Test of Computers 18 (2001) 62–74.

http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=914626&tag=1

[12] Lotfi A. Zadeh, "FUZZY LOGIC SYSTEMS: ORIGIN, CONCEPTS, AND TRENDS", Computer Science Division Department of EECS, UC Berkeley; Hong Kong; November 10, 2004; URL: http://www-bisc.cs.berkeley.edu ; URL: http://www.cs.berkeley.edu/~zadeh/

[13] W. Akkari, A. Belghith, A. Mnaouer, "Enhancing power saving mechanisms for ad hoc networks using neighborhood information",IEEE 2008; 978-1-4244-2202-9/08/

http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4600036

[14] Z. Weihong; X. Shunqing; M. Ting;"A fuzzy classifier based on Mamdani fuzzy logic system and genetic algorithm," Information Computing and Telecommunications (YC-ICT), 2010 IEEE Youth Conference on, vol., no., pp.198-201, 28-30, Nov.2010, doi:10.1109/YCICT.2010.5713079
URL:http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5713079&isnumber=5713046

[15] Taniguchi, T.; Tanaka, K.; Ohtake, H.; Wang, H.O.; "Model construction, rule reduction, and robust compensation for generalized form of Takagi-Sugeno fuzzy systems," Fuzzy Systems, IEEE Transactions on, vol.9, no.4, pp.525-538, Aug 2001 doi: 10.1109/91.940966

http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=940966&isnumber=20371

[16] http://standards.ieee.org/about/get/802/802.11.html

[17] http://www.cisco.com/en/US/products/hw/routers/index.html

[18] http://www.mathworks.com/products/fuzzy-logic/

[19] Y. He; R. Yuan; X. Ma; Jun Li; “The IEEE 802.11 Power Saving Mechanism: An Experimental Study”; On page(s): 1362 – 1367; Issue Date: March 31 2008-April 3 2008; Las Vegas, NV.

http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4489276

[20] D. Paessler, CEO at Paessler AG, “Monitoring of Private Clouds”, Published: May 2011 – Last Update: August 2011.
http://cdn.paessler.com/common/files/pdf/whitepaper/private-cloud-monitoring_en.pdf

Appendix A: Software and Hardware for Prototype Implementation

Software

1. **<u>Server Laptop:</u>**

OS: Windows 7 Ultimate, Service pack 1

2. **<u>Laptop 1:</u>**

OS: Windows 7 Home premium, Service pack 1

3. **<u>Laptop 2:</u>**

OS: Windows 7 Home premium, Service pack 1

4. **<u>Motorola Wireless router:</u>**

WPA-PSKS

Hardware

1. **<u>Server Laptop:</u>**

Processor: Intel(R)Core(TM) i5 CPU

Memory: 4 GB

Wireless Card: Atheros AR9285 Wireless

2. **<u>Laptop 1:</u>**

Processor: Intel(R)Core(TM) i3- 2330 M CPU

Memory: 4 GB

Wireless Card: Intel (R) Centrino (R) Wireless-N 1030

3. **<u>Laptop 2:</u>**

Processor: Intel(R)Pentium(R) CPU B950

Memory: 4 GB

Wireless card: Intel (R) Centrino (R) Wireless-N 100

4. **Motorola Wireless router:**

SSID – MOTOROLA- 2574D

WPA-PSK

Appendix B: Configuration of software tool:

a) Paessler monitoring tool [2]:

Paessler Network monitor

Enterprise Console V9.1.6.1962

© 2011 Paessler AG

b) MATLAB [1]:

MATLAB®, The MathWorks®, Inc.

R2011b (7.13.0.564)

64 bit (Win64)

Aug 14, 2011

License No. 407811