UNIVERSITY OF NAIROBI

SCHOOL OF COMPUTING AND INFORMATICS

A Hybrid Model Driven Scheme For Quality of Service Evaluation in Wireless Machine Type Communication Networks

by

Ngugi, Simon Kinuthia
P56/72747/2012

A research project submitted in partial fulfillment for the Requirements for the award of Degree of Masters of Science in Information Systems at the School of Computing and Informatics of the University of Nairobi

May 2014

DECLARATION

STUDENT

This research project is my original work and has not been presented for a degree in any other university.

Signed: ______________________________ Date: _____________________

Ngugi, Simon Kinuthia
P56/72747/2012

SUPERVISOR

This research project has been submitted for examination with my approval as university supervisor.

Signed: ______________________________ Date: _____________________

Professor William Okelo-Odongo
School of Computing and Informatics
University of Nairobi

DEDICATION

This research project is dedicated to Kathu and Kimani, my loving and understanding family.

ACKNOWLEDMENTS

I take this moment to acknowledge the different people who made an impact in the attainment of this research project. My supervisor Prof. Okelo-Odongo, who saw through the whole idea at the first instance and showed me the path that the research would take each step of the way. I thank the other members of the panel for their valuable advice and guidance, Prof. Timothy Waema, Mr. Eric Ayienga and the panel chair Dr. Robert Oboko whose insights were always illuminating. I also acknowledge the rest of the Faculty staff, the lab and the library for availing resources and support readily**.**

I am grateful to my fellow students in the MSc class as well as in the panel for sharing their different experiences and learnings; the discussions were very constructive. I cannot forget to thank my family for their support and patience during my many and long hours trying fulfil course requirements, their sacrifice is finally rewarded. Finally, I acknowledge all those who may have directly or indirectly contributed to this project in their own way, even though not mentioned here, they are fully appreciated.

ABSTRACT

In the very recent past, it has become increasingly clear that the next revolution in technology will be anchored on the phenomenon of the Internet of Everything (IoT), where every inanimate object has the potential to be connected to other objects in a network. Machine to Machine (M2M) communications is the base building block of the Internet of Things paradigm, and as such, it is expected that consumers will be heavily dependent on these networks at every strata of their lives, such that their interactions with their connected devices will become ubiquitous. The performance of these networks is of paramount importance to the consumer and service provider, with the consumer increasingly becoming involved in the performance deliberations. With the huge amounts of monitoring data generated from these networks, a suitable mechanism for evaluating the quality of service of these networks becomes an absolute necessity.

An effective and simple method of evaluating the performance would be to use objective QoS parameters measured from a network and characterize them into categories that would allow a consumer to subjectively evaluate them from the direct impact they are having on the consumer's experience of the network. In this research, such a method is proposed in the form a two stage model that makes up an evaluation scheme for area networks in machine type communications. The first step uses measured QoS metrics that are fed into a Fuzzy Logic System (FLS) to obtain single output values of different QoS categories that linguistically describe aspects of the performance of a network. These are then used as inputs in a second Analytic Hierarchy Process (AHP) based stage that allows a user to incorporate subjective criteria such as intuition and general observation, to finally arrive at an overall QoS value. The measured network is simulated from Castalia, a Wireless Sensor Network (WSN) and low power embedded devices simulator based on the OMNeT++ platform. The network is a small six node WSN using the IEEE 802.15.4 standard. The FLS is realized in Matlab's Fuzzy Logic Toolbox while the AHP step is done using Microsoft Excel's logic module. From the results, our model is able to show an overall QoS of Good for the simulated network as well as point to network areas that might potentially lead to QoS deterioration and might need improvements. The model compares favorably with other models that implement Fuzzy Logic and Artificial Intelligence methods for QoS Evaluation. That it incorporates subjectivity in the evaluation is a novel feature that makes the scheme more versatile compared to other QoS evaluation models.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

3GPP	3rd Generation Partnership Project
AHP	Analytic Hierarchy Process
BANs	Body Area Networks
BER	Bit Error Rate
ETSI	European Telecommunications Standards Institute
FAHP	Fuzzy Analytic Hierarchy Process
FCM	Fuzzy C-Means Clustering
FFD	Full Function Device
FIS	Fuzzy Inference System
FLS	Fuzzy Logic System
GTS	Guaranteed Timeslot
IEEE	Institute of Electrical and Electronics Engineers
IoT	Internet of Things
ITU-T	International Telecommunications Union - Telecommunications
LR-WPANs	Low Rate Wireless Personal Area Networks
M2M	Machine to Machine
MAC	Medium Access Control
MANETs	Mobile Ad hoc Networks
MOS	Mean Opinion Score
MTBF	Mean time between failures
MTC	Machine Type Communications
MTTF	Mean time to failure
MTTR	Mean time to repair
OECD	Organization for Economic Co-operation and Development
OEMs	Original Equipment Manufacturers
OSI	Open Systems Interconnection
PER	Packet Error Rate
PHY	Physical Layer
QoE	Quality of Experience

QoS	Quality of Service
RFD	Reduced Function Device
RFID	Radio-frequency identification
RX	Receive
SAN	Stochastic Activity Networks
SNMP	Simple Network Management Protocol
SVM	Support Vector Machine
TCP/IP	Transmission Control Protocol/Internet Protocol
TDMA	Time Division Multiple Access
TIA	Telecommunications Industry Association
TX	Transmit
UMTS	Universal Mobile Telecommunications System
VANETs	Vehicular Ad hoc Networks
WAP	Wireless Application Protocol
WLAN	Wireless Local Area Network
WSN	Wireless Sensor Network

CHAPTER ONE: INTRODUCTION

Machine type communications is a less technophobic reference to the technological phenomenon of machine to machine (M2M) communications. Numerous studies and market research reports have consistently predicted exponential growth for the M2M segment. Though different figures are given in each scenario in a spectrum of forecasts, for example the famous so called internet of 50 billion devices by 2020 strategy (Dimou, 2012), analysts are widely agreed that the state of this segments' growth can only be northbound (Brazell et al, 2005). As the world shifts to a connected everything state, it is envisioned that a technological breakthrough will emerge in the form of the Internet Of Things (IoT), defined as a dynamic global network infrastructure with self-configuring capabilities based on standard and interoperable communication protocols, where physical and virtual "things" have identities, physical attributes, virtual personalities that use intelligent interfaces and are seamlessly integrated into the information network (Guillemin and Friess, 2009). M2M communications will no doubt be the head cornerstones of the IoT concept, given the fact that IoT endpoints can be viewed as machines that are able to communicate with remote machines.

It has been correctly postulated that with the emergence of IoT, it is important to define service models which can categorize IoT applications and then determine factors that are necessary to satisfy the requirements of those services, which directly introduces the issue of Quality of Service (QoS) (Nef, M-A et al, 2012). In the M2M architecture, networking solutions are the most important feature, since connectivity is an essential part of the M2M. These networks can be fixed-line or wireless and in both scenarios, all appropriate and necessary factors must be considered in provisioning QoS for the services and applications that will be hosted by the networks. A natural consequence of this progression is the unmuted necessity for a flexible and robust methodology of evaluating the provisioned QoS, both at the network (Machine level) and at the end user (Humancentric) level.

1.1 Background

Machine to Machine (M2M) communications have been variously described as communication among machines, sensors, devices or "things" (Dimou, 2012), or communication between computers, embedded processors, smart sensors, actuators and mobile devices with only limited or no human intervention. A more decomposed definition says; "M2M uses a device (sensor, meter, etc.) to capture an 'event' (temperature, inventory level, etc.), which is relayed through a network (wireless, wired or hybrid) to an application (software program), that translates the captured event into meaningful information (e.g., items need to be restocked)" (Numerex Inc.). In this context, then a simple M2M architecture becomes useful as a visual aid;

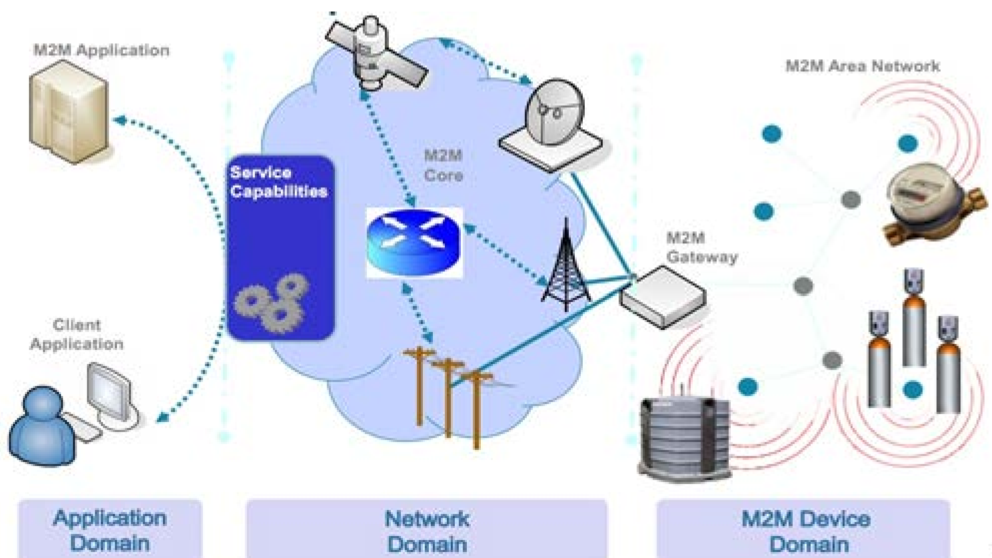

Figure 1: Simple M2M Architecture (Source www.etsi.org)

M2M has found numerous application areas with major standards bodies such ETSI and TIA taking a structured approach in standardization of the industry. 3GPP has also comprehensively categorized machine type communications with their possible uses. Original Equipment Manufacturers (OEMs) such as Ericsson have taken a keen interest in the industry and keep abreast with every development, giving out accurate periodic reports on the state of affairs.

The applications that can be supported present a mix of requirements in terms of delivery and response mechanisms, these being either real time, periodic, or once off in a given period of time. These requirements obviously necessitate careful considerations in the choice of network technology and deployment that most suites the application to be hosted. These requirements are generally termed as Quality of Service.

It is a well-known fact that Quality of Service is a term that has many different meanings to different technical communities (Chen and Varshney, 2004). This study proposes a hybrid QoS evaluation scheme that can be used to evaluate wireless machine type communication networks in a wholesome manner. A Wireless Sensor Network (WSN) will be modeled as an environmental remote monitoring network that will be sending back several measurements such as temperature, humidity and ambient light in an industrial park setting. This network will be the M2M area network in the chosen architecture (ETSI, 2008). Several QoS parameters on the WSN will be measured and used in the scheme that is made up of a two stage model; a Fuzzy Logic System (Al-Sbou, Y., 2010) that will be used in a new way to measure technology based intrinsic QoS parameters, such that singly descriptive characteristics will be output as a quantified values. The second stage is a simple Analytic Hierarchy Process (AHP) that will use the quantified QoS characteristics as criteria and application specific considerations to inform the pair-wise comparison of these criteria. The final goal of the process is an overall evaluated QoS value which will be the most highly ranked alternative.

1.2 Problem statement

The growth of mature electronic, computing and network technologies has resulted in millions, and potentially billions, of machines equipped with communication capabilities (Bonneau, 2005). These communicating machines make up the machine to machine (M2M) solutions, or machine type communications (MTC). These solutions largely allow communication between devices with minimal or no human interventions. Industry experts, scholars and analysts predict enormous growth in new connections; an AT&T study predicts 50 billion devices by 2020, another study says 225 million cellular M2M connections by 2014 (ABI Research, 2010) and yet another report mentioning 428 million embedded mobile M2M connections by 2014. These connections are what is becoming known as the Internet of Things (IoT), and their proliferation will bring benefits for the general masses as well as market opportunities for the many related stakeholders, such as manufacturers of M2M devices and components, software architects, service providers, and communication network operators (Cha, I. et al, 2009). All manner of critical and important applications are hosted by these communications systems, and their timely operation is therefore of utmost importance. As such, the network solution implemented to support a given application should not only be guaranteed by technology but also by a level of Quality of Service (QoS) that can be supported appropriately (Koucheryavy and Al-Naggar, 2013). Indeed, Lien and Chen (2011) are of the opinion that in order to achieve successful M2M communications, quality-of-service (QoS) guarantees provisioning is the most important requirement. A major component of ensuring these QoS guarantees also demands that a suitable mechanism be put in place to accurately assess the QoS supported by the system, i.e. a reliable evaluation system.

Currently, these kinds of QoS evaluation systems in machine type communication are not well researched. It has increasingly become clear in recent years that with the growth in technology and evolution of telecommunications services and networks, the notion quality of service effectively condenses to the degree of satisfaction that an end user of a service will have (ITU-T, 2008). Most of the prevalent QoS evaluation systems research is either purely objective or purely subjective, there are no hybrid models that seek to mesh objective measures with a users subjective input. This can probably be explained by the fact that many networks have so far been carrying Internet driven and multimedia traffic that has a very direct interaction with either the end user or the host network. As a consequence, QoS evaluation efforts have been concentrated on traditional wireless and fixed line networks, not on special cases such as the Wireless Sensor Network.

This situation is evidently not sustainable in an Internet of Things paradigm; network and application users for sensor type networks will increasingly require a quick and reliable means of assessing the quality of service provided by their communications set-up. Furthermore, research efforts have not really interrogated how quality of service guarantees in these kinds of wireless networks can be effectively assessed in view of the applications they're being used for, that is, how to scientifically indulge the end user in determining the QoS they are getting from their networks, especially if the network is remote and layered such that the very end user has little or no interaction with this network. This is often the case with M2M area networks such as WSNs, Mobile Ad hoc Networks (MANETs), Body Area Networks (BANs), Vehicular Ad hoc Networks (VANETs) and such others. In current practices, QoS for these kinds of networks is only estimated from how an application performs at the very end of the network where the user sits; underlying issues in the area network that may lead to poor or unacceptable performance cannot be discerned as directly or in isolation as when the area network segment is evaluated on its own. The currently proposed QoS evaluation models are heavily biased towards aggregating an overall QoS value from the discrete measured parameters, the objective treatment we have explained earlier. Models that have proposed subjective evaluation are mostly tailored towards multimedia networks, where a user is given lee-way to assess the perceived quality, even if in an unstructured manner.

This situation invites us to propose a hybrid model whose importance is manifest in the benefits it will offer. First, it will allow a user's input to be used during the evaluation, based on subjective assessment of the network's current application. This is significant in that it has not been done before and it inherently incorporates the customer satisfaction angle. Secondly, the hybrid model provides a fast and accurate method of identifying parameters that might be a cause for poor QoS. Since the measured QoS parameters are grouped into categories describing a given network characteristic, it is easy to identify and isolate any parameter or parameters that are degraded. This arrangement affords a network user the function of approaching a network problem with a wide field of view, and narrowing down to a specific aspect. Thus, this hybrid model would not only provide a fast and flexible evaluation methodology, it would also serve as a troubleshooting aid in case of network problems. This QoS evaluation scheme will therefore introduce a new methodology that is model based, accurate, fast and flexible. It will be of major importance to wireless M2M area network operators, solution providers and integrators, as well as provide new knowledge in QoS evaluation research.

1.3 Research Objectives

The study has the following objectives;

1). **Propose** a novel **quality of service evaluation scheme** that is robust and flexibly combines objective and subjective assessment methods.

2). **Demonstrate** how the proposed evaluation scheme **can be used to assess** the overall quality of service of a Wireless Sensor Network in a Machine to Machine communications set-up.

3). **Assess** the **effectiveness** of the proposed evaluation scheme by comparing its operation with other evaluation models.

1.4 Significance of the study

The issue of QoS in any networking technology is important for the simple fact that the services supported by that network technology need a certain level of reliability that can at the least be guaranteed. This is especially the case in wireless sensor networks, where the limitations and challenges are different from traditional wireless and fixed line networks (Chen and Varshney, 2004). Thus, this study has a very significant bearing on a little known field that is knowledge deficient. The results of this study are expected to have numerous benefits in several ways. First, a precise knowledge of the network QoS will be invaluable to network administrators in enabling efficient utilization of existing resources before decisions are made to commit more resources to ensure QoS guarantees. The proposed evaluation scheme provides a means of acquiring this precise knowledge. Secondly, much effort goes into the design of protocols and algorithms that enable QoS provisioning and support in the WSN. This effort would greatly be assisted by the product of this research in the way of clarity on how certain QoS parameters and their realization influences the overall QoS experience. Thirdly, with a good QoS evaluation system, new network designs will be more accurately modeled with much better predictions on the effects that factors such as the operational, mechanical and environmental conditions might have on the network during its working lifetime (Jagger, et al. 2013). This will no doubt be an important consideration for network operator planning teams and solution integrators. Fourth, the review of M2M networks, QoS issues and the expected evaluation scheme will add to the body of knowledge in this field. Lastly, the evaluation scheme that will be proposed in this work will not only be useful for QoS evaluation in wireless M2M area networks such as WSNs, the generic framework can also be extended to other types of wireless or fixed line networks employed in the IoT architecture.

1.5 Study limitations

Studies on QoS have traditionally been focused on telephony and Internet type networks, where the foremost objective was mostly on how to guarantee good voice quality and multimedia applications. However with the evolution of wireless networks that carry multimedia and bandwidth hungry applications, QoS support in wireless networks has now become an interesting and pursued area of research, but certain areas such as QoS evaluation have still not caught up. We therefore foresee some possible limitations in conducting the study as follows;

i) Lack of an existing body of knowledge on QoS evaluation methods and models especially for specialized wireless networks or in machine type communications set-ups.

ii) The number of known real wireless sensor networks at our disposal is very limited and they may not be easily or readily accessible to us. As such we may not have the opportunity to work with a real network topology, nor real time network data. This will necessitate use of simulations.

iii) Directly arising from point (ii) above, the simulations and modeling will require use of well-known tools that are verified in academia and industry such as NetSim, OPNET, Matlab's Fuzzy Logic Toolbox, etc. The licensing might be unavailable to us or prohibitive, thus necessitating use of equally good or close approximation open source tools.

CHAPTER TWO: LITERATURE REVIEW

This chapter discusses the existing body of knowledge that informs the major bearings of this study. The review begins with a brief perusal of literature that establishes why machine type communications are an important technological transformation. The review then delves into the concept of Quality of Service in a wireless networking context and briefly touches on its realization and management, thus laying a firm foundation on the ‘why’ and ‘how’ the QoS can be evaluated. An extensive presentation is done on known previous work that has presented QoS evaluation methods and models, generally in a network and narrowing down specifically on wireless sensor networks and other wireless networks. The difference in these previous efforts and our proposed method is elucidated. The review is then summarized, with an emphasis on the most appropriate related previous work.

2.1 M2M/MTC and the Internet of Things

In an incisive industry report, Juniper Networks (2011), a major networking gear vendor, presents a high level synthesis of how the industry is adopting and accommodating M2M/MTC solutions towards the Internet of Things. The report confirms that the market segment is on an upward path as M2M is projected to grow exponentially in the coming years. This is not an isolated claim, it has been reinforced by several other studies; Ericsson has consistenly predicted 50 billion connecetd devices by 2020 (Ericsson, 2011), same as Cisco (Evans, 2011). Academic discourse is also widely prevalent on a similar vein, with several foundational papers published discussing overviewsof M2M(Gupta and Hirdesh, 200-) and and its expected impact on the telecommunications industry (Galetić, et al 2011). The main applications of M2M can be summarized in the Tables 1 and 2 as below (Juniper Networks, 2011).

Service Area	MTC Applications
Security	Surveillance systems
	Backup for landline
	Control of physical access
Tracking, Tracing	Fleet management
	Order management
	Pay as you drive
	Asset tracking
	Navigation
	Traffic Information
	Road tolling
	Road traffic optimization/steering
Payment	Point of sales
	Vending machines
	Gaming machines
Health	Monitoring vital signs
	Supporting the aged or handicapped
	Web access telemedicine points
	Remote diagnostics
Remote Maintenance/Control	Sensors
	Lighting
	Pumps
	Valves
	Elevator control
	Vending machine control
	Vehicle diagnostics
Metering	Power
	Gas
	Water
	Heating
	Grid control
	Industrial metering
Consumer Devices	Digital photo frame
	Digital camera
	e-book

Table 1: 3GPP Machine-Type Communication Applications (Source: Juniper Networks)

Domains	Example
Security	Surveillance application, alarms, tracking (object/person).
Transportation	Fleet management, emission control, toll payment, road safety.
Healthcare	Related to eHealth, personal well-being and security.
Smart Energy	Measurement, provisioning/billing of utilities, metering.
Supply/Provisioning	Freight supply and distribution monitoring, vending machines.
City Automation	Public lighting, waste management.
Future	Many more new domains to be developed.

Table 2: ETSI's high-level domains. (Source: Juniper Networks)

As can be seen the spectrum of possible M2M usage is large. There are also future development reports like Wu et. al. (2011) and Dohler et. al (2010) from specialised technical bodies, as well as a comprehensive examination of M2M with respect to mobile wireless networks from OECD (2012). These works leave no doubt that indeed M2M is the future and requires the full interest of the technology fraternity in academia and industry.

2.2 The Networks, Sensors and QoS.

2.2.1 Quality of Service

Hardy's (2001) seminal work on QoS measurement and evaluation for telecommunications services casts a foundation on our chosen line of enquiry. In the work, QoS is broadly defined in terms of a technical outlook, what is termed as intrinsic QoS, and two user centric aspects that encompass the service users' subjective determination, i.e. the perceived QoS and the resultant assessed QoS. Definitions on several key concepts such Measures, Evaluation, Quantifiers and Metrics are then discussed illustratively. Measurement and evaluation on network accessibility, routing speed and reliability, connection reliability and connection continuity are some of the main topics covered concerning QoS for telecommunications services.

In this work, we propose to define QoS in the context of measurement and evaluation of telecommunications services. As such, the concept of QoS has three distinct notions that contribute to the eventual evaluation of how a network has performed (Hardy, 2001). The first one is *intrinsic* quality of service, which basically encompasses the technical design of the network and how it is provisioned technically to handle access and capacity as demand might require. This QoS is therefore measurable from a set of metrics or parameters that can be used to gauge operational performance and thus compared against set values.

The second aspect is the *perceived* QoS, which is essentially a synthesis of a user's response to the intrinsic QoS they experience once they have used the network service. This response will usually be influenced by comparison with other previous experience from a similar network service. Then there is the *assessed* QoS, which is really the other side of the perceived QoS coin, given that it is a user's determination of whether the QoS is good enough for continued use of the service. In another form, we are saying that perceived and assessed QoS are the human rationalizations of the machine oriented intrinsic QoS. From this discussion, the important facet is then how to use the intrinsic QoS measures to quantitatively describe the perceived QoS which we have seen is all important in how a service will be experienced by the user.

It therefore follows that in a Machine to Machine communication scenario, the most effective manner of evaluating the QoS a service or application will get from a network is to assess the intrinsic QoS quantitatively, then extend this QoS to an appropriate mechanism that can allow a human user to assess the QoS. A good way to realize this is to view the QoS nomenclature in a framework as proposed in Figure 2.

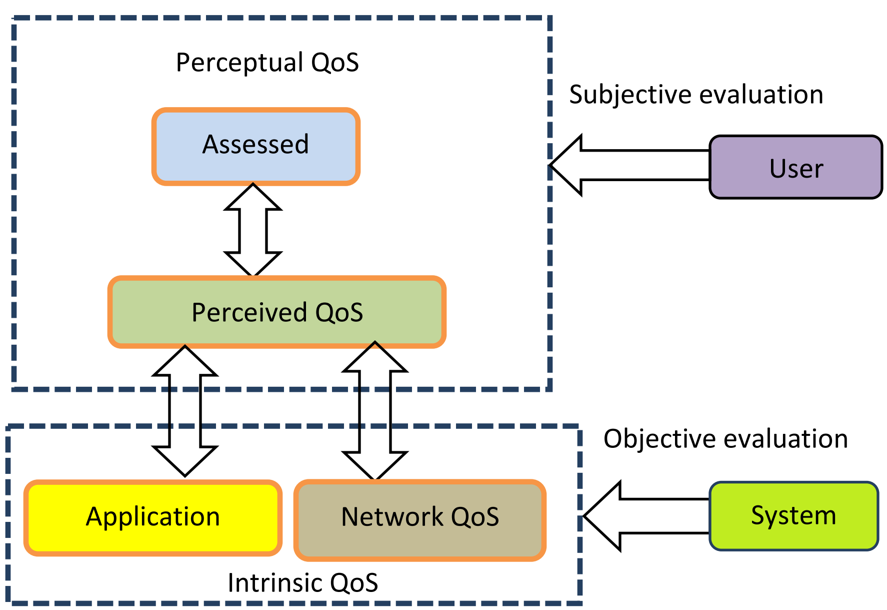

Figure 2: A hybrid QoS framework (Source: Author)

The two aspects of perceived and assessed QoS represent the human centric subjective QoS evaluation (Marshall, 2009) while intrinsic represents objective evaluation, mainly represented by analysis and mathematical methods (Al-Sbou, Y., 2010). As we focus on the intrinsic QoS, we will adopt and utilize the technology based QoS characteristics to categorize QoS parameters in a manner that gives us a leeway to effectively assess an overall QoS value. These characteristics are described in Table 3 (Chalmers and Sloman, 1999).

Category	**Parameter**	**Description/Example**
Timeliness	Delay	Time taken for a message to be transmitted
	Response time	Round-trip time from request transmission to reply receipt
	Jitter	Variation in delay or response time
Bandwidth	Systems level data rate	Bandwidth required or available, in bits or bytes per second
	Application level data rate	Bandwidth required or available, in application specific units per second, e.g., video frame rate
	Transaction rate	Number of operations requested or processed per second
Reliability	Mean time to failure (MTTF)	Normal operation time between failures.
	Mean time to repair (MTTR)	Down time from failure to restarting normal operation
	Mean time between failures (MTBF)	MTBF = MTTF + MTTR
	Percentage of time available	MTTF/MTTF + MTTR
	Loss or corruption rate	Proportion of total data that does not arrive as sent, e.g., network error rate
	Bit Error Rate	BER
	Packet Error Rate	PER

Table 3: Technology-based QoS characteristics (Source: Chalmers and Sloman, 1999)

2.2.2 Wireless Sensor Networks

As mentioned earlier, the choice of the network technology to deploy as well as the implementation directly influence the eventual QoS that a service or application will have. In this study, we focus on a specific type of a wireless network, the Wireless Sensor Network (WSN). As we have seen previously, it is a major building block of the M2M ecosystem and indeed a major component of the Internet of Things (Nef, M-A et al, 2012). Quality of service in Wireless Sensor Networks as a research area has only seen some marginal effort and this only in certain aspects such as QoS provisioning protocols and QoS aware routing algorithms. QoS evaluation for Wireless Sensor Networks in a M2M communication set-up is still largely a gray area that calls for intensive research.

Chalmers and Sloman, (1999) have given an excellent survey that seeks to categorize QoS in several ways; in terms of characteristics, the technology based and user based parameters, and in terms of QoS management where dynamic and static QoS management aspects are discussed. From this discussion, QoS can be generically defined as the set of those quantitative and qualitative characteristics of a communication system necessary to achieve the required functionality of an application or service (Bochmann and Hafid, 1999).

A number of authors have competently handled QoS considerations in wireless networks of different types, including the standards employed for M2M area networks in wireless sensor node operations. Weber, (2012) provides a survey of QoS mechanisms using 802.15.4 (Zigbee) as the primary WSN method. The study has informative conclusions on aspects such limitations to realization of a dynamic end-to-end QoS in WSN and other infrastructure (wired or wireless), it also highlights the benefit to having the QoS support both passive (static configuration) and activity managed (i.e. adjusting for changes like network size and such) scenarios. Younis et al. (2004) also visits current handling of QoS constrained traffic and its challenges. The work goes on to identify how overcoming bandwidth limitation, effective energy and delay trade-off, handling buffer size limitation, supporting multiple traffic types and the removal of redundancy are open research areas in WSNs.

Chen and Varshney (2004), have a widely cited survey on QoS support in WSNs that gives us a well-researched status on QoS requirements for WSNs vis-à-vis QoS support in traditional data networks. Using a simple QoS model in Figure 3, the work examines the challenges in supporting QoS in WSNs and reviews current research efforts in that field. These are identified as traditional end-to-end QoS, reliability assurance, and application-specific QoS.

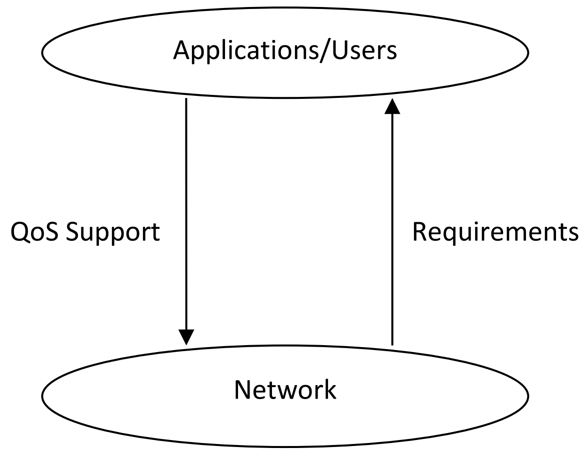

Figure 3: A simple QoS Model (Source: Chen and Varshney (2004

A major finding from this work is that end-to-end QoS parameters were inadequate measures of QoS in WSN, thus the proposal to use collective QoS parameters termed as collective latency, collective packet loss, collective bandwidth and information throughput. This conclusion will influence the formulation of our evaluation model.

Zakaria (ca 2005) provides yet another survey of QoS in WSNs that delves into issues that make QoS different in WSNs. The author substantially discusses and compares different communications protocols in WSNs with the final conclusions that QoS in WSNs need to not have very stringent requirements since these networks have inherent limitations. As well, the work concludes that QoS provision in WSNs is unnecessarily heavily influenced by traditional Internet driven models, thus calling for a paradigm shift in the thought process that goes into WSN QoS support formulation efforts.

Jäger et al. (2013) present a novel approach to QoS support investigations among other considerations by simulating a WSN designed in a specific avionic environment. The study shows how QoS parameters can be evaluated accurately, and in this case the application employed is used to check if MAC parameters are chosen well or if overload situations may occur.

2.2.3 A survey of QoS evaluation methods

Most existing work on QoS evaluation in wireless networks alludes to scarcity in existing applicable knowledge. This lends credence to the assertion that this is an open research area that is only attracting attention now with the explosive growth in mobile wireless and machine type communications, and the need for service assurance thereof. Methodologies and models that have been employed in sensor network architectures for QoS evaluation become of importance to us, and in this respect Al-Sbou (2010) is a prime reference. In that study, a performance measurement method for estimating the QoS of multimedia audio application has been proposed based on fuzzy logic. The proposed fuzzy system consists of fuzzy inputs, fuzzy rules, fuzzy reasoning and fuzzy outputs. A single hop network topology consisting of 10 nodes was simulated for a cross traffic scenario ad-hoc network meant to carry multimedia traffic. The QoS parameters of jitter, delay and packet loss were then combined to give a single output QoS value. This result is purely an objective assessment of the network with no subjective end user assessment contemplated at any stage. Moreover, the QoS parameters are inclined towards reliability and timeliness only, crucial notions in multimedia traffic. In a situation where traffic is simple but time critical, the model needs to incorporate parameters with more characteristics, and this will be addressed by the proposed scheme in our research. A hallmark of this work was the lack of complex mathematical analysis and modeling, and instead a simple but powerful technique anchored in logic. This approach is adopted in this study.

Koucheryavy and Al-Naggar (2013) propose a set-up close to Al-Sbou (2010) for physiological monitoring in e-health. The fuzzy logic is used with input parameters being quantified delay, losses and data rates while the output is an overall QoS value for physiological monitoring. As with Al-Sbou (2010), there is no aspect of end user assessment, and the QoS parameter characterization is restricted to the three metrics mentioned above only. Dogman et al. (2012) have also proposed a QoS evaluation model that combines fuzzy C-means (FCM) clustering and a regression model to analyze and assess the QoS in a simulated wireless-cum-wired network topology that is also carrying multimedia traffic. The Fuzzy C-means is used to intelligently cluster QoS parameters of delay, jitter and loss, after which the generated cluster centroids are fed into a regression model that estimates the overall QoS. The two concepts employed in this concept have well established mathematical underpinnings which are well explained. As well, the threes QoS parameters are also defined and mathematically described. The network simulations modeled a combination of the IEEE 802.11e WLAN standard and 5Mbps wired segment with 2ms propagation delay. This use of a regression model showed more accurate results compared to the fuzzy inference system method since QoS obtained from devised regression model spanned between (0%-100%) whereas the range of QoS values produced by

FIS was between (10%-90%). The authors argue that robustness of the FCM to cope with imprecise QoS patterns made it an excellent clustering mechanism.

However, while imprecise patterns can be a big problem in large data sets that need to be classified, we note that fuzzy logic is still the most appropriate technique that provides a mechanism for handling uncertainties and nonlinearities that exist in physical systems (Zadeh, 1965). Its advantage is in dealing with the complicated systems in a simple way which allows the natural description, in linguistic terms, of problems that should be solved, rather than in terms of relationships between precise numerical values (Nedeljkovic, 2004). This property of the fuzzy logic paradigm fits very well with a model that deals with the vagueness or uncertainty that an end user would typically invoke when attempting to assess something as imprecise as the notion of service quality from a network. The FCM and regression model is empirically data centric and does not jell well with the subjective evaluation aspect that our scheme seeks to integrate in its philosophical outlook from the outset.

An effort that provides another integral plank to our research is by Du et al. (2009) in which the notion of Quality of Experience (QoE) as a measure of service based on customer perception is derived from an appropriate extraction of QoS parameters and their subsequent mapping to a set of so called QoE parameters. An evaluation architecture is put forward in which a Fuzzy Analytic Hierarchy Process (FAHP) is used to evaluate a Wireless Application Protocol (WAP) service. The QoS parameter with the greatest weight contributes most to the eventual QoE value and thus it’s of interest to the Service Provider. The FAHP in this case combines qualitative and quantificational methods that guarantee rationality and accuracy of the evaluation. The qualitative aspect is of prime importance to our subjective assessment requirement in our eventual evaluation scheme.

In the research done by Song et al. (2012) a QoS evaluation model is proposed based on the evaluation index system for test- bed in a cloud computing environment. In this system, shown in Figure 4, the evaluation parameters are generally classified into subjective index, objective index and third-party index, after which each index can be divided into several sub-indexes. This approach compares well with other previous references we have looked at so far, although the specific mechanisms and eventual outcomes in the work are of no direct import to our research.

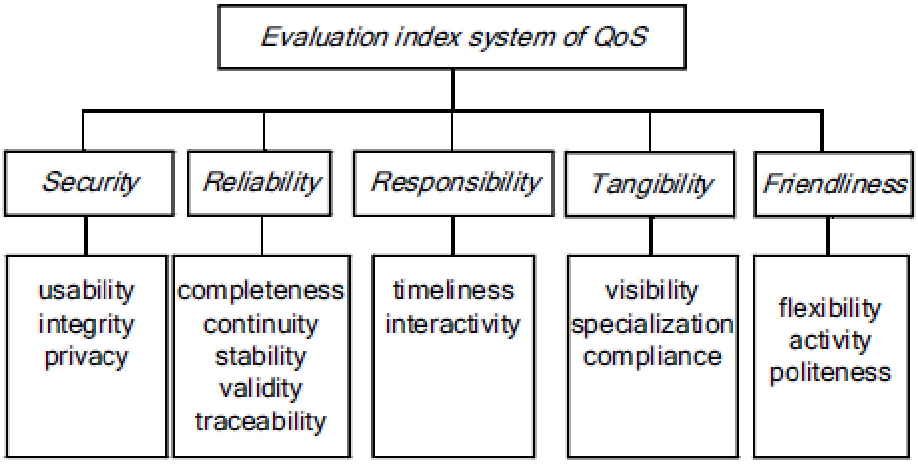

Figure 4: General Evaluation index system of QoS (Source: Song et al. (2012)

Wang et al (2011) proposes a QoS evaluation method based on the computational learning theory. The study is premised on the observation that a network needs to have a certain ability to learn dynamically from historical observation. A QoS evaluation model based on the machine learning process is thus built as in Figure 5;

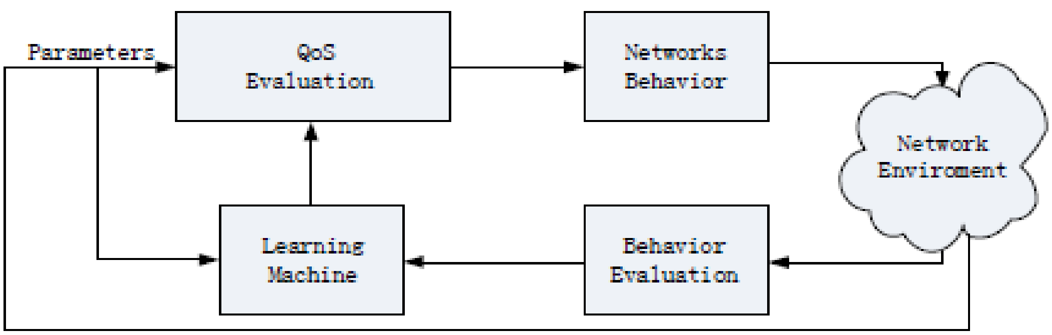

Figure 5: Behavioral model of QoS evaluation. (Source: Wang et al (2011)

The support vector machine (SVM) is employed in the model, the argument being that the SVM can deal with the problem of limited training data, a likely scenario with QoS data. A simulation was done with voice traffic over an unspecified network type with interest on two parameters of jitter and delay. Results showed that the SVM model was more accurate and robust against a linear classification model such as the Mean Opinion Score (MOS) method proposed by ITU-T recommendation P.830. Furthermore, the author intuits that the model is more suitable for the subjective assessment since the SVM is trained by the historical observation and therefore more favorable to reflect the users' experience. Apart from being mathematically complex and computationally intensive, the model also lacks the kind of instatainety that maybe required to quickly assess a quickly set-up network whose QoS information is scant. This implies that it is not friendly to users who lack knowlegde in computational or mathemtically inclined models. In contrast to this, our proposed scheme is made up of simple models that can be easily understood even with lack of networking or specialised computer knowlegde, and can be used on a temporary set-up that can provide minimum QoS measurements and some insight on the type of applications or services in use.

Several other refernces have provided useful insight into our enquiry and it is prudent to mention them in this survey of the literature. Lollini et al. (2005) give a QoS evaluation methodby modelling a UMTS cell under several service classes. The UMTS cell and some QoS measures of the cell are modelled using the stochastic network activity (SAN) formalism. The outcome is that QoS measures, relevant from both a user' s and an operator's perspective, are evaluated to better understand the underlying processes and get useful insights on proper configurations of UMTS cells. This effort is a work in progress and limits itself to a very specific scenario of only four QoS parameters, two relevant to the user and the other two to the network operator. The model is inherently predisposed to handle complex cellular network types and does not have the generic appeal to M2M area networks that we seek to have in our evalution scheme.

A simple and compelling proposal is introduced by Griera et al. (ca. 2000), where a mathematical model that discards the classical approach to QOS that considers parameters such as delay or jitter is discussed. Instead, quality is evaluated as a function of the availability indexes for either the services or the network itself. While this main idea in this work is compelling, the overall picture suffers from several severe limitations; it is structured for TCP/IP type data networks that must conditionally support SNMP, thus it is heavily biased towards LANs and WANs QoS evaluation. The theoretic derivation of the mathematical model lacks coherence, thus it does not give a firm foundation to the whole enterprise. Furthermore, the proposed model works with relative metrics in an unstructured manner, such as influnce of a network segment failure on the global network. This contrasts with what is envisioned in our evaluation scheme, where each of the models, be it for an objective or a subjective assessment, uses a well known and scientifically verified technique for quantification and evaluation.

2.3 A model driven evaluation scheme

The research work surveyed this far, if not entirely exhaustive, represents a good picture of what has been done in the area of QoS evaluation for wireless networks. A common feature that runs through existing research is a focus on guranteeing and thus finding ways of monitoring and evaluating QoS for multimedia traffic over wireless netwroks. This is entirely understandable because the growth of the internet and the wireless networks meant more and more bandwidth intensive multimedia applications finding their way into the wireless domain. However, as noted earlier, with the coming of the Internet of Things, the point in time when more “things or objects” are connected to the Internet than people (Evans, 2011) through sensor and RFID networks, QoS has inevitably become of great importance in the M2M/MTC area networks of networks that make up the IoT revolution as portrayed in Figure 6.

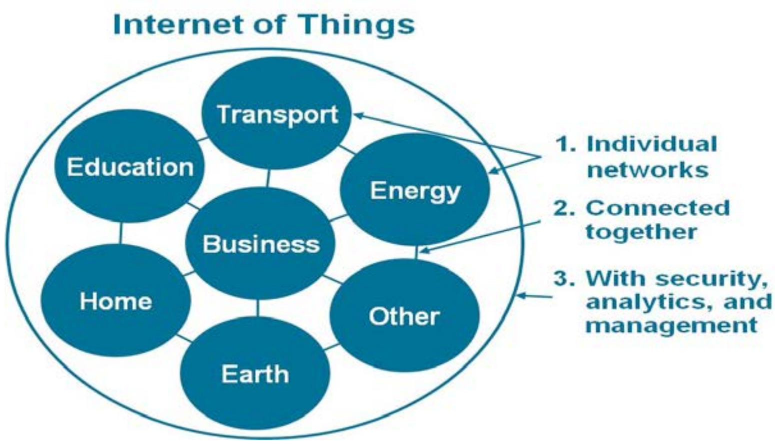

Figure 6: IoT Can Be Viewed as a Network of Networks (Source: Cisco IBSG, April 2011)

With this reason in mind, the QoS evaluation scheme proposed in this research intends to accommodate a mechanism that allows the network users to accurately determine the QoS offered by the network for the specific application or service. This QoS will ideally be an integral value of both the network QoS parameters and the users’ perception. Furthermore, it will be possible to assess the network QoS only, even if the user aspect is not required. This kind of flexibility lacks in all previously studied models, and indeed our study intends to bridge that gap. In addition, the scheme marries the well-known computation paradigm of fuzzy logic that is mathematically grounded, with a multi-criteria decision making tool, AHP, to come up with a new QoS evaluation scheme that is both simply construed and logically elegant.

2.3.1 The Fuzzy Logic System (FLS).

A fuzzy logic system (FLS) can be thought of as a nonlinear mapping of an input data set to a scalar output data. In this respect, it is unique in that it is simultaneously able to handle numeric data and linguistics variables (Mendel, 2005). A FLS consists of four main parts: fuzzifier, rules, inference engine, and defuzzifier. A general architecture showing these components is as in Figure 7 below;

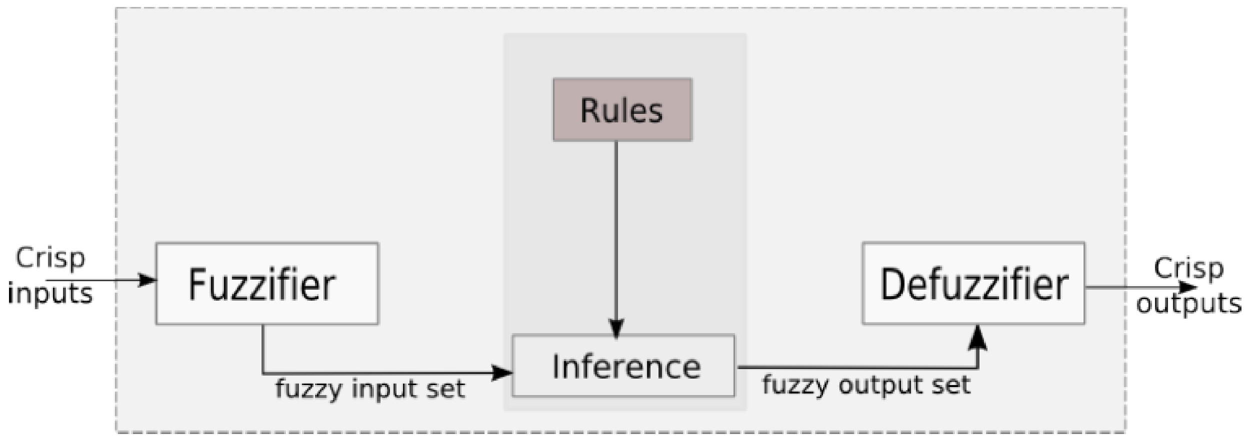

Figure 7: A Fuzzy Logic System. (Source: Bulbul, M. A. 2010)

To begin with, a crisp set of input data is converted to a fuzzy set using fuzzy linguistic variables, fuzzy linguistic terms and membership functions in a step known as fuzzification. An inference is then made based on a set of rules. The resulting fuzzy outputs are put through a defuzzification stage to map them to crisp outputs using membership functions. A generic algorithm for this scenario is as follows;

```
1. Define the linguistic variables and terms
(initialization)
2. Construct the membership functions (initialization)
3. Construct the rule base (initialization)
4. Convert crisp input data to fuzzy values using the
membership functions (fuzzification)
5. Evaluate the rules in the rule base (inference)
6. Combine the results of each rule (inference)
7. Convert the output data to non-fuzzy values
(defuzzification).
```

The linguistic variables are the input and output variables of the system whose values will normally be words or sentences from a natural language, instead of numerical values. A linguistic variable is generally broken down into a set of linguistic terms that generally describe varying levels or values of that variable (Bulbul, 2010).

Membership functions are used to quantify the linguistic variables, thus they are used at the fuzzification and defuzzification steps to map the non-fuzzy input values to fuzzy linguistic terms and vice versa. A membership function can take one of these popular shapes;

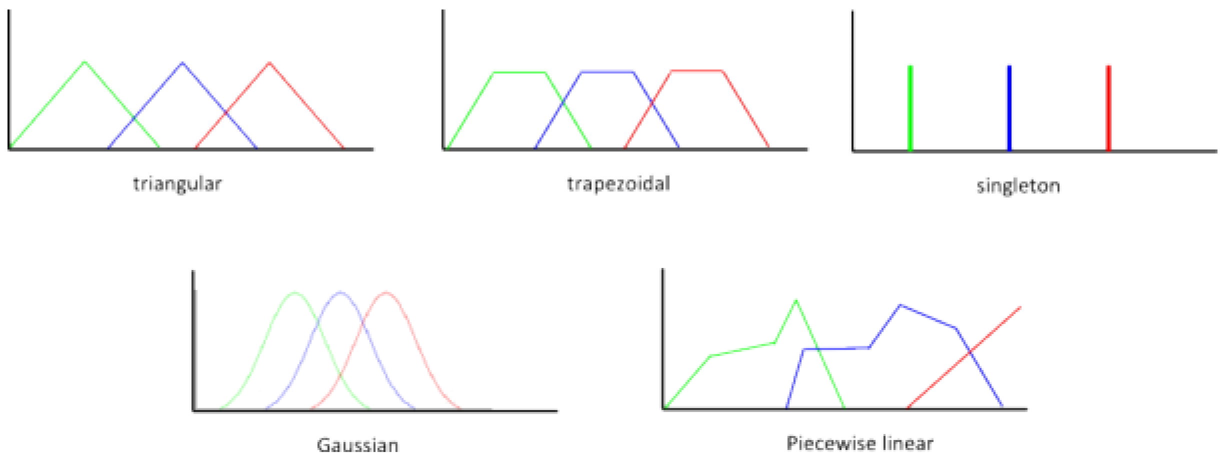

Figure 8: Membership functions shapes. (Source: Bulbul, 2010.)

The choice of the shape is chosen by the user from the context and probably experience.
Fuzzy rules are simple IF-THEN rules with condition and conclusions. In a FLS, a rule base is created to control the output variable. To evaluate the fuzzy rules and the possible combinations of the results of the individual rules, fuzzy set operations are performed. The operations on fuzzy sets are a bit different from the operations on non-fuzzy sets. If we let µA and µB be the membership functions for fuzzy sets A and B, Table 4 contains possible fuzzy operations for OR and AND operators on these sets, comparatively. The mostly used operations for OR and AND operators are max and min, respectively. Eq. 1 is used for complement (NOT) operation in fuzzy sets;

$$\mu A(x) = 1 - \mu A(x) \quad (1)$$

OR (Union)	AND (intersection)
MAX Max{μA(x),μB(x)}	MIN Min{μA(x),μB(x)}
ASUM μA(x)+μB(x)−μA(x)μB(x)	PROD μA(x)μB(x)
BSUM Min{1,μA(x)+μB(x)}	BDIF Max{0,μA(x)+μB(x)−1}

Table 4: Fuzzy set operations (Source: Bulbul, 2010.)

The process of inference results in a final result from combining the evaluated result of each rule. This combining of individual rules can be done in different ways, the maximum algorithm is however the most common. Table 5 shows accumulation methods.

Operation	Formula
Maximum	$Max\{\mu_A(x), \mu_B(x)\}$
Bounded sum	$Min\{1, \mu_A(x) + \mu_B(x)\}$
Normalized sum	$\frac{\mu_A(x)+\mu_B(x)}{Max\{1,Max\{\mu_A(x'),\mu_B(x')\}\}}$

Table 5: Accumulation methods (Source: Bulbul, 2010.)

The overall result after the inference step is a fuzzy value. This result is now defuzzified according to the membership function of the output variable to obtain a final crisp output. There are several different algorithms for defuzzification, table 6 shows the most common.

Operation	Formula
Center of Gravity	$U = \frac{\int_{min}^{max} u\,\mu(u)\,du}{\int_{min}^{max} \mu(u)\,du}$
Center of Gravity for Singletons	$\frac{\sum_{i=1}^{p}[u_i\,\mu_i]}{\sum_{i=1}^{p}[\mu_i]}$
Left Most Maximum	$U = inf(u'),\ \mu(u') = sup(\mu(u))$
Right Most Maximum	$U = sup(u'),\ \mu(u') = sup(\mu(u))$

Table 6: Defuzzification algorithms (Source: Bulbul, 2010.)

The variable in these formulae have meanings which are explained in table 7;

Variable	Meaning
U	result of defuzzification
u	output variable
p	number of singletons
μ	membership function after accumulation
i	index
min	lower limit for defuzzification
max	upper limit for defuzzification
sup	largest value
inf	smallest value

Table 7: Variable definitions. (Source: Bulbul, 2010.)

For our proposed Fuzzy Logic System, it will consist of fuzzy inputs, fuzzy rules, fuzzy reasoning and fuzzy output. Three fuzzy logic controllers (FLC) will be set up to give a single output assessment for each of the three categories we are using for QoS, that is, timeliness, bandwidth and reliability. The fuzzy input variables to be used in each case will be represented by appropriate fuzzy linguistic variables as shown in Table 8.

The membership functions for the three categories of QoS will then be defined using the range limits given. Defuzzification procedure will use the center of gravity method.

(x) Input Parameters	X^1	Parameter	Latency
		Membership function fuzzy set	{Low, Medium, High}
		Value limits	[0,100ms]
	X^2	Parameter	Response time
		Membership function fuzzy set	{Small, Average, Large}
		Value limits	[0,30ms]
	X^3	Parameter	Jitter
		Membership function fuzzy set	{Low, Medium, High}
		Value limits	[0,25ms]
(y) Output parameters		Parameter	Timeliness
		Membership function fuzzy set	{Excellent, Acceptable, Unacceptable}
		Value limits	[0,100%]

Table 8: FLC parameters for timeliness

(x) Input Parameters	X^1	Parameter	Consumed Energy
		Membership function fuzzy set	{Low, High}
		Value limits	[0J, 1J]
	X^2	Parameter	Remaining Energy
		Membership function fuzzy set	{Low, High}
		Value limits	[18719J, 18720J]
	X^3	Parameter	Estimated Network Lifetime
		Membership function fuzzy set	{Short, Long}
		Value limits	[50Days,70Days]
(y) Output parameters		Parameter	Lifetime
		Membership function fuzzy set	{Short, Long}
		Value limits	[0,100%]

Table 9: FLC parameters for timeliness

(x) Input Parameters	X^1	Parameter	Packet Loss Ratio
		Membership function fuzzy set	{Acceptable, Unacceptable}
		Value limits	[0,1]
	X^2	Parameter	Bit Error Rate (BER)
		Membership function fuzzy set	{Low, Average, High}
		Value limits	$[0,10^{-3}]$
	X^3	Parameter	Packet Error Rate (PER)
		Membership function fuzzy set	{Low, Average, High}
		Value limits	[1,5%]
(y) Output parameters		Parameter	Reliability
		Membership function fuzzy set	{Excellent, Acceptable, Unacceptable}
		Value limits	[0,100%]

Table 10: FLC parameters for reliability

Table 10 shows how each FLC will have fuzzy rules that will be determined by the number of fuzzy inputs and fuzzy sets associated with this input variable.

Rule Number	Latency	Response Time	Jitter	Timeliness
1	Low	Small	Low	Excellent
2	Low	Small	Medium	Acceptable
3	Low	Small	High	Acceptable
4	Low	Average	Low	Acceptable
5	Low	Average	Medium	Unacceptable
6	Low	Average	High	Unacceptable
7	Low	Large	Low	Acceptable
8	Low	Large	Medium	Unacceptable
9	Low	Large	High	Unacceptable
10	Medium	Small	Low	Acceptable
11	Medium	Small	Medium	Unacceptable
12	Medium	Small	High	Unacceptable
13	Medium	Average	Low	Unacceptable
14	Medium	Average	Medium	Unacceptable
15	Medium	Average	High	Unacceptable
16	Medium	Large	Low	Unacceptable
17	Medium	Large	Medium	Unacceptable
18	Medium	Large	High	Unacceptable
19	High	Small	Low	Acceptable
20	High	Small	Medium	Unacceptable
21	High	Small	High	Unacceptable
22	High	Average	Low	Unacceptable
23	High	Average	Medium	Unacceptable
24	High	Average	High	Unacceptable
25	High	Large	Low	Unacceptable
26	High	Large	Medium	Unacceptable
27	High	Large	High	Unacceptable

Table 11: Fuzzy Rules for timeliness FLC

Rule Number	Consumed Energy	Remaining Energy	Estimated Network Lifetime	Lifetime
1	Low	Low	Long	Long
2	Low	Low	Short	Long
3	Low	High	Long	Long
4	Low	High	Short	Long
5	High	Low	Long	Short
6	High	Low	Short	Short
7	High	High	Long	Short
8	High	High	Short	Short

Table 12:Fuzzy Rules for Network Lifetime FLC

Rule Number	Packet Loss Ratio	Bit Error Rate (BER)	Packet Error Rate (PER)	Reliability
1	Acceptable	Low	Low	Excellent
2	Acceptable	Low	Average	Acceptable
3	Acceptable	Low	High	Acceptable
4	Acceptable	Average	Low	Acceptable
5	Acceptable	Average	Average	Acceptable
6	Acceptable	Average	High	Acceptable
7	Acceptable	High	Low	Acceptable
8	Acceptable	High	Average	Acceptable
9	Acceptable	High	High	Unacceptable
10	Unacceptable	Low	Low	Unacceptable
11	Unacceptable	Low	Average	Unacceptable
12	Unacceptable	Low	High	Unacceptable
13	Unacceptable	Average	Low	Unacceptable
14	Unacceptable	Average	Average	Unacceptable
15	Unacceptable	Average	High	Unacceptable
16	Unacceptable	High	Low	Unacceptable
17	Unacceptable	High	Average	Unacceptable
18	Unacceptable	High	High	Unacceptable

Table 13: Fuzzy Rules for reliability FLC

2.3.2. The Analytic Hierarchy Process

Analytic Hierarchy Process (AHP) is one of the Multi Criteria decision making methods that was originally developed by Prof. Thomas L. Saaty. The input can be obtained from actual measurements of the entity under consideration, length, price, weight, duration etc., or from subjective opinion such as satisfaction feelings, preferences, past observations and such (Liu, 2008). It is a methodology for structuring, measurement and synthesis (Forman and Gaas, 2001.) This methodology involves comparison of alternatives and objectives in a natural, pairwise manner, which makes it an appropriate tool for the resolution of choice problems in a multi-criteria environment. Individual preferences are converted by the AHP into ratio-scale weights which are then combined into linear additive weights for the associated alternatives. The alternatives are then ranked using the resultant weights from the previous step. This ranking assists the decision maker in making a choice or forecasting an outcome.

Forman and Gaas (2001) believe that the AHP is more than just a methodology for choice situations and by describing it's three basic functions of structuring complexity, measuring on a ratio scale and synthesizing, they seek to show that the AHP is not just another analysis tool. This shows why AHP is widely applicable. Of note however, is the caution that the AHP is rarely used in isolation. Rather, it is used along with, or in support of, other methodologies, and this fact has been well observed and supported in our proposed study.

A mathematical summary of the AHP starts with the three commonly agreed to decision making steps:

(1) Given $i = 1, \dots, m$ objectives, determine their respective weights w_i,

(2) For each objective i, compare the $j = 1, \dots, n$ alternatives and determine their weights w_{ij} with respect to objective i, and

(3) Determine the final (global) alternative weights (priorities) W_j with respect to all the objectives by $W_j = w_{1j}w_1 + w_{2j}w_2 + \dots + w_{mj}w_m$. The alternatives are then ordered by the W_j, with the most preferred alternative having the largest W_j.

The rule is that if there are n objectives, then there are $\frac{n(n-1)}{2}$comparisons. To achieve the weights described here, a comparison matrix has to be constructed. The nature of the matrix is dictated by the number of comparisons, so if there for example 3 comparisons, then the matrix will be a 3 X 3 matrix. The matrix is then filled up with values, note the diagonal row always has a value of 1 since we're comparing criteria against itself. To fill up, first the upper triangular part of a matrix is inserted with appropriate weights. There are rules that guide this step if one uses the relative scale proposed originally, but flexibility is also allowed according to the nature of the compared entities.

The rest of the matrix is then filled up using the reciprocal values of the upper diagonal. This means if a_{ij} is the element of row i column j of the matrix, then the lower diagonal is filled using this formula

$$a_{ji} = \frac{1}{a_{ij}} \qquad (2)$$

An example matrix of three objects would look as follows in Figure 9.

$$\begin{array}{c} \\ x \\ y \\ z \end{array} \begin{array}{c} \begin{array}{ccc} x & y & z \end{array} \\ \left|\begin{array}{ccc} 1 & 1/3 & 5 \\ 3 & 1 & 7 \\ 1/5 & 1/7 & 1 \end{array}\right| \end{array}$$

Figure 9: Example comparison matrix.

Once the comparison matrix is successfully constructed, the priority vector is then computed, and this is basically the normalized Eigen vector of the matrix. There are several ways of computing the eigenvector, but the most popular is to raise the pairwise matrix to powers that are successively squared each time**,** then the rows are summed up and normalized. This is iterated until the difference between the sums in two consecutive calculations is smaller than a prescribed value or there is no difference. The values in the eigenvector are the relative ranking of each alternative. There is more that can be inferred from the computation matrix such as relative weights among the objects under comparison and the consistency of the comparisons.

2.4 Summary

Many reports have predicted exponential growth for machine to machine (M2M) networks also known as Machine Type Communications (MTC). These networks will enable communication between connected objects with minimal or no human intervention at all. Major equipment vendors, industry analysts, development organizations, academia, government agencies and regulatory bodies worldwide have recognized this reality and are appropriately preparing for a world of ‘connected everything’, popularly known as the Internet of Things (IoT). These area networks, as identified in the standardization architecture, will in no small part be largely made up of sensor networks that use wireless networking standards. The importance of ensuring the proper design, deployment and functioning of these networks therefore become a clear and present objective.

This leads to the notion of QoS, generally understood to be a measure of service assurance from a network, service or application to guarantee a certain level of performance. QoS can therefore be broadly assessed in two ways; objective QoS which is about the measurement of specific indicators in reality, and subjective QoS which is based on the subjective experience of the users (Wang et al. 2011). The topic of QoS in wireless Sensor Networks has seen some research efforts, mainly on protocol design and architecture, MAC access algorithms, energy conservation, coverage and buffer size limitations. Little research has gone into QoS evaluation, models and methodologies. Our research intends to contribute in this particular aspect and contribute a generic effort that can easily be extended to any type of communication network.

Several QoS evaluation models have been proposed in wireless networks, and the most relevant to our work adopt computational models that are mathematically supported. Chief among these are fuzzy logic based evaluations, reason being that fuzzy logic is naturally adapted to dealing with non-linearities and uncertainties in natural systems, such as would be in QoS determination. Machine learning methods are also explored: Fuzzy C Means, Regression analysis and Support Vector Machines have been used in QoS evaluation methods. While their accuracy is high, they are mathematically complex and computationally intensive, which makes them unfriendly to non-technical network users. Additionally, they are more inclined to objective QoS measurement, thus excluding the very important perceptual user assessment that is incorporated in our QoS evaluation scheme. Other methods have been explored utilizing diverse models including multi-criteria decision making tools such as Simple Additive Weighting and Fuzzy Analytic Hierarchy Process. These methods, while engaging in their quest for accuracy, do not meet our initial requirements of either simplicity, inclusiveness or both.

The QoS evaluation scheme proposed in this study will not only introduce a flexible approach to QoS assessment, it will also shift the intense focus on multimedia traffic over WSNs to other types of applications and services that have found use on diverse types of WSNs. This is a response to the current reality of a growing Internet of Things.

The Fuzzy Logic System (FLS) will be used in the two stage model. It will give out quantified single QoS characteristics values which will be used to as inputs to the AHP stage. The AHP stage will accommodate the user’s subjective comparisons in light of available measured QoS to determine an overall QoS value.

CHAPTER THREE: METHODOLOGY.

This chapter describes the methods that have been employed to achieve our proposed QoS evaluation scheme. It contains a description of the simulated Wireless Sensor Network, the QoS parameters and their derivation. A description of the Fuzzy Logic System and its realization then follows as well as the inputs and expected outputs that lead to the AHP stage and its implementation. At this point the stage will be set for the presentation and analysis of results.

3.1 The QoS framework

This research will utilize a QoS nomenclature that adopts the technology based QoS characteristics to show the difference between intrinsic, perceived and assessed QoS as discussed in section 1.1. This framework is suitable for our envisioned evaluation scheme and accommodates the hybrid of objective and subjective QoS notions that we have cultivated thus far. Therefore, we shall use the categories and QoS parameters in Table 3 as the basis of the models that will be developed.

3.2 Wireless Sensor Network Design

A wireless sensor network based on the Zigbee standard will be designed and simulated. This well-known standard adopts the IEEE 802.15.4, also known as the Low Rate Wireless Personal Area Networks (LR-WPANs) PHY and MAC layers but then proceeds to define the network layer architecture and above as correlated with the OSI layer model. The network in this work is meant to be a simplified example of a typical M2M area network; it will therefore have a small number of components and segments. The network consists of 6 wireless nodes in total, 5 being sensor nodes (Reduced Function Device (RFD) or end device), and a Full Function Device (FFD) also known as a coordinator node. The coordinator node will connect to the Network Domain in the M2M network architecture (ETSI, 2008).

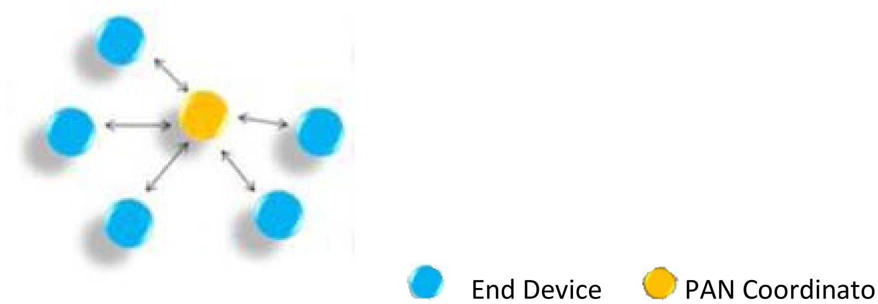

Figure 10: WSN topology (Source: Author)

3.2.1 Castalia Simulator

The simulation will be carried out on Castalia, a WSN and low power embedded devices simulator based on the OMNeT++ platform. This tool is chosen due to its support of realistic wireless channel and radio models with realistic node behavior, especially relating to access of the radio. Castalia's basic module structure is shown in the diagram below.

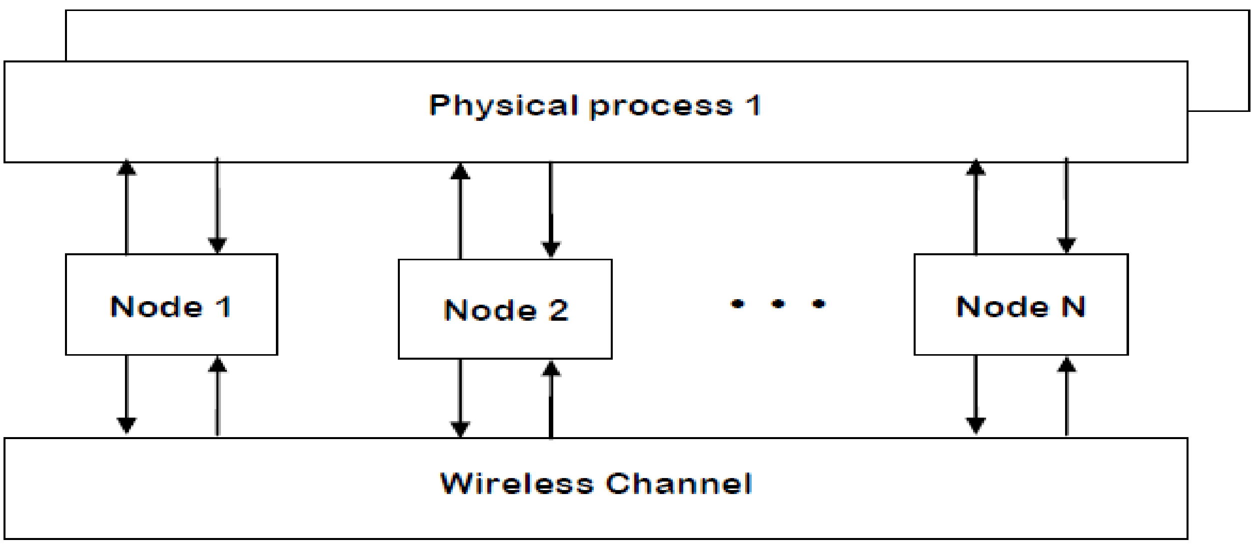

Figure 11: Castalia simulator basic structure (Source: Castalia User Manual)

The simulator is made up of several modules and is highly parameterized. Specific parameters have to be set for each simulation scenario according to what is envisioned in the outcome. These nodes are hierarchically linked as shown below in Figure 12;

```
SN
├─ node[*]
│   ├── Application {BridgeTest, ConnectivityMap, ThroughputTest, ValuePropagation,
│   │                 ValueReporting}
│   ├── Communication
│   │     ├──── MAC {BaselineBANMAC, BypassMAC, Mac802154, TMAC,
│   │     │          TunableMAC}
│   │     ├──── Radio
│   │     └──── Routing {BypassRouting, MultipathRingsRouting}
│   ├── MobilityManager {LineMobilityManager, NoMobilityManager}
│   ├── ResourceManager
│   └── SensorManager
├─ physicalProcess[*] {CarsPhysicalProcess, CustomizablePhysicalProcess}
└─ wirelessChannel
```

Figure 12: Castalia Simulator Module Hierarchy (Source: Castalia User Manual)

The critical modules to define parameters are the node and communication. Within the node module, the Sensor Network size and topology is defined. Within the communication modules, three important sub-modules are further defined, that is the MAC, Radio and Routing. Depending on the desired simulation scenario, other modules' parameters can be set accordingly. To use the IEEE 802.15.4 in our desired simulation, an application module existing in Castalia called ThroughputTest will be used. It simulates a scenario where sensor nodes periodically send a packet to a sink node, a situation that most appropriately emulates our desired simulation.

3.2.2 Castalia Installation

Since Castalia uses Omnet++ as the base platform, a current version should be installed first before Castalia can be installed. In this case Omnet++ version 4.1 was successfully installed on Ubuntu 12.04 LTS as seen below in Figure 12. The latest versions of Omnet++ are available for download at http://www.omnetpp.org/omnetpp/cat_view/17-downloads/1-omnet-releases.

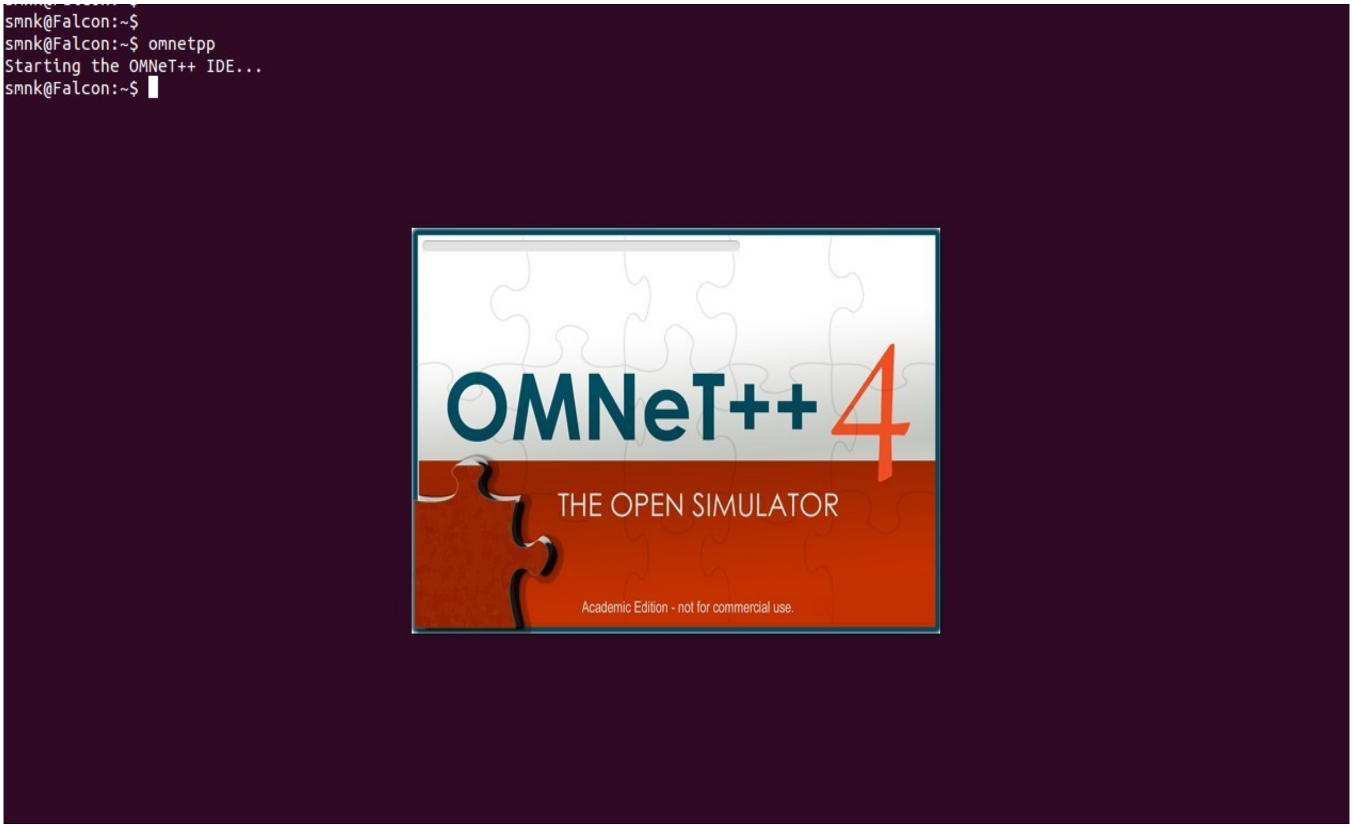

Figure 13: Launching Omnet++ after installation

Once Omnet++ was successfully installed, the latest Castalia release was then installed on the same OS, version 3.3 which was made available for download at the Castalia Github repository https://github.com/boulis/Castalia. Castalia has no default graphical user interface (GUI), any graphical representation is done as developer extensions. The directory structure after installation was as follows:

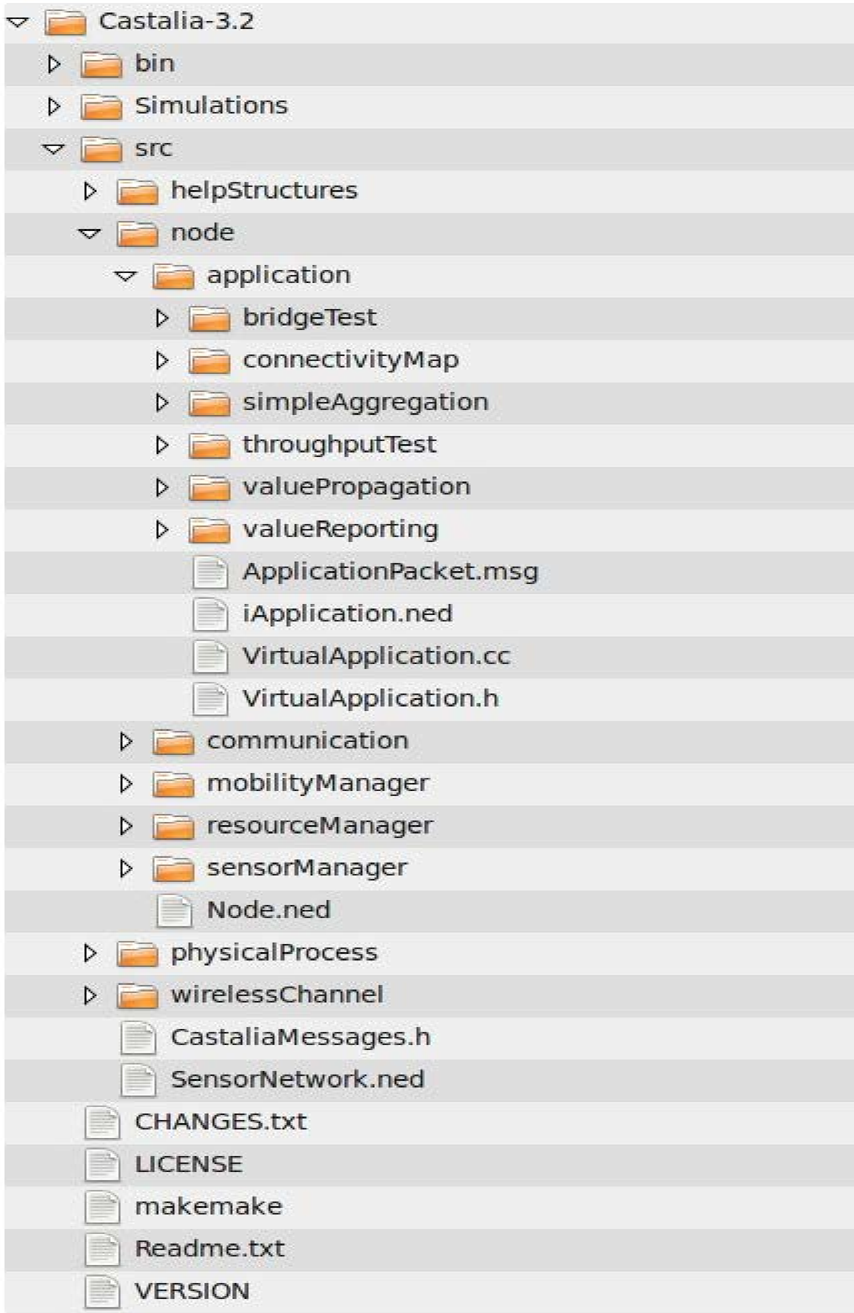

Figure 14: Castalia folder directory structure in Linux

3.2.3 Simulation set-up

For our simulation, we require a small sensor network occupying an area 40 meters by 40 meters, the kind of area occupied by a typical industrial or warehouse space. All modules and parameters were set out as shown in Table 10;

Module	**Sub-Module**	**Parameter**	**Value**
Sensor Network	Field	Field x	40m
		Field y	40m
	Nodes	numNodes	6
		Deployment	Node 0 center, 1-5 Uniform
Communication	MAC	MACProtocolName	StaticGTS802154
		Node 0 is PAN coordinator	True
		Node 0 is FFD	True
		Physical data rate	1024Kbps
		Physical Bits per symbol	2
	GTS	GTS on and GTS off	3, 0
	Radio	RadioParametersFile	BAN Radio
		Symbols for RSSI	16
		TxOutputPower	-5dBm
	Routing	Routing Protocol	Bypass Routing (No Routing).
		Node 0 is sink	True
Application		Application Name	ThroughPutTest
		Application Packet rate	5
		Data Payload	2000bytes
		Startup Delay	1ms
		Node 0 is sink	True
		Latency Maximum delay	20ms
Wireless Channel		Static nodes only	True
		Sigma (Randomness of Shadowing)	0
		Bidirectional Sigma	0
		PathLoss Expononent	2.0
		Pathloss mapfile	Provided in Castalia
		Temporal Model	Provided in Castalia

Table 14: WSN Simulator configuration parameters

Our simulation will test how Zigbee performs when its Guaranteed Time Slot (GTS) functionality is turned on or off and also when we are having a wireless channel that exhibits temporal pathloss variation versus one that does not. GTS is a TDMA-based scheme that 802.15.4 is using. In our particular simulation scenario, each round has 16 slots. Each of the 5 nodes is requesting and getting 3 slots, thus 15 slots in total are devoted to TDMA. The remaining slot is always the first slot after the beacon, and is using a contention based scheme. When GTS is off then all 16 slots are using contention-based access. For all of these 4 scenarios we will vary the packet rate of the sending nodes. We also run every scenario with 5 different seed sets.

The configuration file will normally contain sections with different configurations but must always include the Castalia binary named Castalia.ini and a general section that defines the general simulation scenario. This includes such parameters as the network size and simulation time. In addition, activating the switch for collecting trace information allows tracking of events for all the different modules involved. In our simulation case, the configuration file looks as follows:

```
[General]

# ============================================================
# Always include the main Castalia.ini file
# ============================================================
include ../Parameters/Castalia.ini

sim-time-limit = 51s  # 50 secs of data + 1 sec of MAC setup

#####  Network  #####

SN.field_x = 40       # meters
SN.field_y = 40       # meters
SN.numNodes = 6       # num of nodes + sink node
SN.deployment = "[0]->center;[1..5]->uniform"

SN.wirelessChannel.pathLossMapFile =
"../Parameters/WirelessChannel/BANmodels/pathLossMap.txt"
SN.wirelessChannel.temporalModelParametersFile =
"../Parameters/WirelessChannel/BANmodels/TemporalModel.txt"

SN.node[*].Communication.Radio.RadioParametersFile =
"../Parameters/Radio/BANRadio.txt"
SN.node[*].Communication.Radio.symbolsForRSSI = 16
SN.node[*].Communication.Radio.TxOutputPower = "-5dBm"
```

```
SN.node[*].Communication.MAC.collectTraceInfo = true
SN.node[*].Application.collectTraceInfo = true

SN.node[*].ResourceManager.baselineNodePower = 0

SN.node[*].ApplicationName = "ThroughputTest"
SN.node[*].Application.startupDelay = 1     #wait for 1sec before
starting sending packets
SN.node[0].Application.latencyHistogramMax = 600
SN.node[0].Application.latencyHistogramBuckets = 30

SN.node[3].Application.packet_rate = 5

[Config ZigBeeMAC]
SN.node[*].Communication.MACProtocolName = "StaticGTS802154"
SN.node[0].Communication.MAC.isFFD = true
SN.node[0].Communication.MAC.isPANCoordinator = true
SN.node[*].Communication.MAC.phyDataRate = 1024
SN.node[*].Communication.MAC.phyBitsPerSymbol = 2

[Config GTSon]
SN.node[*].Communication.MAC.requestGTS = 3

[Config GTSoff]
SN.node[*].Communication.MAC.requestGTS = 0

[Config noTemporal]
SN.wirelessChannel.temporalModelParametersFile = ""

[Config setRate]
SN.node[*].Application.packet_rate = 25

[Config setPower]
SN.node[*].Communication.Radio.TxOutputPower = "-15dBm"

[Config allNodesVaryPower]
SN.node[*].Communication.Radio.TxOutputPower = ${power="-10dBm","-
12dBm","-15dBm","-20dBm"}

[Config varyReTxNum]
SN.node[*].Communication.MAC.maxPacketTries = ${pktTries=1,2,3,4}
```

3.2.4 Running the simulation

To run a simulation, Castalia provides an input script that is able to take input files and different configurations within those files to run concurrently using appropriate syntactic constructs. The script has various arguments that it can take as input. Our input file called MACtest1 will be run with two scenarios; when GTS is on and when GTS is off while the nodes are varying their transmission power. There will be five different seed sets. The command and simulation progress looks as follows;

```
smnk@Falcon:~/Project/Castalia/Castalia-3.3/Castalia/Simulations/BANtest$
smnk@Falcon:~/Project/Castalia/Castalia-3.3/Castalia/Simulations/BANtest$ ~/Project/Castalia/Castalia-3.3/Castalia/bin/Castalia -i MACtest1.ini -
c ZigBeeMAC,allNodesVaryRate,[GTSon,GTSoff][noTemporal,General] -r 5
Running Castalia:       Configuration 1/4       Run 1/45        Complete 100%   Time taken 0:00:00.584000
Running Castalia:       Configuration 1/4       Run 2/45        Complete 100%   Time taken 0:00:00.566000
Running Castalia:       Configuration 1/4       Run 3/45        Complete 100%   Time taken 0:00:00.573000
Running Castalia:       Configuration 1/4       Run 4/45        Complete 100%   Time taken 0:00:00.567000
Running Castalia:       Configuration 1/4       Run 5/45        Complete 100%   Time taken 0:00:00.566000
Running Castalia:       Configuration 1/4       Run 6/45        Complete 100%   Time taken 0:00:00.627000
Running Castalia:       Configuration 1/4       Run 7/45        Complete 100%   Time taken 0:00:00.628000
Running Castalia:       Configuration 1/4       Run 8/45        Complete 100%   Time taken 0:00:00.628000
Running Castalia:       Configuration 1/4       Run 9/45        Complete 100%   Time taken 0:00:00.649000
Running Castalia:       Configuration 1/4       Run 10/45       Complete 100%   Time taken 0:00:00.633000
Running Castalia:       Configuration 1/4       Run 11/45       Complete 100%   Time taken 0:00:00.689000
Running Castalia:       Configuration 1/4       Run 12/45       Complete 100%   Time taken 0:00:00.692000
Running Castalia:       Configuration 1/4       Run 13/45       Complete 100%   Time taken 0:00:00.701000
Running Castalia:       Configuration 1/4       Run 14/45       Complete 100%   Time taken 0:00:00.734000
Running Castalia:       Configuration 1/4       Run 15/45       Complete 100%   Time taken 0:00:00.821000
Running Castalia:       Configuration 1/4       Run 16/45       Complete 100%   Time taken 0:00:00.779000
Running Castalia:       Configuration 1/4       Run 17/45       Complete 100%   Time taken 0:00:00.765000
Running Castalia:       Configuration 1/4       Run 18/45       Complete 100%   Time taken 0:00:00.780000
Running Castalia:       Configuration 1/4       Run 19/45       Complete 100%   Time taken 0:00:00.867000
Running Castalia:       Configuration 1/4       Run 20/45       Complete 100%   Time taken 0:00:00.845000
Running Castalia:       Configuration 1/4       Run 21/45       Complete 100%   Time taken 0:00:00.833000
Running Castalia:       Configuration 1/4       Run 22/45       Complete 100%   Time taken 0:00:00.809000
Running Castalia:       Configuration 1/4       Run 23/45       Complete 100%   Time taken 0:00:00.809000
Running Castalia:       Configuration 1/4       Run 24/45       Complete 100%   Time taken 0:00:00.824000
Running Castalia:       Configuration 1/4       Run 25/45       Complete 100%   Time taken 0:00:00.820000
Running Castalia:       Configuration 1/4       Run 26/45       Complete 100%   Time taken 0:00:00.880000
Running Castalia:       Configuration 1/4       Run 27/45       Complete 100%   Time taken 0:00:00.880000
Running Castalia:       Configuration 1/4       Run 28/45       Complete 100%   Time taken 0:00:00.879000
Running Castalia:       Configuration 1/4       Run 29/45       Complete 100%   Time taken 0:00:00.879000
Running Castalia:       Configuration 1/4       Run 30/45       Complete 100%   Time taken 0:00:00.883000
Running Castalia:       Configuration 1/4       Run 31/45       Complete 100%   Time taken 0:00:00.932000
Running Castalia:       Configuration 1/4       Run 32/45       Complete 100%   Time taken 0:00:01.013000
Running Castalia:       Configuration 1/4       Run 33/45       Complete 100%   Time taken 0:00:00.946000
Running Castalia:       Configuration 1/4       Run 34/45       Complete 100%   Time taken 0:00:00.941000
Running Castalia:       Configuration 1/4       Run 35/45       Complete 100%   Time taken 0:00:00.946000
```

Figure 15: Castalia simulation command and process

3.2.5 Collecting Output

Castalia allows for collection of output in simple form and histogram. For our purposes the simple form will be enough, graphs will be useful for later analysis. This is presented in form of a human readable text file saved in the form YYMMDD-HHMMSS.txt. This file will normally be processed using two other scripts provided for analyzing the results. In addition, if the collect trace info parameter is set true, another text file called Castalia-Trace is created. This trace file gets appended with new output whenever a simulation is run in the same folder. These files are shown as follows in the folder

```
smnk@Falcon:~/Project/Castalia/Castalia-3.3/Castalia/Simulations/BANtest$
smnk@Falcon:~/Project/Castalia/Castalia-3.3/Castalia/Simulations/BANtest$
smnk@Falcon:~/Project/Castalia/Castalia-3.3/Castalia/Simulations/BANtest$ ls -l
total 1663048
-rw-rw-r-- 1 smnk smnk    1443958 Feb 22 16:00 140222-155713.txt
-rw-rw-r-- 1 smnk smnk 1701482997 Feb 22 16:00 Castalia-Trace.txt
```

Figure 16: Castalia simulation output files

These files are consequently available for further processing, either with the Castalia scripts or user developed tools.

3.3 Fuzzy Logic System

The fuzzy system will be done using Matlab's Fuzzy Logic Toolbox. To develop the necessary FLCs, a Matlab installation had to be done.

3.3.1 Matlab Installation

Matlab Release R2013a for Linux was successfully installed in the same system as the simulator. MATLAB is a high-level language and interactive environment for numerical computation, visualization, and programming and is commercially available from www.mathworks.com. The installation can be invoked on the command line where the executable resides or via a desktop shortcut as shown in the figure below;

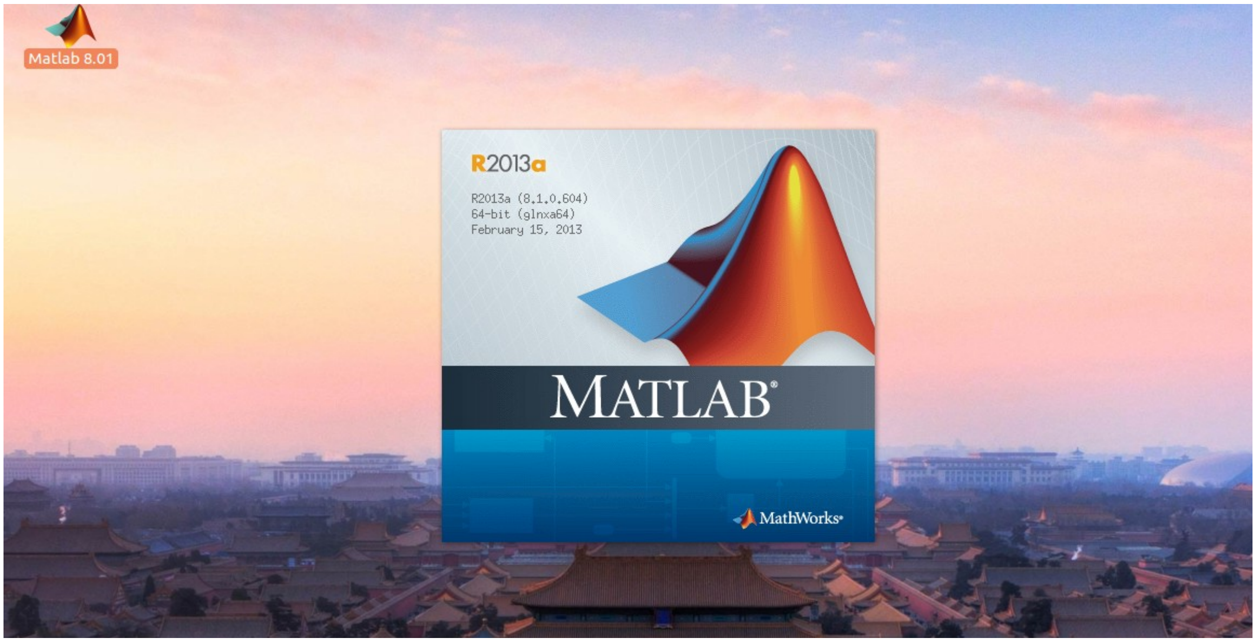

Figure 17: Matlab launch from the desktop shortcut

3.3.2 Developing the Fuzzy Logic Control Systems

The Fuzzy Logic Toolbox is launched from the Matlab command line by typing 'fuzzy' and entering;

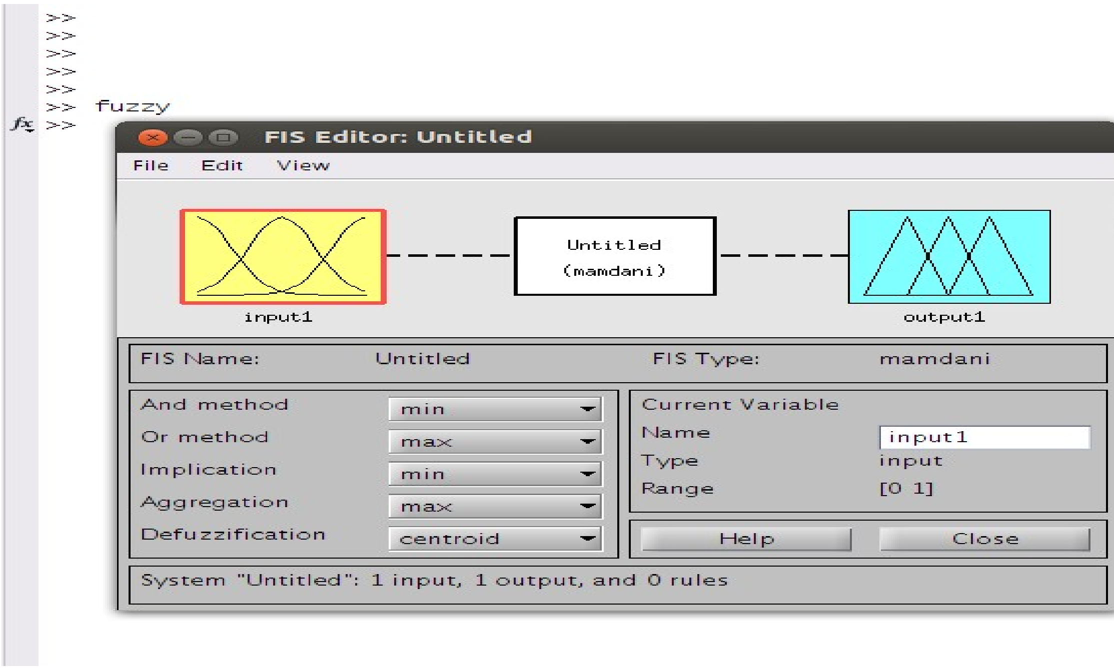

Figure 18: Launching the Fuzzy Logic Toolbox from Matlab command line

These systems are developed following the steps described in section 2.4.1; definition of the inputs and outputs, construction of appropriate membership functions and the setting of rules for the inference engine. The three FLCs appear as shown below in the Fuzzy Inference System editor screenshots in Figure 17.

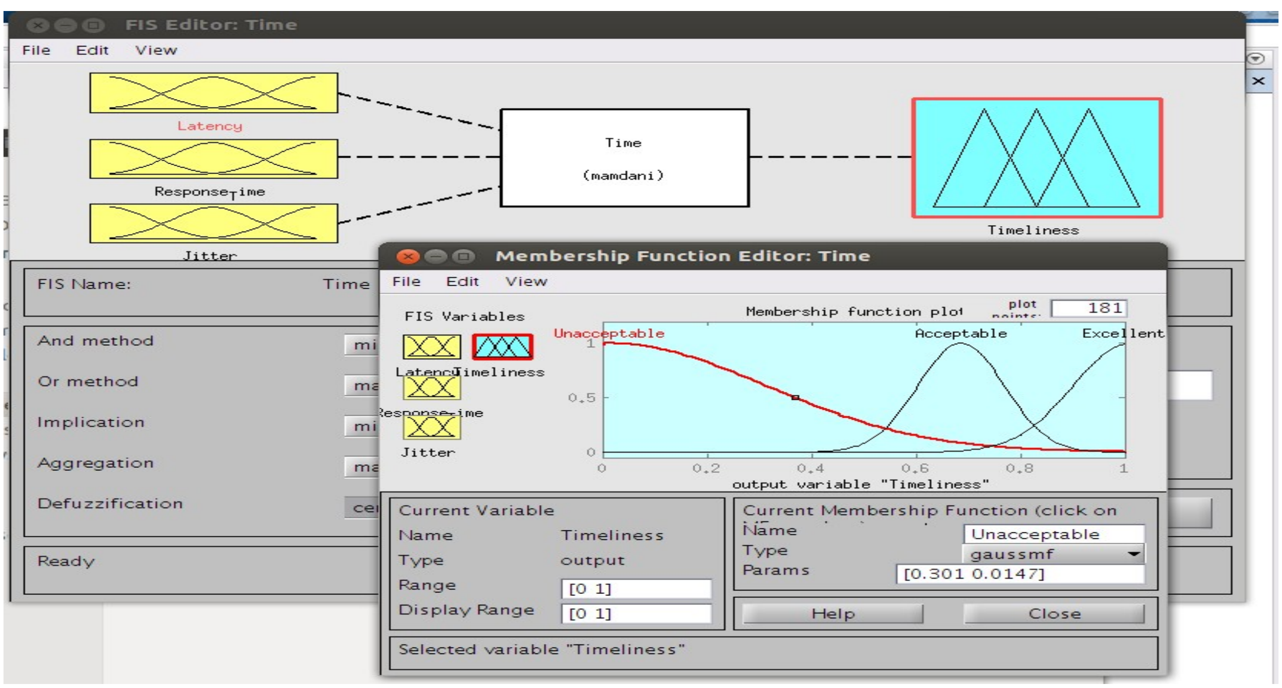

Figure 19: FIS for Timeliness

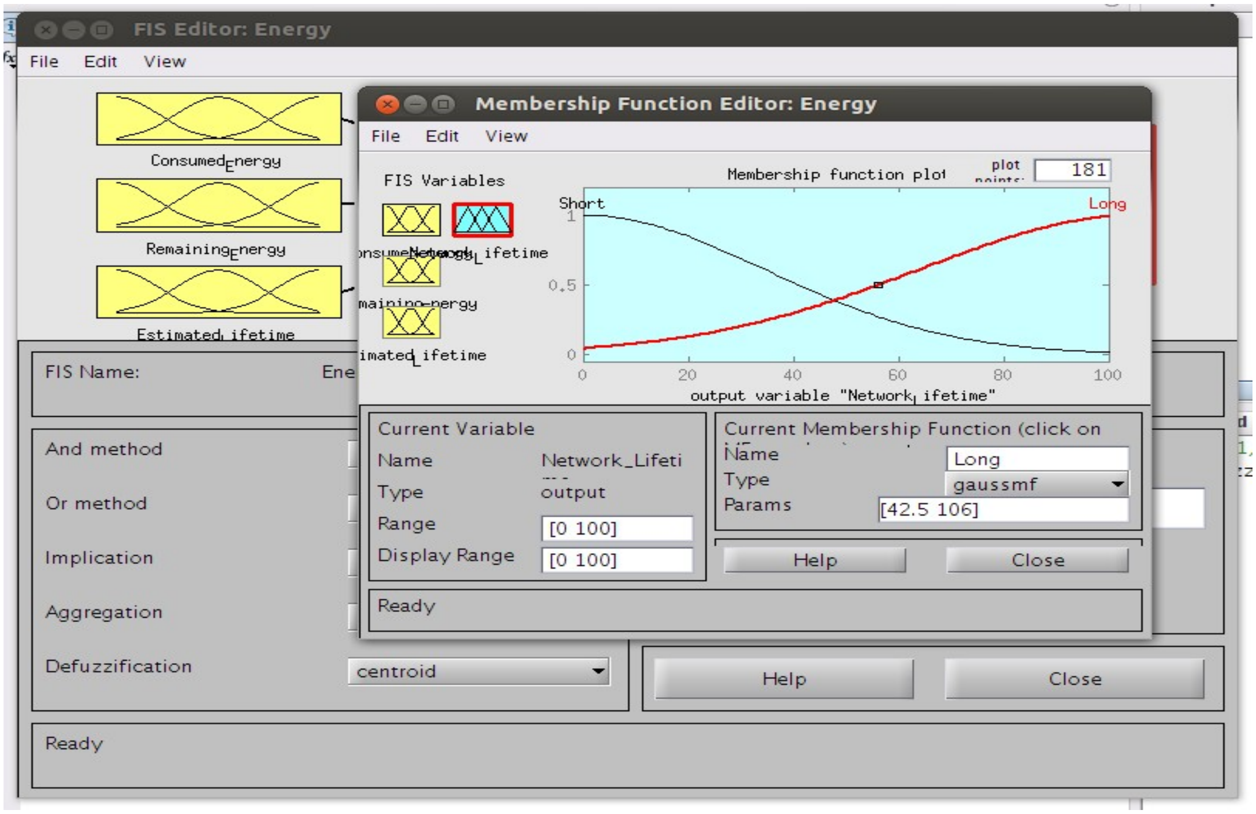

Figure 20: FIS for Network Lifetime

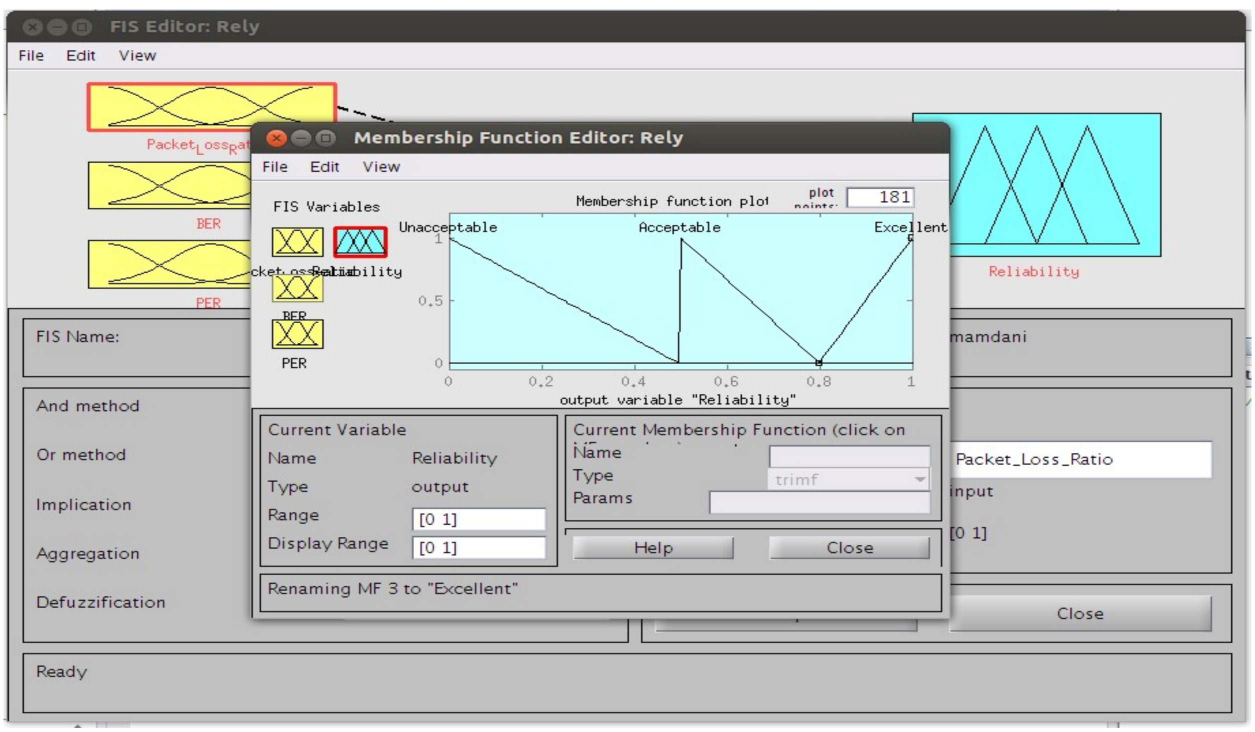

Figure 21: FIS for Reliability

These FLCs are now ready to receive values for the various input variables and give appropriate output in form of empirically quantified linguistic variables.

3.3.3 Defuzzification

The process of obtaining single output variables for the three QoS characteristics of Timeliness, Network Lifetime and Reliability is achieved via the defuzzification process as described in section 2.4.1. For each output variable, the input variables can be added on graphically or by typing on the rule viewer as shown below in case of the timeliness;

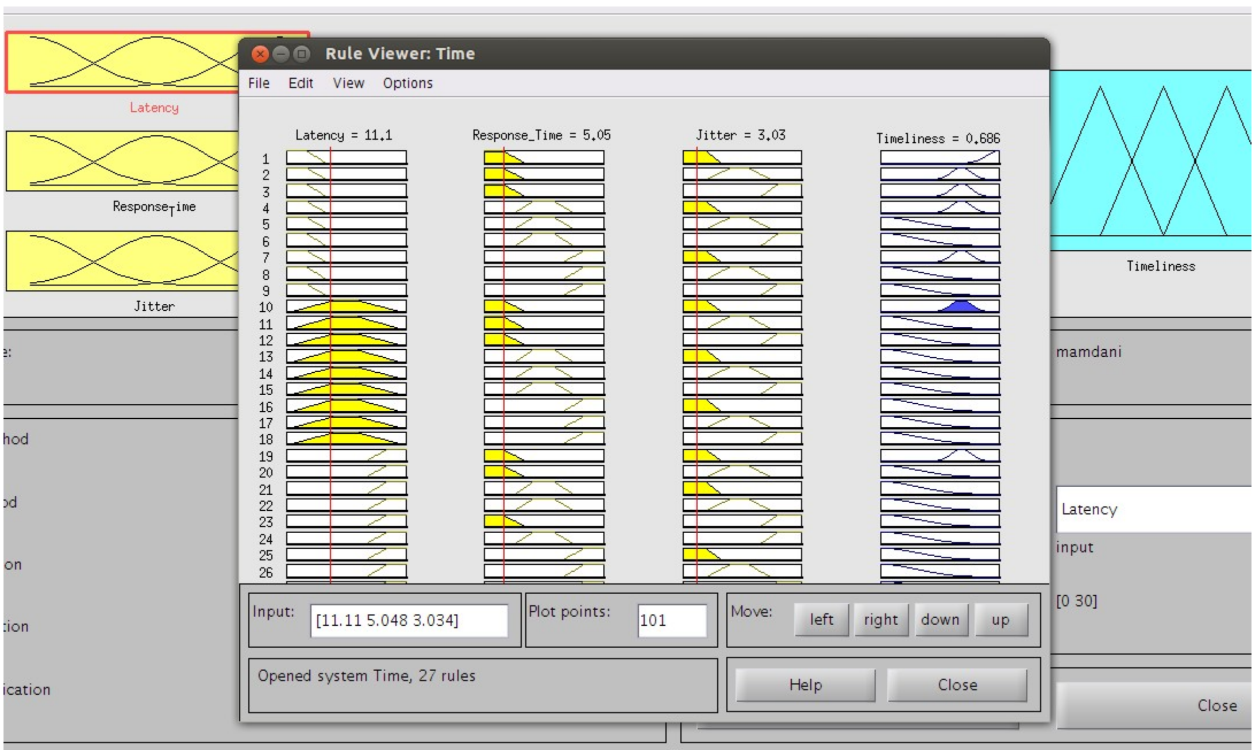

Figure 22: Defuzzification for timeliness example

The output variables at this stage are the objective QoS parameters for the WSN network we have designed and simulated.

3.4 The Analytic Hierarchy Process

The single output values for QoS characteristics obtained in section 3.2 will be used as criteria in AHP for determining the QoS of network for the remote environmental monitoring that our M2M area network implemented in section 3.2 is being used for. The hierarchy will look as shown in figure 9. The alternatives of usable, limited and unusable can be linguistically mapped to the values of Good, Average and Poor respectively.

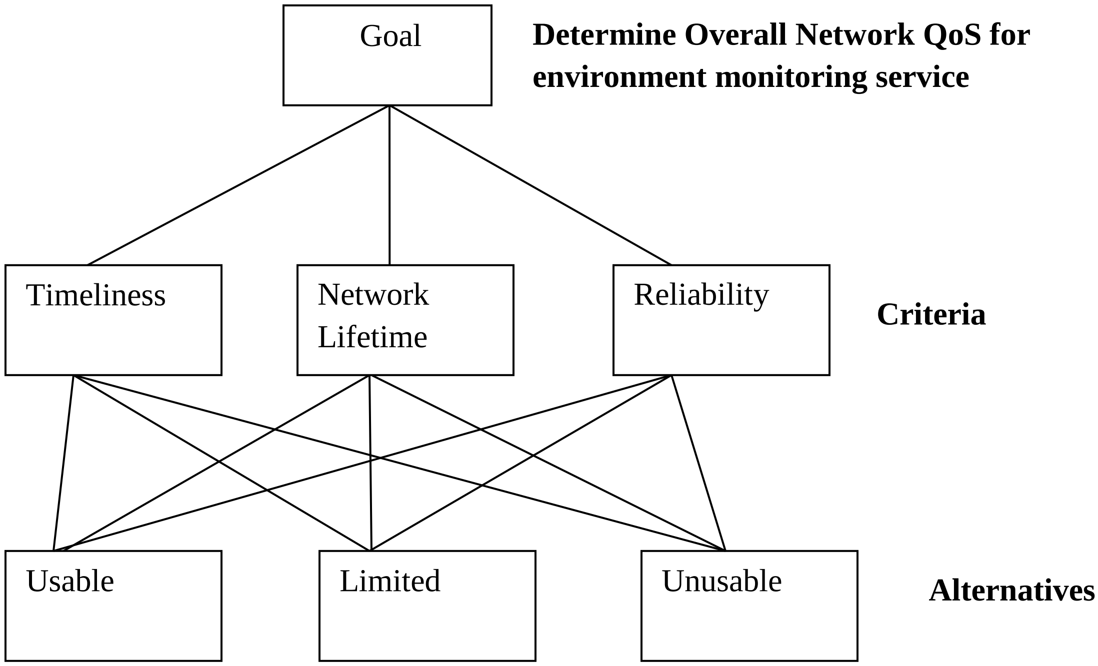

Figure 23: AHP hierarchy for QoS determination (Source: Author)

3.4.1 Pairwise Comparisons

At this crucial stage, pair-wise comparison on the criteria will be done using quantitative weights obtained from fuzzy system process and the qualitative consideration of the service on the network, in this case the environmental monitoring. A guide of the relative scale for example between timeliness and bandwidth might look as follows in figure 13:

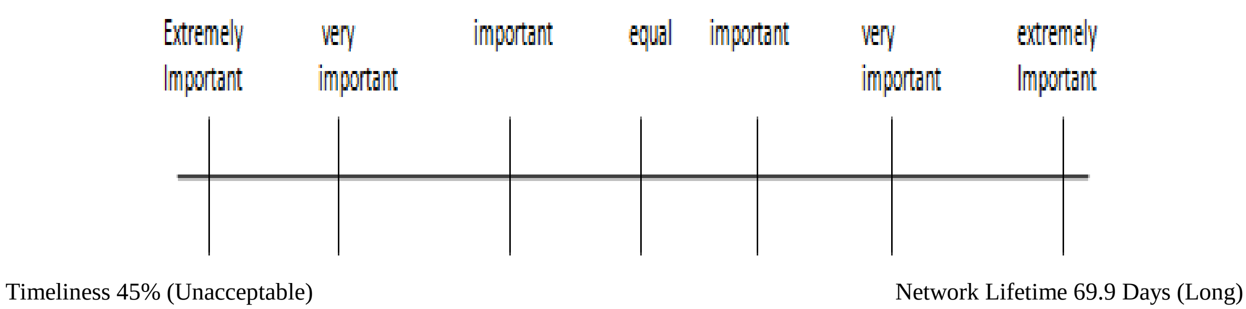

Figure 24: Example of the relative scale (Source: Liu, 2008)

The question might then be: Given that we are transmitting the temperature level in an industrial park with a timeliness of 45% which is unacceptable, on a WSN with a long lifetime of 69.9days, in this case how important is timeliness to the lifetime? These sorts of comparison will lead to comparison matrices for the criteria and alternatives. After the requisite computations, the resultant composite weight for each alternative will give the final indication of the overall QoS value. The AHP can be implemented in an any environment with logic application or programming like Matlab or even normal Microsoft Excel spreadsheet.

CHAPTER FOUR: RESULTS AND ANALYSIS

4.1 Simulation Results

Running the simulation results in two files as mentioned earlier, the output time stamped results file and a trace file. Castalia provides two scripts for processing and handling the results file; CastaliaResults and CastaliaPlot. CastaliaResults shows results in a tabulated format, mostly text oriented, while CastaliaPlot provides line graphs and histograms. From our simulation running CastaliaResults with our results file gives output that is summarized in the figures below. This output is not well suited for analysis, but used together with CastaliaPlot it becomes visually flexible for conclusive study. Each figure shows the command that gave the output.

4.1.1 WSN Simulation output

The results file appears as follows before processing with CastaliaResults;

```
Castalia| what:ZigBeeMAC,allNodesVaryRate,[GTSon,GTSoff][noTemporal,General] (5)
Castalia| when:2014-02-22 15:57
Castalia| repeat:0 label:GTSon,noTemporal,rate=14
Castalia|               module:SN.Simulation
Castalia|                       simple output name:Execution time, seconds
Castalia|                               0.673
Castalia|                       simple output name:Execution ratio (simtime/realtime)
Castalia|                               75.7858621481
Castalia|       module:SN.node[0].ResourceManager
Castalia|               simple output name:Consumed Energy
Castalia|                       0.157937
Castalia|               simple output name:Estimated network lifetime (days)
Castalia|                       69.9702
Castalia|               simple output name:Remaining Energy
Castalia|                       18719.8
Castalia|       module:SN.node[0].Communication.Radio
Castalia|               simple output name:RX pkt breakdown
Castalia|                       47 Failed with interference
Castalia|                       10 Received despite interference
Castalia|                       3492 Received with NO interference
Castalia|               simple output name:TXed pkts
Castalia|                       3927 TX pkts
Castalia|       module:SN.node[0].Communication.MAC
Castalia|               simple output name:Beacons sent
Castalia|                       425
Castalia|       module:SN.node[0].Application
Castalia|                index:1 simple output name:Packets loss rate
Castalia|                       0.00142854 total
Castalia|                index:2 simple output name:Packets loss rate
Castalia|                       0.00143063 total
Castalia|                index:3 simple output name:Packets loss rate
Castalia|                       0.00143063 total
Castalia|                index:4 simple output name:Packets loss rate
Castalia|                       0.00143063 total
Castalia|                index:5 simple output name:Packets loss rate
Castalia|                       0 total
Castalia|                index:1 simple output name:Packets received per node
Castalia|                       699
Castalia|                index:2 simple output name:Packets received per node
Castalia|                       698
:
```

Figure 25: Unprocessed results file

When CastaliaResults is given the above output file, it gives the summarized and tabulated output as below;

```
smnk@Falcon:~/Project/Castalia/Castalia-3.3/Castalia/Simulations/BANtest$
smnk@Falcon:~/Project/Castalia/Castalia-3.3/Castalia/Simulations/BANtest$
smnk@Falcon:~/Project/Castalia/Castalia-3.3/Castalia/Simulations/BANtest$ ~/Project/Castalia/Castalia-3.3/Castalia/bin/CastaliaResults -i 140222-
155713.txt

+--------------------+-----------------------------------------+-----------+
|             Module |                                  Output | Dimensions |
+--------------------+-----------------------------------------+-----------+
|        Application |       Application level latency, in ms | 1x1(31)    |
|                    |                           Energy nJ/bit | 5x1        |
|                    |                       Packets loss rate | 1x5        |
|                    |                Packets received per node | 1x5        |
|                    |                   Packets reception rate | 1x5        |
|   Communication.MAC |                        Beacons received | 5x1        |
|                    |                            Beacons sent | 1x1        |
|                    | Fraction of time without PAN connection | 5x1        |
|                    |                        Packet breakdown | 5x1(5)     |
|                    |                   pkt TX state breakdown | 5x1(2)     |
| Communication.Radio |                        RX pkt breakdown | 6x1(5)     |
|                    |                               TXed pkts | 6x1        |
|     ResourceManager |                         Consumed Energy | 6x1        |
|                    |         Estimated network lifetime (days) | 1x1        |
|                    |                        Remaining Energy | 6x1        |
|          Simulation |      Execution ratio (simtime/realtime) | 1x1        |
|                    |                  Execution time, seconds | 1x1        |
|     wirelessChannel |                  Fade depth distribution | 1x1(14)    |
+--------------------+-----------------------------------------+-----------+
NOTE: select from the available outputs using the -s option

smnk@Falcon:~/Project/Castalia/Castalia-3.3/Castalia/Simulations/BANtest$
```

Figure 26: Summarized simulation output

In addition, the trace file gives important information concerning time sequences and provides a good source for deriving time dependent parameters. This appears as shown below;

```
0.027539895218  SN.node[0].Communication.MAC        Transmitting [PAN beacon packet] now, BSN = 124
0.027539895218  SN.node[0].Application              Not sending packets
0.027680528653  SN.node[0].Communication.MAC        Attempt transmission, description: ATTEMPT_TX timer
0.027680528653  SN.node[0].Communication.MAC        Nothing to transmit
0.027710520217  SN.node[1].Communication.MAC        transmitPacket([PAN associate request],0,0,0)
0.027710520217  SN.node[1].Communication.MAC        Attempt transmission, description: transmitPacket() called
0.027710520217  SN.node[1].Communication.MAC        Transmitting [PAN associate request] in CAP, starting CSMA_CA
0.027710520217  SN.node[1].Communication.MAC        CSMA/CA random backoff value: 4, in 0.000171874999 seconds
0.027710520217  SN.node[1].Communication.MAC        Attempt transmission, description: CAP started
0.027710520217  SN.node[1].Communication.MAC        Transmitting [PAN associate request] in CAP, starting CSMA_CA
0.027710520217  SN.node[1].Communication.MAC        CSMA/CA random backoff value: 15, in 0.000601562499 seconds
0.027710520217  SN.node[4].Communication.MAC        transmitPacket([PAN associate request],0,0,0)
0.027710520217  SN.node[4].Communication.MAC        Attempt transmission, description: transmitPacket() called
0.027710520217  SN.node[4].Communication.MAC        Transmitting [PAN associate request] in CAP, starting CSMA_CA
0.027710520217  SN.node[4].Communication.MAC        CSMA/CA random backoff value: 16, in 0.000640624999 seconds
0.027710520217  SN.node[4].Communication.MAC        Attempt transmission, description: CAP started
0.027710520217  SN.node[4].Communication.MAC        Transmitting [PAN associate request] in CAP, starting CSMA_CA
0.027710520217  SN.node[4].Communication.MAC        CSMA/CA random backoff value: 30, in 0.001187499999 seconds
0.027710520217  SN.node[5].Communication.MAC        transmitPacket([PAN associate request],0,0,0)
0.027710520217  SN.node[5].Communication.MAC        Attempt transmission, description: transmitPacket() called
0.027710520217  SN.node[5].Communication.MAC        Transmitting [PAN associate request] in CAP, starting CSMA_CA
0.027710520217  SN.node[5].Communication.MAC        CSMA/CA random backoff value: 7, in 0.000289062499 seconds
0.027710520217  SN.node[5].Communication.MAC        Attempt transmission, description: CAP started
0.027710520217  SN.node[5].Communication.MAC        Transmitting [PAN associate request] in CAP, starting CSMA_CA
0.027710520217  SN.node[5].Communication.MAC        CSMA/CA random backoff value: 23, in 0.000914062499 seconds
0.028351149571  SN.node[1].Communication.MAC        Transmitting [PAN associate request] now, remaining attempts 1
0.02850614957   SN.node[0].Communication.MAC        Accepting association request from 1
0.028599905193  SN.node[0].Communication.MAC        Attempt transmission, description: ATTEMPT_TX timer
0.028599905193  SN.node[0].Communication.MAC        Nothing to transmit
0.028624612617  SN.node[5].Communication.MAC        CSMA/CA random backoff value: 17, in 0.000703065197 seconds
0.028629899569  SN.node[1].Communication.MAC        Associated with PAN:0
0.028629899569  SN.node[1].Communication.MAC        Transmission outcome for [PAN associate request]: Success
0.028629899569  SN.node[1].Communication.MAC        transmitPacket([GTS request],0,0,0)
0.028629899569  SN.node[1].Communication.MAC        Attempt transmission, description: transmitPacket() called
0.028629899569  SN.node[1].Communication.MAC        Transmitting [GTS request] in CAP, starting CSMA_CA
0.028629899569  SN.node[1].Communication.MAC        CSMA/CA random backoff value: 10, in 0.000424364396 seconds
0.0289371002    SN.node[4].Communication.MAC        Transmitting [PAN associate request] now, remaining attempts 1
0.02905426685   SN.node[1].Communication.MAC        CSMA/CA random backoff value: 9, in 0.000390619231 seconds
0.029092100199  SN.node[0].Communication.MAC        Accepting association request from 4
:
```

Figure 27: Castalia-Trace files contents

4.2. Results Processing

The simulation output provides information on several aspects of the network. This includes information on packet breakdown at the application, Medium Access Control (MAC) and radio layers. It also has data on latency at the application layer, energy consumption, simulation execution times and the fading behavior of the wireless channel when temporal variation is included. The tabulated and graphical outputs are presented in the next sections.

4.2.1 Packet Transmission and Reception Data application layer

Tabulated results on packets per node are given using packet loss rate, received packets per node and the packet reception rate at the application layer as shown below;

Application:Packets loss rate - total

	index=1	index=2	index=3	index=4	index=5
GTSoff,General,rate=14	0.08149	0.01144	0.13648	0.05579	0.13554
GTSoff,General,rate=16	0.09487	0.01402	0.13116	0.07532	0.15187
GTSoff,General,rate=18	0.08652	0.01268	0.12458	0.08051	0.12878
GTSoff,General,rate=20	0.08907	0.01741	0.12893	0.07226	0.13594
GTSoff,General,rate=22	0.08844	0.01747	0.11845	0.08297	0.14486
GTSoff,General,rate=24	0.10105	0.02819	0.13194	0.08107	0.15157
GTSoff,General,rate=26	0.10593	0.03433	0.14455	0.09469	0.16549
GTSoff,General,rate=28	0.12636	0.03703	0.15826	0.11537	0.17658
GTSoff,General,rate=30	0.13946	0.05391	0.19853	0.12236	0.20859
GTSoff,noTemporal,rate=14	0.00629	0.00229	0.0083	0.00601	0.00744
GTSoff,noTemporal,rate=16	0.00851	0.0025	0.01001	0.00676	0.00901
GTSoff,noTemporal,rate=18	0.0089	0.00245	0.01179	0.01134	0.01312
GTSoff,noTemporal,rate=20	0.01621	0.0042	0.01662	0.01601	0.01722
GTSoff,noTemporal,rate=22	0.01583	0.00637	0.0202	0.01929	0.01947
GTSoff,noTemporal,rate=24	0.02635	0.01034	0.02519	0.02335	0.02385
GTSoff,noTemporal,rate=26	0.03233	0.01339	0.0314	0.03233	0.02602
GTSoff,noTemporal,rate=28	0.04145	0.01816	0.04089	0.0386	0.04075
GTSoff,noTemporal,rate=30	0.05405	0.03082	0.05537	0.05485	0.05725
GTSon,General,rate=14	0.05605	0.00658	0.11302	0.0578	0.0815
GTSon,General,rate=16	0.05156	0.0035	0.10213	0.03979	0.08607
GTSon,General,rate=18	0.06139	0.00489	0.07809	0.04226	0.08697
GTSon,General,rate=20	0.05484	0.0046	0.08669	0.04303	0.07608
GTSon,General,rate=22	0.04932	0.00582	0.08333	0.03366	0.06915
GTSon,General,rate=24	0.05036	0.0045	0.05822	0.03403	0.0672
GTSon,General,rate=26	0.05189	0.00585	0.08097	0.02679	0.05773
GTSon,General,rate=28	0.03931	0.00415	0.09292	0.02659	0.06591
GTSon,General,rate=30	0.0431	0.00347	0.15196	0.02989	0.09529
GTSon,noTemporal,rate=14	0.00172	0.00143	0.00143	0.00143	0.00143
GTSon,noTemporal,rate=16	0.0015	0.0015	0.0015	0.0015	0.002
GTSon,noTemporal,rate=18	0.00156	0.00178	0.00156	0.00178	0.00178
GTSon,noTemporal,rate=20	0.0016	0.0022	0.0018	0.0016	0.0014
GTSon,noTemporal,rate=22	0.002	0.00164	0.00164	0.00218	0.00218
GTSon,noTemporal,rate=24	0.00167	0.00183	0.00183	0.0015	0.00167
GTSon,noTemporal,rate=26	0.002	0.002	0.00216	0.00169	0.00169
GTSon,noTemporal,rate=28	0.00172	0.00157	0.00157	0.00157	0.00143
GTSon,noTemporal,rate=30	0.00147	0.00147	0.00147	0.0016	0.00147

Application:Packets received per node

	index=1	index=2	index=3	index=4	index=5
GTSoff,General,rate=14	642.4	691	603.6	660	604.6
GTSoff,General,rate=16	723.2	787.8	694.2	739	678
GTSoff,General,rate=18	821.4	887.8	787	826.8	783.4
GTSoff,General,rate=20	910.2	982	870.2	927	863.2
GTSoff,General,rate=22	1001.8	1079.8	969	1008	939.8
GTSoff,General,rate=24	1078.2	1165.2	1040.8	1101.8	1017.6
GTSoff,General,rate=26	1161.4	1254.4	1111.4	1176	1084.2
GTSoff,General,rate=28	1222.4	1347	1177.6	1237.6	1151.8
GTSoff,General,rate=30	1289.6	1418	1201.4	1315.4	1186
GTSoff,noTemporal,rate=14	695	697.4	693.2	694.8	694.2
GTSoff,noTemporal,rate=16	792.2	797	791	793.8	792.2
GTSoff,noTemporal,rate=18	891.2	897	888.4	889	887.4
GTSoff,noTemporal,rate=20	983	995.2	982.4	983.2	981.8
GTSoff,noTemporal,rate=22	1081.6	1092	1077	1078	1077.6
GTSoff,noTemporal,rate=24	1167.8	1186.6	1168.8	1171	1170.8
GTSoff,noTemporal,rate=26	1257	1281.6	1258.4	1257	1265.4
GTSoff,noTemporal,rate=28	1341.2	1373.4	1341.8	1345	1341.8
GTSoff,noTemporal,rate=30	1417.6	1452.6	1416	1416.6	1412.8
GTSon,General,rate=14	660.2	694.4	620	658.6	642.4
GTSon,General,rate=16	757.8	796.2	717.4	767.4	730.6
GTSon,General,rate=18	844	894.8	828.8	861.2	821
GTSon,General,rate=20	944.4	994.8	912.4	956.2	923
GTSon,General,rate=22	1044.8	1092.6	1007.6	1062.2	1023
GTSon,General,rate=24	1139	1193.6	1129.2	1158.2	1118.8
GTSon,General,rate=26	1231.6	1291.4	1194	1264.2	1224.2
GTSon,General,rate=28	1344.2	1393	1269	1361.8	1306.6
GTSon,General,rate=30	1434	1493.6	1271.2	1454	1355.8
GTSon,noTemporal,rate=14	698.2	698	698	698	698.4
GTSon,noTemporal,rate=16	797.8	797.8	797.8	798	797.8
GTSon,noTemporal,rate=18	897.8	897.6	897.6	897.6	897.6
GTSon,noTemporal,rate=20	997.6	997.2	997.2	997.6	997.6
GTSon,noTemporal,rate=22	1096.8	1097.2	1097.4	1096.8	1096.6
GTSon,noTemporal,rate=24	1197.4	1196.8	1196.8	1197.2	1197.4
GTSon,noTemporal,rate=26	1296.4	1296.4	1296.4	1296.8	1297
GTSon,noTemporal,rate=28	1396.8	1396.6	1396.8	1396.8	1396.8
GTSon,noTemporal,rate=30	1496.4	1496.6	1496.8	1496.4	1496.4

Application:Packets reception rate - total

	index=1	index=2	index=3	index=4	index=5
GTSoff,General,rate=14	0.91851	0.98856	0.86352	0.94421	0.86446
GTSoff,General,rate=16	0.90513	0.98598	0.86884	0.92468	0.84813
GTSoff,General,rate=18	0.91348	0.98732	0.87542	0.91949	0.87122
GTSoff,General,rate=20	0.91093	0.98259	0.87107	0.92774	0.86406
GTSoff,General,rate=22	0.91156	0.98253	0.88155	0.91703	0.85514
GTSoff,General,rate=24	0.89895	0.97181	0.86806	0.91893	0.84843
GTSoff,General,rate=26	0.89407	0.96567	0.85545	0.90531	0.83451
GTSoff,General,rate=28	0.87364	0.96297	0.84174	0.88463	0.82342
GTSoff,General,rate=30	0.86054	0.94609	0.80147	0.87764	0.79141
GTSoff,noTemporal,rate=14	0.99371	0.99771	0.9917	0.99399	0.99256
GTSoff,noTemporal,rate=16	0.99149	0.9975	0.98999	0.99324	0.99099
GTSoff,noTemporal,rate=18	0.9911	0.99755	0.98821	0.98866	0.98688
GTSoff,noTemporal,rate=20	0.98379	0.9958	0.98338	0.98399	0.98278
GTSoff,noTemporal,rate=22	0.98417	0.99363	0.9798	0.98071	0.98053
GTSoff,noTemporal,rate=24	0.97365	0.98966	0.97481	0.97665	0.97615
GTSoff,noTemporal,rate=26	0.96767	0.98661	0.9686	0.96767	0.97398
GTSoff,noTemporal,rate=28	0.95855	0.98184	0.95911	0.9614	0.95925
GTSoff,noTemporal,rate=30	0.94595	0.96918	0.94463	0.94515	0.94275
GTSon,General,rate=14	0.94395	0.99342	0.88698	0.9422	0.9185
GTSon,General,rate=16	0.94844	0.9965	0.89787	0.96021	0.91393
GTSon,General,rate=18	0.93861	0.99511	0.92191	0.95774	0.91303
GTSon,General,rate=20	0.94516	0.9954	0.91331	0.95697	0.92392
GTSon,General,rate=22	0.95068	0.99418	0.91667	0.96634	0.93085
GTSon,General,rate=24	0.94964	0.9955	0.94178	0.96597	0.9328
GTSon,General,rate=26	0.94811	0.99415	0.91903	0.97321	0.94227
GTSon,General,rate=28	0.96069	0.99585	0.90708	0.97341	0.93409
GTSon,General,rate=30	0.9569	0.99653	0.84804	0.97011	0.90471
GTSon,noTemporal,rate=14	0.99828	0.99857	0.99857	0.99857	0.99857
GTSon,noTemporal,rate=16	0.9985	0.9985	0.9985	0.9985	0.998
GTSon,noTemporal,rate=18	0.99844	0.99822	0.99844	0.99822	0.99822
GTSon,noTemporal,rate=20	0.9984	0.9978	0.9982	0.9984	0.9986
GTSon,noTemporal,rate=22	0.998	0.99836	0.99836	0.99782	0.99782
GTSon,noTemporal,rate=24	0.99833	0.99817	0.99817	0.9985	0.99833
GTSon,noTemporal,rate=26	0.998	0.998	0.99785	0.99831	0.99831
GTSon,noTemporal,rate=28	0.99828	0.99843	0.99843	0.99843	0.99857

Figure 28: Tabulated results on all packets per node application layer

These results can be well interperted if graphed using CastaliaPlot. To start, we can graph packets received per node;

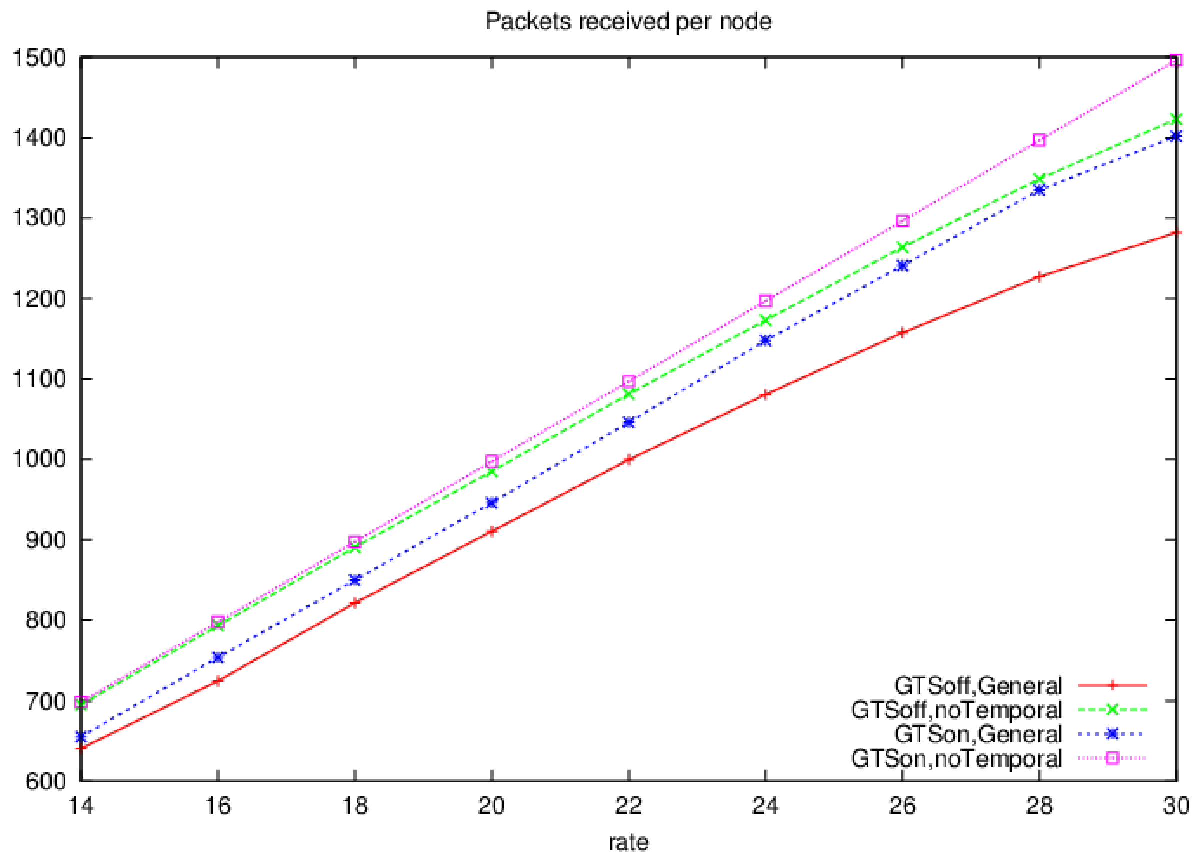

Figure 29: Graph of received packets per node against packet rate for all scenarios

In the graph, the ordinate represents the average packets received per node. In our case only node 0 receives packets, but it does so from multiple nodes, thus the "per node" refers to this aspect. The abscissa shows the sending rate for each node measured in packets/sec. From our configuration, nodes are sending packets for 50s, so if there is perfect reception then a rate of 1500 packets per node for the 30packets/sec/node case would be obtained. From the graph we can see that for low traffic the GTSon noTemporal curve almost achieves the maximum, for example at 16packets/sec/node we get 800packets received per node. This would be expected since generally the 802.15.4 protocol performs better when the GTS is turned on and the fact that TDMA schemes make a more efficient use of the wireless medium and reducing interference. Another expected fact is the better performance (in packets received) when the channel has no temporal variation, because now the deep fades introduced by the temporal variations causing breaks in the connectivity between the sender nodes and the hub are avoided. With no temporal pathloss variation the links are kept in a relatively good state.

Graphs of the packet loss ratio and packet reception rate are closely related and present the observations above in another form as seen below;

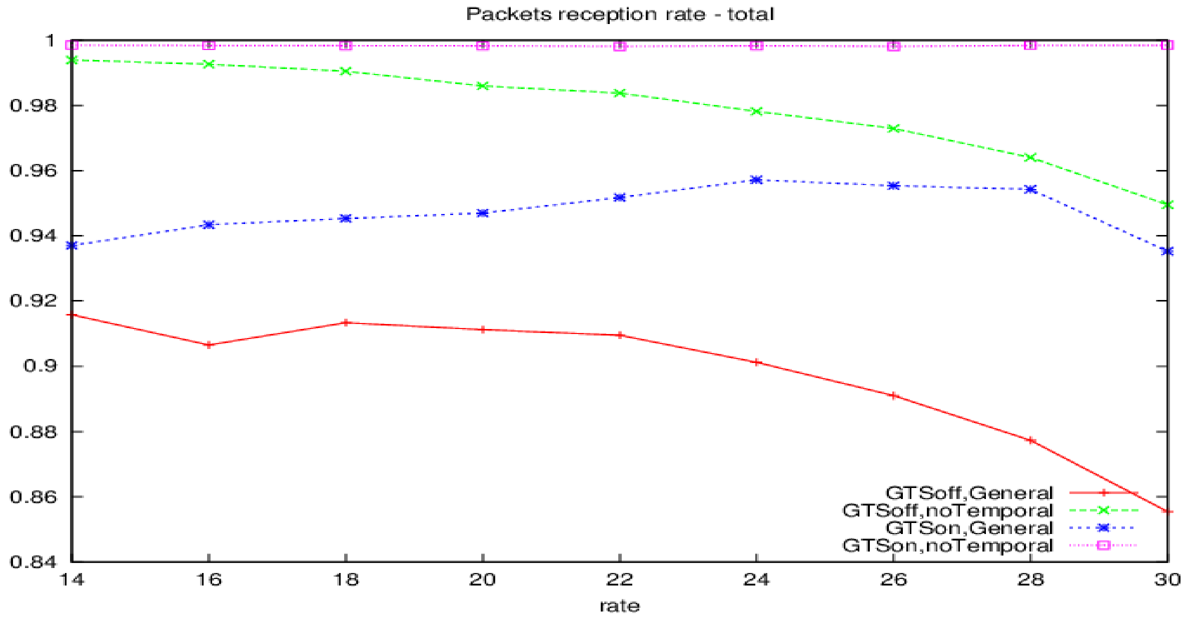

Figure 30: Graph of packets reception rate against packet rate for all scenarios

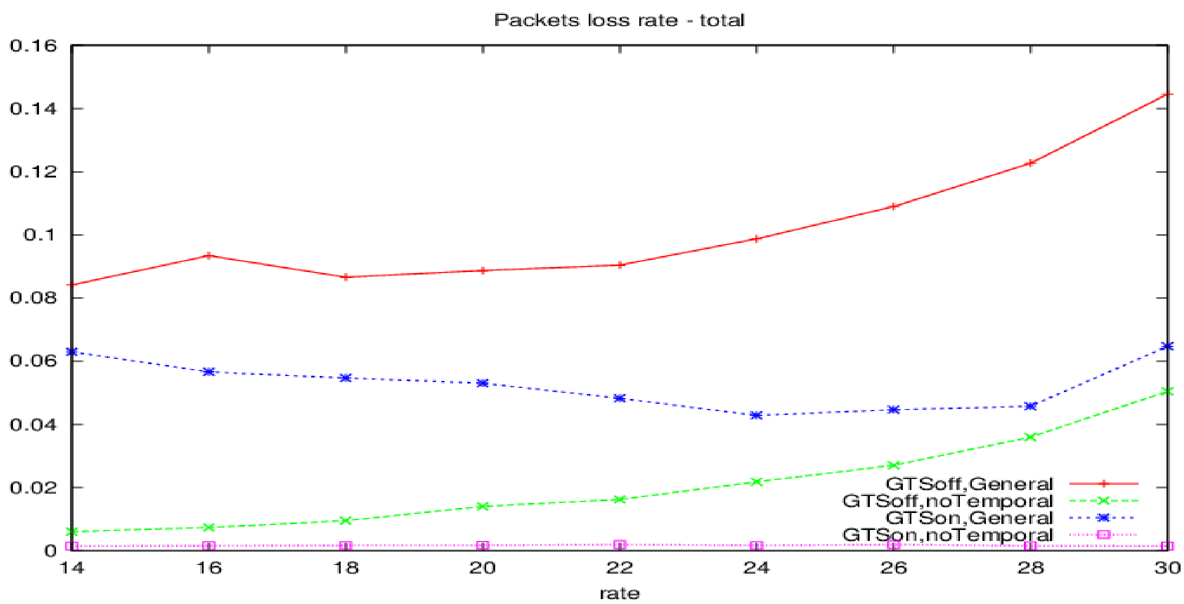

Figure 31: Graph of packets loss rate against packet rate for all scenarios

4.2.2 Packet Transmission and Reception Data MAC layer

The MAC also reports on the packet transmission state as shown below, both the tabulated and graphed output;

Communication.MAC:pkt TX state breakdown

	Contention	Contention-free
GTSoff,General,rate=14	799.4	0
GTSoff,General,rate=16	916.8	0
GTSoff,General,rate=18	1019.48	0
GTSoff,General,rate=20	1130.12	0
GTSoff,General,rate=22	1235.24	0
GTSoff,General,rate=24	1346.04	0
GTSoff,General,rate=26	1452.4	0
GTSoff,General,rate=28	1553.64	0
GTSoff,General,rate=30	1626.2	0
GTSoff,noTemporal,rate=14	723.48	0
GTSoff,noTemporal,rate=16	826.96	0
GTSoff,noTemporal,rate=18	931.08	0
GTSoff,noTemporal,rate=20	1035.4	0
GTSoff,noTemporal,rate=22	1134.56	0
GTSoff,noTemporal,rate=24	1235.92	0
GTSoff,noTemporal,rate=26	1335.4	0
GTSoff,noTemporal,rate=28	1428.28	0
GTSoff,noTemporal,rate=30	1518.24	0
GTSon,General,rate=14	90.68	676.24
GTSon,General,rate=16	88.92	781.48
GTSon,General,rate=18	87.68	885.12
GTSon,General,rate=20	87.68	990.12
GTSon,General,rate=22	88.08	1089.44
GTSon,General,rate=24	88.84	1189.24
GTSon,General,rate=26	88.68	1286.92
GTSon,General,rate=28	89.88	1381.08
GTSon,General,rate=30	88.92	1445.88
GTSon,noTemporal,rate=14	88.4	618.12
GTSon,noTemporal,rate=16	88.2	716.8
GTSon,noTemporal,rate=18	88.12	816.8
GTSon,noTemporal,rate=20	88.76	916.84
GTSon,noTemporal,rate=22	87.88	1016.24
GTSon,noTemporal,rate=24	88.2	1116.16
GTSon,noTemporal,rate=26	88.92	1215.84
GTSon,noTemporal,rate=28	88.44	1316.6
GTSon,noTemporal,rate=30	88.2	1415.56

Figure 32: Packet TX state tabulated output.

The stacked chart graphically provides this information. We note that the number of transmitted packets at the MAC layer is considerably higher than at the application layer.

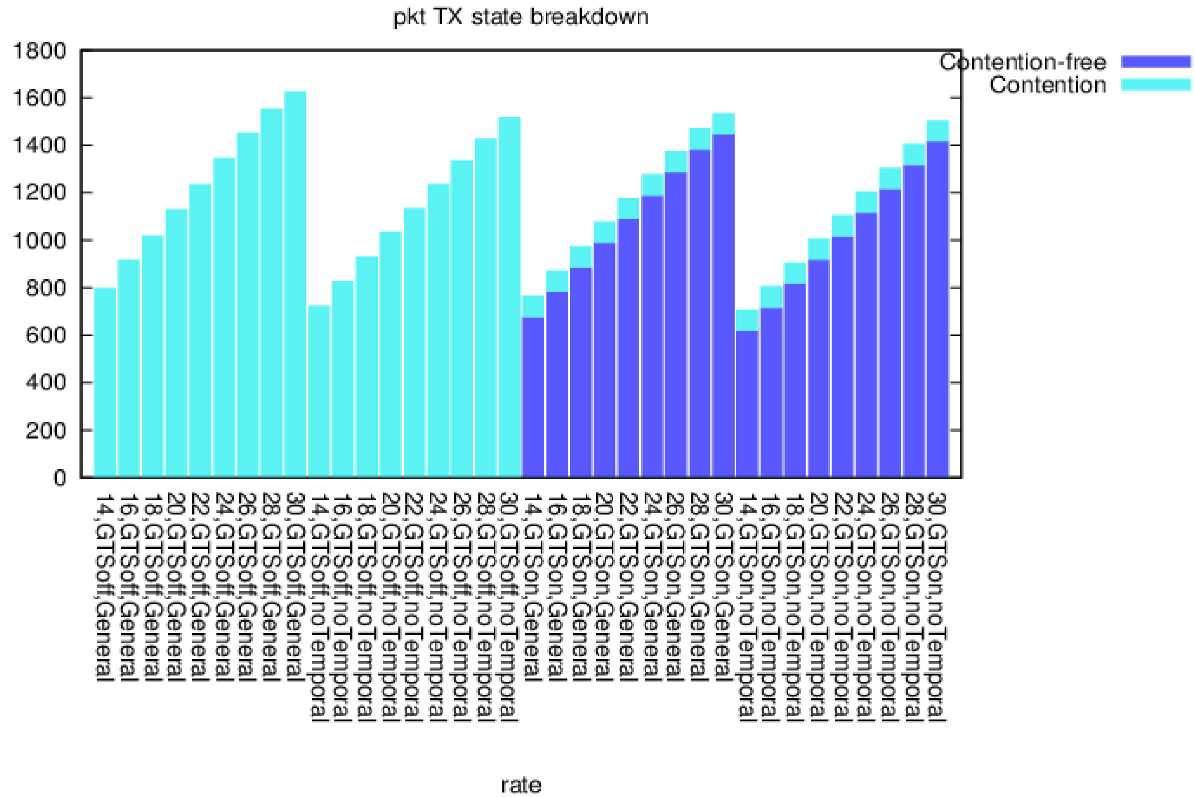

Figure 33: Transmitted packets breakdown at the MAC

4.2.3 Packet breakdown at radio layer

The packet numbers have changes at the radio layer as expected. We note that with GTSon the transmitted packets increase considerably in numbers, a reflection of the fact that there is no contention and nodes transmit without having to wait. A converse of this is the high number of failed packets during reception due to non RX state, reflecting the fact that node 0 only receives at given times. These results reflect as shown in the tabulated and graphed formats;

Communication.Radio:TXed pkts - TX pkts

	node=0	node=1	node=2	node=3	node=4	node=5
GTSoff,General,rate=14	3699.6	800.8	718.2	848	780.6	858.2
GTSoff,General,rate=16	4127.4	929	817.8	956.6	911.8	976
GTSoff,General,rate=18	4613.4	1022.8	917.8	1051.8	1031	1082.6
GTSoff,General,rate=20	5081.4	1130.8	1021	1179.2	1124.2	1205.6
GTSoff,General,rate=22	5519.2	1233.6	1117	1278	1231.2	1324.4
GTSoff,General,rate=24	5937.8	1343.6	1223.6	1403.4	1333.4	1433.2
GTSoff,General,rate=26	6320.6	1436.4	1311.4	1520.6	1447.6	1554.2
GTSoff,General,rate=28	6659.8	1562.4	1395.2	1611	1550.2	1657.8
GTSoff,General,rate=30	6937.2	1635.4	1482.8	1644.2	1622.6	1753
GTSoff,noTemporal,rate=14	3904.6	729	698.4	734.8	728.4	731.8
GTSoff,noTemporal,rate=16	4396.2	834.2	798	837	833.8	836.8
GTSoff,noTemporal,rate=18	4883	940	898	937.8	941	944
GTSoff,noTemporal,rate=20	5355.6	1043	996.2	1047.6	1047.8	1047.8
GTSoff,noTemporal,rate=22	5836.4	1142.8	1093	1151.2	1146	1145.8
GTSoff,noTemporal,rate=24	6295	1248	1187.6	1245.2	1250.8	1253
GTSoff,noTemporal,rate=26	6749.4	1351.4	1282.6	1350.6	1343.8	1353.8
GTSoff,noTemporal,rate=28	7173.2	1442.2	1374.4	1441	1445.6	1443.6
GTSoff,noTemporal,rate=30	7545.6	1534.6	1453.6	1531.4	1541.4	1535.8
GTSon,General,rate=14	3780.8	763.2	711.8	814.8	764	796.6
GTSon,General,rate=16	4287.6	871.6	806.6	913.6	859.8	911.8
GTSon,General,rate=18	4777	982.8	908	1008	965.8	1018.6
GTSon,General,rate=20	5266.8	1094.2	1005.8	1144.6	1069.4	1109.6
GTSon,General,rate=22	5748.4	1187.6	1111.2	1232	1160	1211.6
GTSon,General,rate=24	6269.2	1291.8	1213.2	1301.8	1276.6	1335.6
GTSon,General,rate=26	6740.8	1397.6	1309	1432.8	1367.2	1419.6
GTSon,General,rate=28	7203.2	1486.8	1409.4	1494	1467.2	1512
GTSon,General,rate=30	7551.6	1575	1507.4	1510	1558.2	1547
GTSon,noTemporal,rate=14	3925.6	709.8	700	711.4	711.4	711.8
GTSon,noTemporal,rate=16	4424.2	809.6	799.8	807.4	809.8	808.6
GTSon,noTemporal,rate=18	4923.2	908.4	899.6	907.8	910.2	909
GTSon,noTemporal,rate=20	5422.2	1010.6	999.2	1008.4	1009.2	1011
GTSon,noTemporal,rate=22	5920	1107.2	1099.2	1108.6	1108.2	1108.4
GTSon,noTemporal,rate=24	6420.6	1211.2	1198.8	1206.4	1208	1207.4
GTSon,noTemporal,rate=26	6918	1310.4	1298.4	1307.2	1308.6	1310
GTSon,noTemporal,rate=28	7418.8	1409.8	1398.6	1410	1410.6	1408
GTSon,noTemporal,rate=30	7917.6	1507.4	1498.6	1508.4	1505.8	1509.2

Figure 34: Transmitted packets at the radio layer

Communication.Radio:RX pkt breakdown

	Failed with interference	Failed, below sensitivity	Failed, non RX state	Received despite interference	Received with NO interference
GTSoff,General,rate=14,node=0	292.6	356.8	33.8	119.8	3154.8
GTSoff,General,rate=14,node=1	233.2	341.6	368	71.6	5838.2
GTSoff,General,rate=14,node=2	237	249.4	80.4	99	6283
GTSoff,General,rate=14,node=3	189	419.4	573.6	83.2	5503.2
GTSoff,General,rate=14,node=4	217	348.2	331.2	82.6	5883.4
GTSoff,General,rate=14,node=5	144.4	638.2	543.6	32.8	5369.4
GTSoff,General,rate=16,node=0	342.6	446.6	36.6	130	3572.4
GTSoff,General,rate=16,node=1	251	393.2	485.4	84	6497.4
GTSoff,General,rate=16,node=2	275.4	271.8	96.6	115.2	7093.8
GTSoff,General,rate=16,node=3	233.8	432.2	661.4	81	6268.8
GTSoff,General,rate=16,node=4	244.6	416.2	396.4	88.8	6599
GTSoff,General,rate=16,node=5	149	700	647.6	37.8	6076.4
GTSoff,General,rate=18,node=0	392	432.6	38.2	152	4036.4
GTSoff,General,rate=18,node=1	321.8	401.2	478.4	91.6	7338.4
GTSoff,General,rate=18,node=2	310.4	341.8	115.2	130.4	7853.2
GTSoff,General,rate=18,node=3	270.8	480.2	797.2	109.2	

Figure 35: Received packets breakdown at the radio layer

The graphs also collaborate these findings;

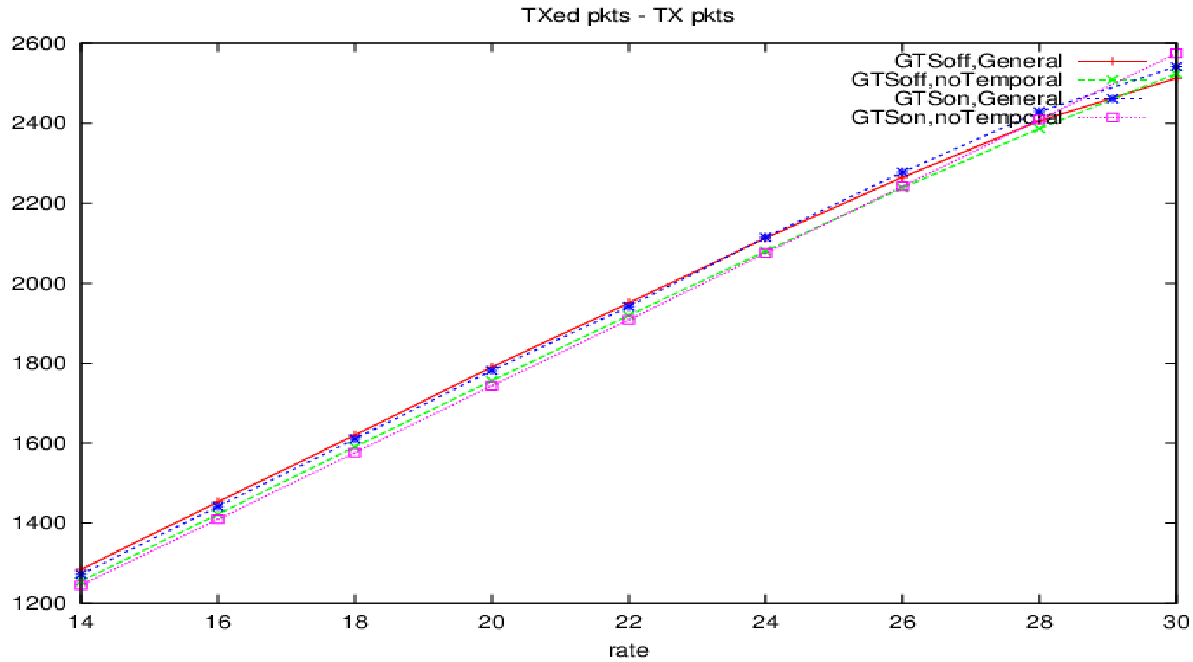

Figure 36: Transmitted packets against packet rate at the radio layer

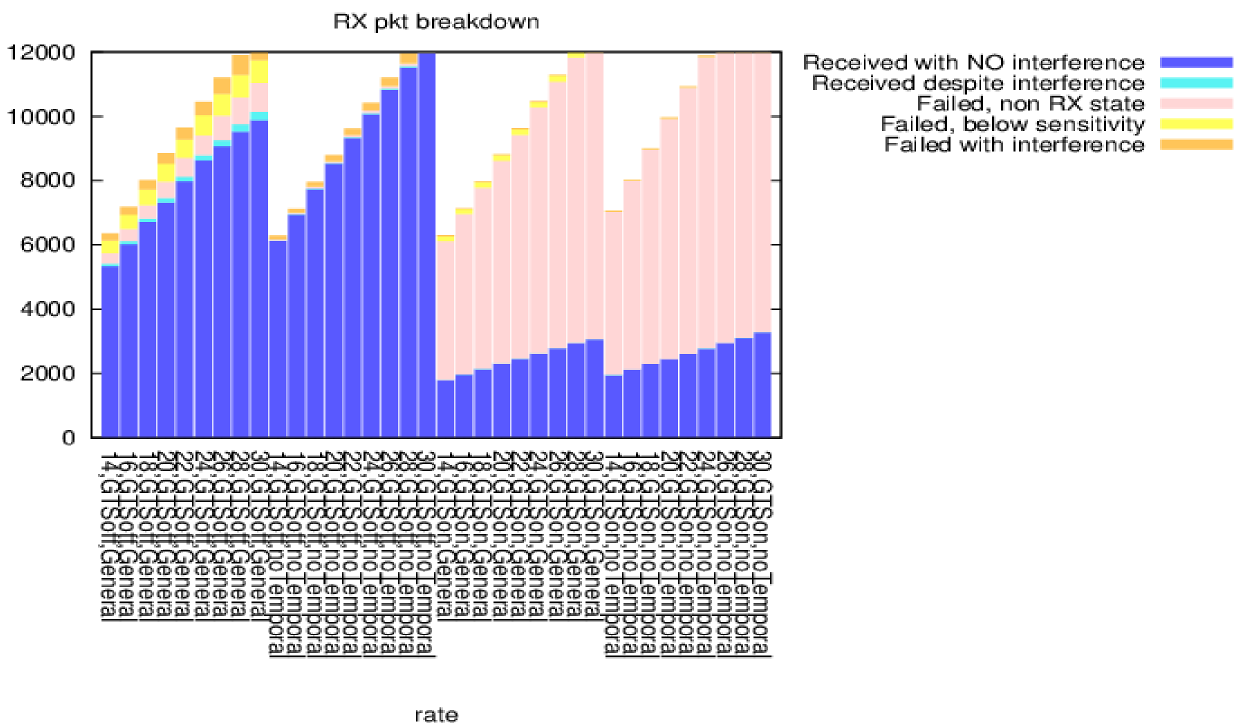

Figure 37: Received packets breakdown at the radio layer

4.3 Results on latency

The latency measurements occur at the application layer. The values are given in time buckets and are best represented in a histogram. As an example, we could choose the latencies at our rate of 22 packets/sec/node as a good representation, and generate the histograms for both cases of GTSon and GTSoff. Tabulated results for latency appear as follows;

```
Application:Application level latency, in ms
+---------------------------+--------+---------+---------+---------+----------+----------+----------+----------+----------+------
-----+----------+----------+----------+----------+----------+----------+----------+----------+----------+----------+----------+-------
----+----------+----------+----------+----------+----------+----------+----------+----------+
|                           | [0,20) | [20,40) | [40,60) | [60,80) | [80,100) | [100,120) | [120,140) | [140,160) | [160,180) | [180,200) | [200,
220) | [220,240) | [240,260) | [260,280) | [280,300) | [300,320) | [320,340) | [340,360) | [360,380) | [380,400) | [400,420) | [420,440) | [440,4
60) | [460,480) | [480,500) | [500,520) | [520,540) | [540,560) | [560,580) | [580,600) | [600,inf) |
+---------------------------+--------+---------+---------+---------+----------+----------+----------+----------+----------+------
-----+----------+----------+----------+----------+----------+----------+----------+----------+----------+----------+----------+-------
----+----------+----------+----------+----------+----------+----------+----------+----------+
| GTSoff,General,rate=14    | 942.4  | 607.2   | 508.4   | 480.4   | 428.2    | 48.8     | 29.8     | 28.6     | 23.6     | 25.8     | 26.4
    | 6        | 6.6      | 8.2      | 4.8      | 5.8      | 5.8      | 1.8      | 0.8      | 2.8      | 2.2      | 0.2      | 1.8
   | 1        | 0.4      | 0.2      | 0.4      | 0.6      | 0        | 0.2      | 2.4      |
| GTSoff,General,rate=16    | 1030   | 652.4   | 601     | 557     | 490.4    | 70.4     | 33       | 33.4     | 32.6     | 34.2     | 24.2
    | 10.8     | 8.4      | 7.4      | 8.2      | 7.4      | 7        | 2.4      | 1        | 2.4      | 1.8      | 1        | 1.4
   | 0.6      | 0.2      | 1        | 0.4      | 0.2      | 0.8      | 0        | 1.2      |
| GTSoff,General,rate=18    | 1141.2 | 659.8   | 761.6   | 619     | 563.6    | 98.8     | 37       | 39.6     | 38.4     | 38.2     | 31
    | 12       | 11.4     | 9.4      | 9        | 8.8      | 7.2      | 3.2      | 3.4      | 2.2      | 2.2      | 2.8      | 1.4
   | 1        | 0.8      | 1        | 0.4      | 0        | 0.8      | 0.2      | 1        |
| GTSoff,General,rate=20    | 1199.4 | 718.4   | 815.6   | 720.8   | 609.2    | 125      | 54.8     | 49       | 55.2     | 43.8     | 39.4
    | 18.8     | 13.8     | 14.6     | 11.4     | 12.2     | 9.8      | 7.8      | 3.4      | 6.2      | 3.2      | 4.6      | 3.8
   | 0.8      | 3.2      | 1        | 2        | 0.8      | 1        | 1        | 2.6      |
| GTSoff,General,rate=22    | 1243   | 795.2   | 878.6   | 803.4   | 669.2    | 149.4    | 80.2     | 57.6     | 65.8     | 49.2     | 51
    | 21.2     | 22.4     | 16.6     | 20.6     | 14.2     | 13.8     | 6.4      | 6.6      | 5        | 5.4      | 3.8      | 4.2
   | 1.8      | 1.8      | 1.4      | 1.8      | 1        | 0.8      | 0.8      | 6.2      |
| GTSoff,General,rate=24    | 1188.4 | 915.2   | 876.4   | 903.8   | 726      | 190.2    | 98.2     | 81.8     | 67.6     | 66       | 54
    | 28.8     | 26.8     | 22       | 22.2     | 19.8     | 17.8     | 13       | 10.2     | 9.4      | 8.8      | 8.2      | 6.4
   | 6.6      | 4.4      | 3.8      | 3.6      | 3        | 2        | 2.4      | 16.8     |
| GTSoff,General,rate=26    | 1052.6 | 882     | 859.4   | 970.4   | 752.4    | 306      | 170.8    | 127.2    | 110.8    | 93.6     | 80.4
    | 56.6     | 45.2     | 40.8     | 41.6     | 31       | 31.2     | 24       | 17.4     | 14.6     | 12.2     | 10.2     | 8.8
   | 6.4      | 5.6      | 6.2      | 4.8      | 3.6      | 3        | 2.8      | 15.8     |
| GTSoff,General,rate=28    | 818.2  | 731.8   | 803.4   | 909.8   | 773.6    | 389.4    | 254.6    | 204.4    | 167.6    | 143.6    | 119.2
    | 80       | 77.4     | 64.4     | 62.6     | 54.6     | 50.4     | 46.8     | 34.2     | 36       | 30.8     | 25.2     | 27.2
   | 19.8     | 22       | 18.2     | 15       | 16.6     | 11.4     | 12       | 116.2    |
| GTSoff,General,rate=30    | 429.6  | 438.8   | 495.2   | 600     | 580.4    | 389      | 299      | 267      | 229      | 196      | 173.6
    | 145.2    | 123.4    | 117      | 99.4     | 93       | 88.4     | 85.8     | 82.6     | 78.2     | 76       | 71.6     | 61.4
   | 59.2     | 58.4     | 49.2     | 53.6     | 50.8     | 44.6     | 46.8     | 828.2    |
| GTSoff,noTemporal,rate=14 | 1090   | 713     | 577.2   | 569.4   | 491      | 34       | 0        | 0        | 0        | 0        | 0
```

Figure 38: Latency results output

As evident, tabulated values are hard to follow, thus necessitating a graphical representation. We start with the case of GTS off;

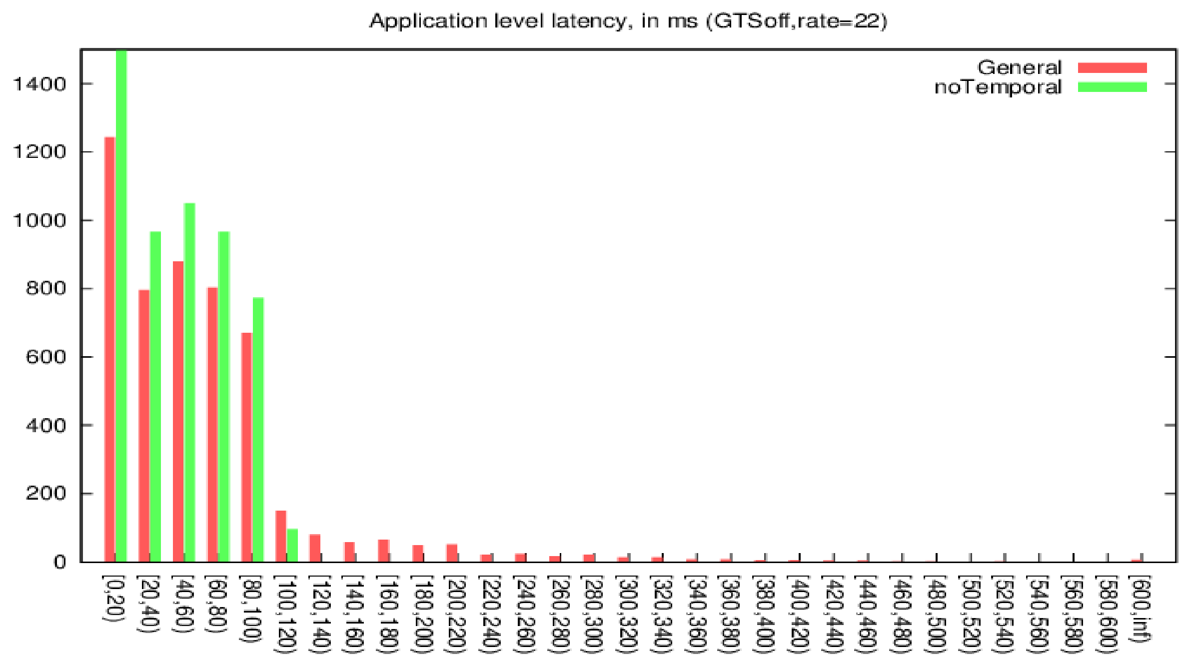

Figure 39: Application level latency (GTSoff, rate=22)

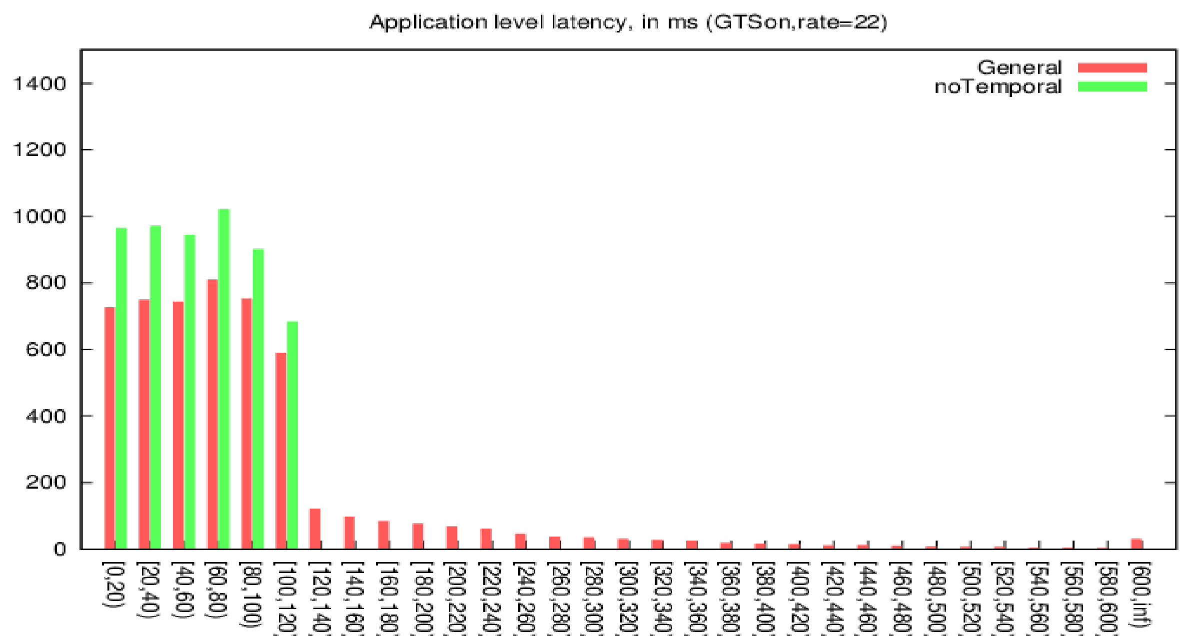

Figure 40: Application level latency (GTSon, rate=22)

In both cases most packets are well under 100ms, implying that they are mostly transmitted within the first MAC frame.

4.4 Results on energy consumption

Energy usage is reported at the application layer and by the dedicated Castalia module of resource manager. The Resource Manager tracks energy consumption at the node while application gives an estimation in nJ/bit. Their respective tabular and graphical representations are as shown here.

4.4.1 Energy consumption Results;

Combined tabular representations;

Application:Energy nJ/bit

	rate=14	rate=16	rate=18	rate=20	rate=22	rate=24	rate=26	rate=28	rate=30
GTSoff,General	76.61278	67.25036	59.41052	53.5233	48.42465	44.96974	41.87246	39.57055	37.82749
GTSoff,noTemporal	73.71291	64.60931	57.46618	51.88198	47.3041	43.55423	40.44083	37.85364	35.86444
GTSon,General	25.55446	21.91296	19.78204	17.58453	15.88848	14.43556	13.41541	12.3203	11.90929
GTSon,noTemporal	24.25168	21.24935	18.77819	16.8269	15.34701	14.00589	12.96664	11.97087	11.16785

ResourceManager:Consumed Energy

	rate=14	rate=16	rate=18	rate=20	rate=22	rate=24	rate=26	rate=28	rate=30
GTSoff,General	0.06059	0.06038	0.06044	0.06039	0.06018	0.06029	0.06021	0.06026	0.06017
GTSoff,noTemporal	0.06218	0.06221	0.06215	0.0621	0.06213	0.06208	0.0621	0.06204	0.06203
GTSon,General	0.03803	0.03788	0.03808	0.03796	0.03794	0.0379	0.03796	0.03781	0.03793
GTSon,noTemporal	0.03818	0.0382	0.03812	0.03807	0.03811	0.03805	0.03809	0.03801	0.038

ResourceManager:Remaining Energy

	rate=14	rate=16	rate=18	rate=20	rate=22	rate=24	rate=26	rate=28	rate=30
GTSoff,General	18719.96667	18719.96667	18719.96667	18719.96667	18719.96667	18719.96667	18719.96667	18719.96667	18719.96667
GTSoff,noTemporal	18719.96667	18719.96667	18719.96667	18719.96667	18719.96667	18719.96667	18719.96667	18719.96667	18719.96667
GTSon,General	18719.96667	18719.96667	18719.96667	18719.96667	18719.96667	18719.96667	18719.96667	18719.96667	18719.96667
GTSon,noTemporal	18719.96667	18719.96667	18719.96667	18719.96667	18719.96667	18719.96667	18719.96667	18719.96667	18719.96667

smnk@Falcon:~/Project/Castalia/Castalia-3.3/Castalia/Simulations/BANtest$

Figure 41: Energy consumption results

The histograms for energy consumption can be presented as presented by CastaliaResults – at the application and by Resource manager;

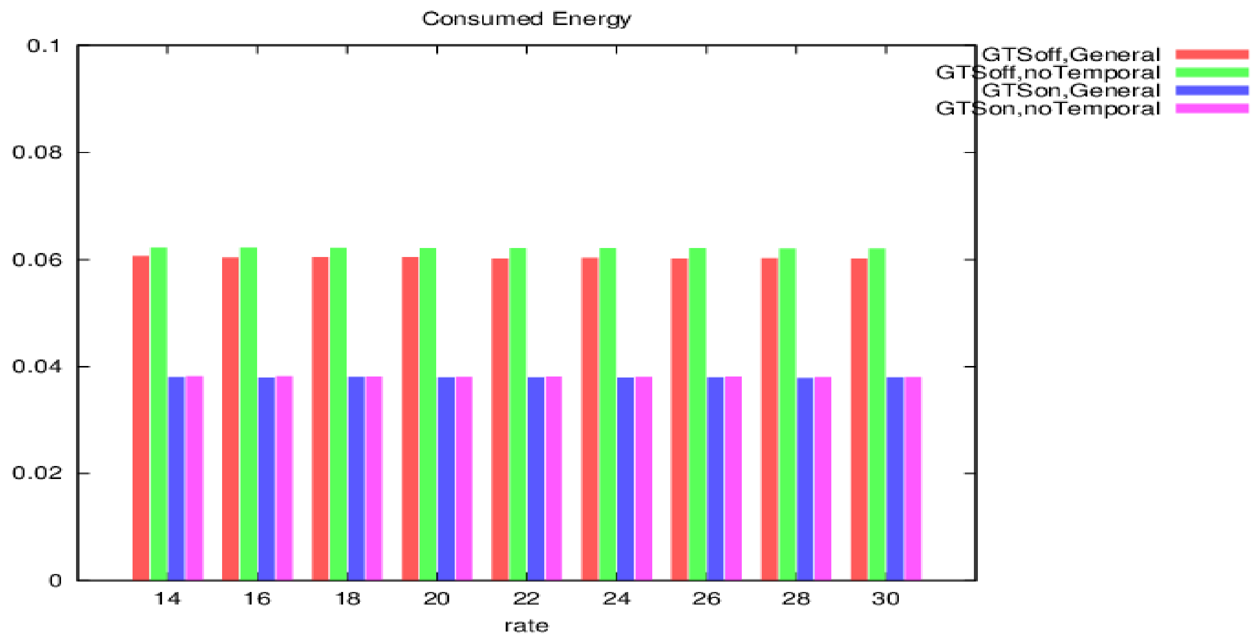

Figure 42: Consumed energy from Resource Manager

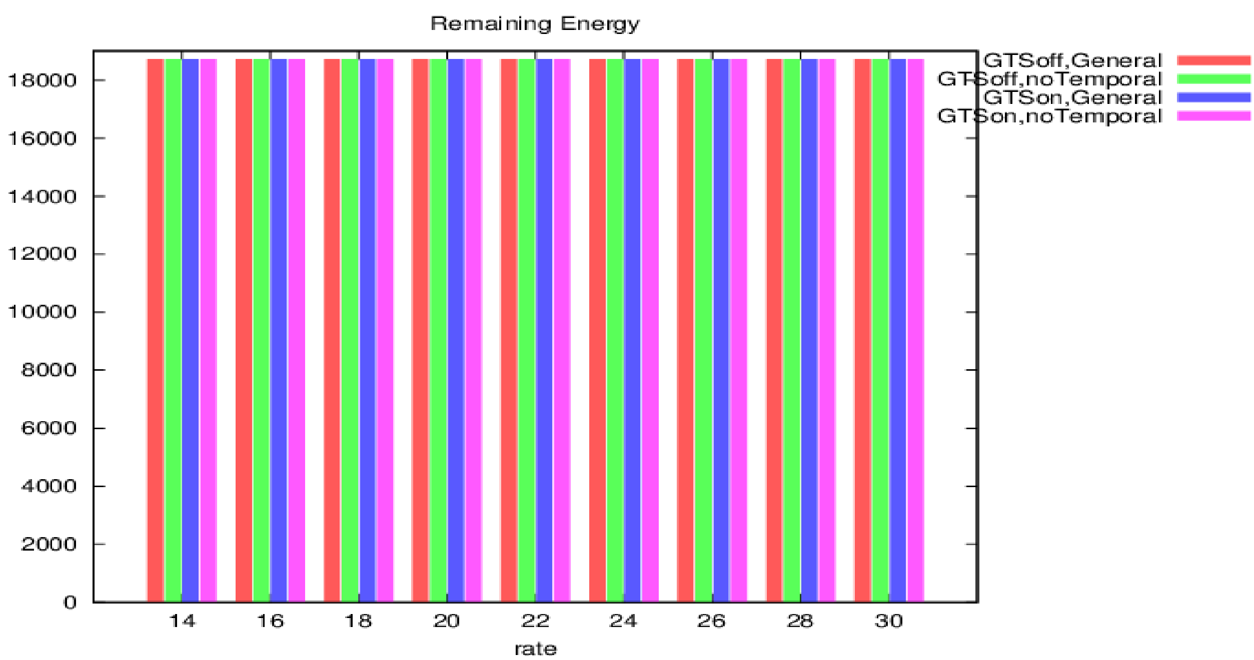

Figure 43: Remaining energy from Resource Manager

4.5 Wireless channel fade characteristic

The wireless channel also reports on the fade depth distribution. This is useful while interpreting results concerning the presence of temporal variations. The tabulated and graphical results are as below;

wirelessChannel:Fade depth distribution

	[-50,-45)	[-45,-40)	[-40,-35)	[-35,-30)	[-30,-25)	[-25,-20)	[-20,-15)	[-15,-10)	[-10,-5)	[-5,0)	[0,5)	[5,10)	[10,15)	[15,inf)
GTSoff,General,rate=14	13	29.4	79.6	184.2	413.4	1021.8	2186	4590.4	8998	13666	11713	3194	132.6	1
GTSoff,General,rate=16	20	28.4	76.2	220.4	451	1146.6	2486.6	5220	10170.8	15492.4	13141.2	3700.2	143.2	0
GTSoff,General,rate=18	24.6	40.8	89.8	231.4	526	1272	2807.8	5742.6	11222.8	17121.2	15092.6	3997.2	132.8	1
GTSoff,General,rate=20	29.8	52.6	111.8	261.2	598.4	1391	3148	6666.8	12748.2	18729.8	16061.2	4468	169	1
GTSoff,General,rate=22	28	49.6	114	284.8	640.2	1557.4	3282.2	7025.4	13279.4	20205	18380.2	5166.4	191.6	3
GTSoff,General,rate=24	20	57.8	107.2	301.6	703.6	1642.6	3616.4	7563.2	14585.4	22335	19434.6	5449.8	208.4	1
GTSoff,General,rate=26	21.6	56.6	118.4	295.8	793.4	1760	3879.6	7984.8	15875.8	23606.6	21098.2	5847.6	187	2
GTSoff,General,rate=28	27.2	63.2	142.4	346.4	808.6	1824.2	3942.2	8498.2	16270.2	25049.6	22706.8	6659.8	263	0
GTSoff,General,rate=30	29	61.6	133.8	298.4	804.6	1925	4236.8	9203.4	17252.4	26305	23445.4	6498	232.4	0
GTSon,General,rate=14	18.4	39.2	65.4	165.6	388.2	1000.8	2143	4513.2	8906.4	13567.8	11638.8	3204.6	122.4	1
GTSon,General,rate=16	21.8	40.4	67.2	194	459	1125.6	2444.2	5224.4	10124.2	15284.8	13085.6	3666	155.6	0
GTSon,General,rate=18	18.2	33.4	85.6	211.4	520.4	1230	2776	5693.6	10989.2	16998.6	15088	4139.6	157	1
GTSon,General,rate=20	24.8	34.2	104.4	244	574	1460.8	3104.6	6382.6	12405	18654.2	16272.2	4669.6	191.2	1
GTSon,General,rate=22	19.6	41	96.4	260.4	583	1442.8	3146.6	6744	13365.6	20488.6	18334.2	5158.8	203.2	3
GTSon,General,rate=24	30.8	49	118	269.8	609.6	1568.6	3424.4	7457.6	14571.4	22411	19703.8	5663.2	230	1

Figure 44: Fade depth distribution

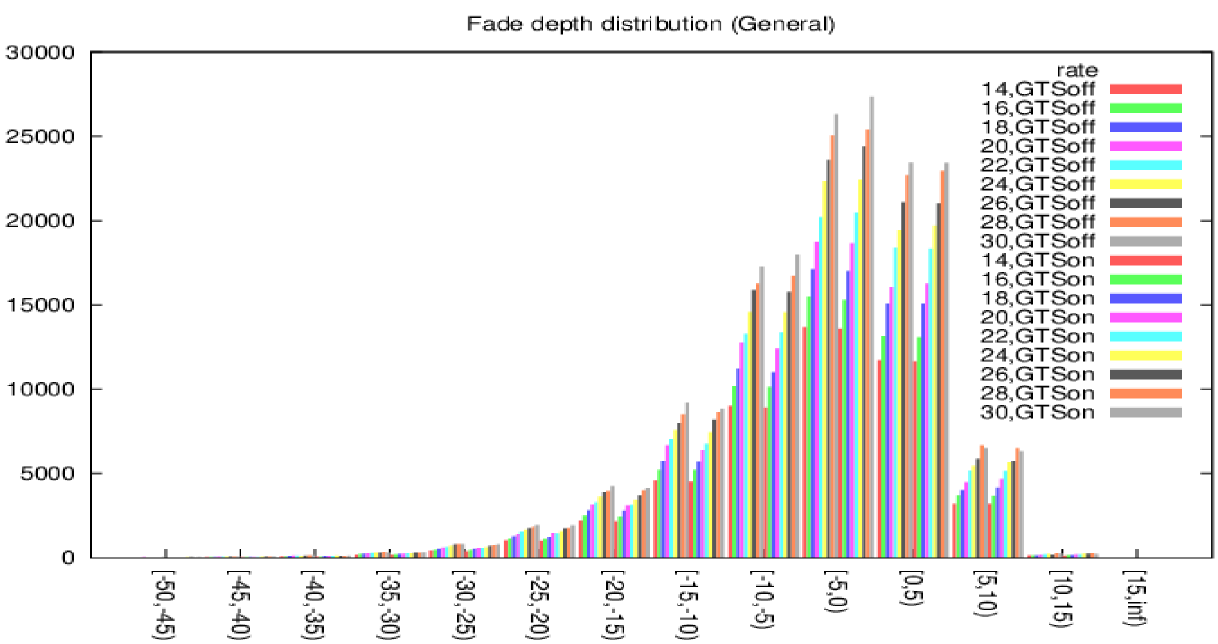

Figure 45: Fade depth distribution at different packet rates

4.6 Measured and Derived QoS Parameters

The Castalia simulation output produces the primary QoS metrics that are measurable for a given network set-up. In this work we take the assumption that any parameter that is directly measured in Castalia is a primary metric. From these other metrics can be derived. The required inputs for the various FLC created in section 3.3.2 and their values are either primary or derived metrics. This division is shown in Table 12 below;

QoS Characteristic	**QoS Metric**	**Primary or Derived**
Timeliness	Latency	Primary
	Response Time	Derived
	Jitter	Derived
Network Lifetime	Consumed Energy	Primary
	Remaining Energy	Primary
	Estimated Lifetime	Derived
Reliability	Packet Loss Ratio	Primary
	Bit Error Rate	Derived
	Packet Error Rate	Derived

Table 15: QoS parameters derivation

All the derived metrics require a calculation supported by observation and measurement. For the metrics above, Estimated Lifetime and Packet Error Rate and Packet Loss Ratio are already calculated by Castalia. For the rest their formulae are as below:

(i) Response time = Time T_1 when last association request is sent – Time T_2 when first association request is received from the PAN coordinator.

(ii) Jitter is the absolute value of the variations in delays between two consecutive packets for a given traffic flow. This means the absolute value in the difference between the latencies of two consecutive packets.

Therefore Jitter = $|D_1\text{-}D_2|$ where D_1 the latency for packet1 and D_2 is the latency for packet 2.

(iii) The Packet Error Rate (PER) is the number of incorrectly transferred data packets and can be relate to BER by this formula;

PER = 1 - (1 - BER)^PL (4)

where PL is lengh of the packet (header + payload)

From our simulation we assume this means incorrectly transferred data packets will be those received despite interference.

(iv) Bit Error Rate = Number of Errors/Total Number of bits sent. In our simulation we assume that all bits interfered with are in error thus not received.

4.6.1 Measured Simulation QoS values

Thus from our simulation, if we take results for a case of packet transmission at our chosen trial rate of 22 with all scenarios, the range of all inputs can be tabulated as follows:

QoS Characteristic	**QoS Metric**	**Formula**	**Derivation**	**Value with GTSoff and No temporal**	**Value with GTSoff and Temporal on**	**Value with GTSon and No temporal**	**Value with GTSon and temporal on**
Timeliness	Latency	Average of largest values per node.	From Castalia latency	24.90ms	25ms	54.5ms	**39ms**
	Response Time		From first PAN beacon sent to last node association accept				**0.121794s**
	Jitter		Average of latency for packets sent from node 3				**5.561ms**
Network Lifetime	Consumed Energy	Taken for Node 0 the sink	From Castalia	0.0621	0.06039	0.03811	**0.03794**
	Remaining Energy	Taken for Node 0 the sink	From Castalia	18719.967J	18719.967J	18719.967J	**18719.967J**
	Estimated Lifetime		From Castalia	69.9354 Days	69.9354	69.9354	**69.9354**
Reliability	Packet Loss Ratio		From Castalia	0.018	0.09	0.002	**0.056**
	Bit Error Rate		From Castalia	2.88141E-05	9.92205E-05	3.81451E-06	**7.65832E-06**
	Packet Error Rate		From Castalia	0.056	0.18	0.0076	**0.0152**

Table 16: Simulation results for objective QoS

4.7 Fuzzy Logic System Results.

Once the inputs for QoS parameters in section 4.6 are obtained as outlined in Table 13, they are fed into the FLC systems developed earlier. For simplicity, we shall take the realistic case of

GTSon with temporal variations. The defuzzified outputs for Timeliness, Network Lifetime and Reliability were then obtained as shown in the subsequent sections.

4.7.1 FLC system for Timeliness characteristic

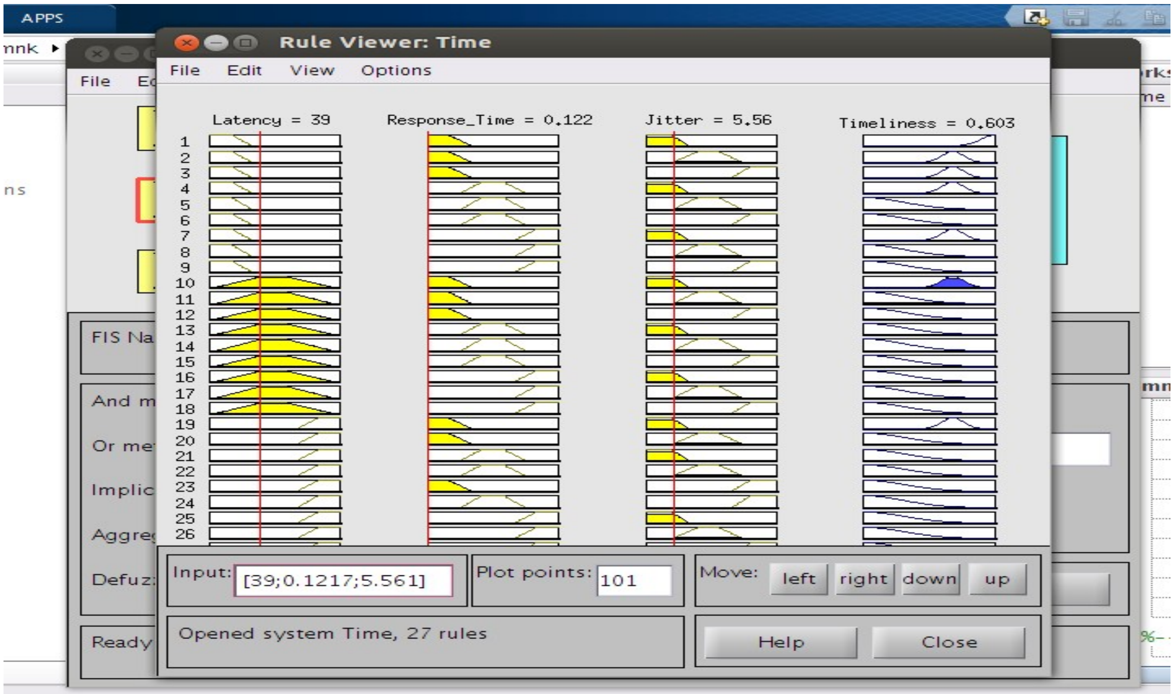

Figure 46: FLC deffuzifier for Timeliness - rules

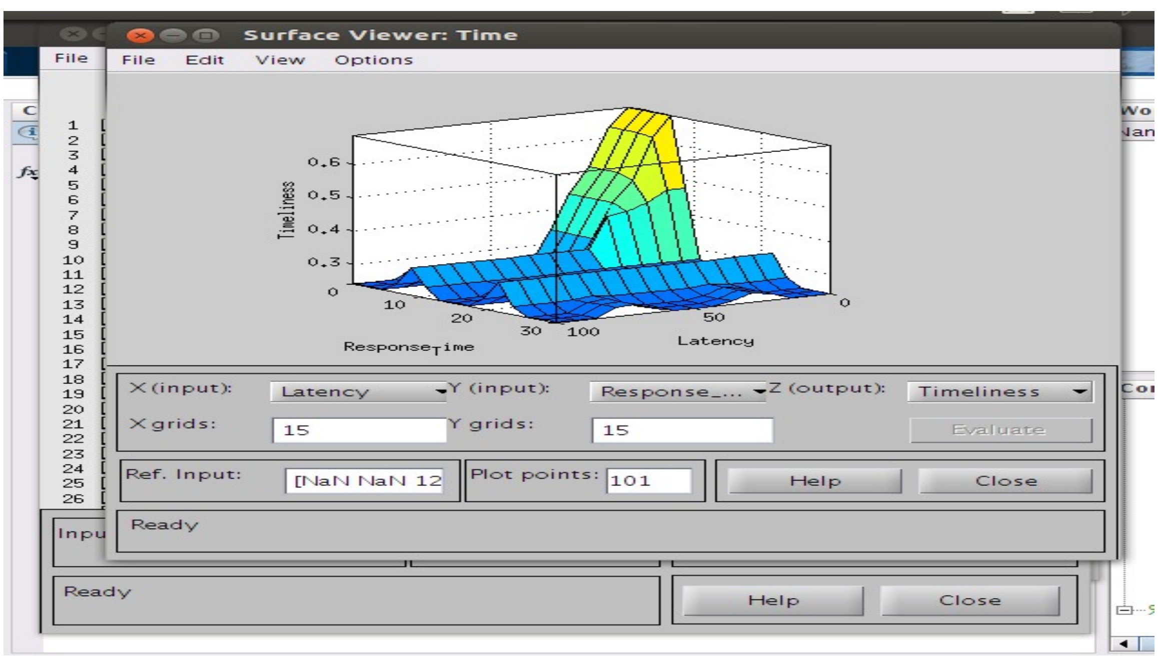

Figure 47: FLC deffuzifier for Timeliness -surface

4.7.2 FLC system for Network Lifetime characteristic

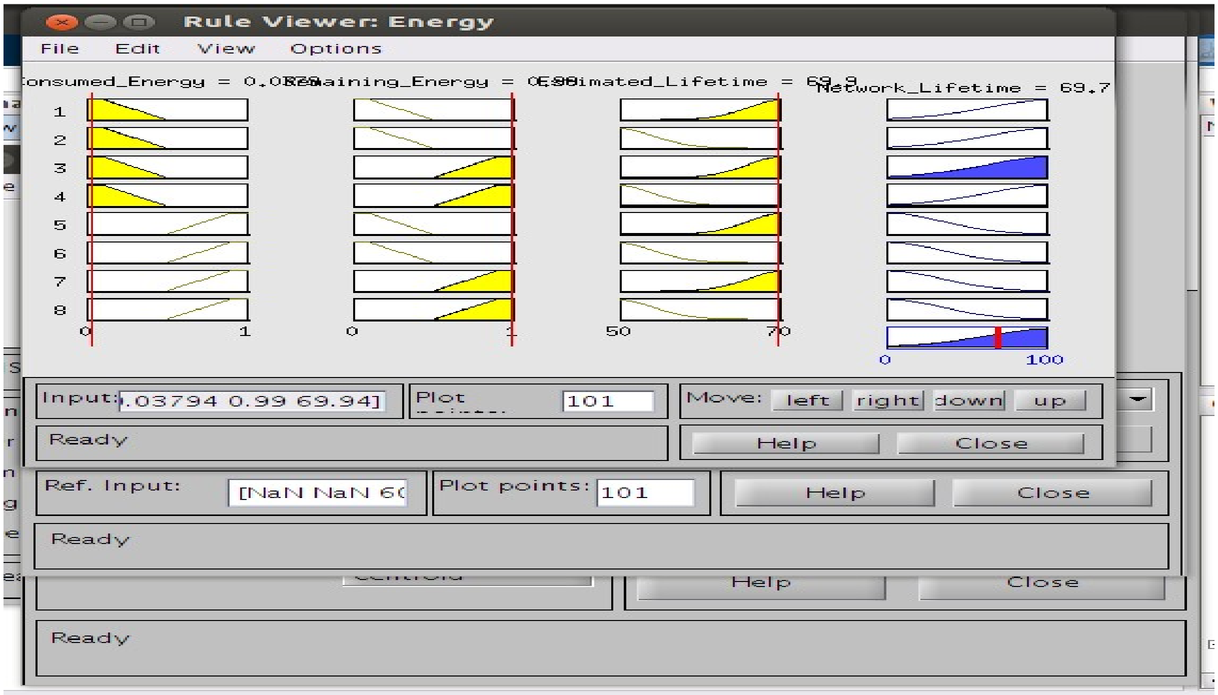

Figure 48: FLC deffuzifier for Network Lifetime – rules

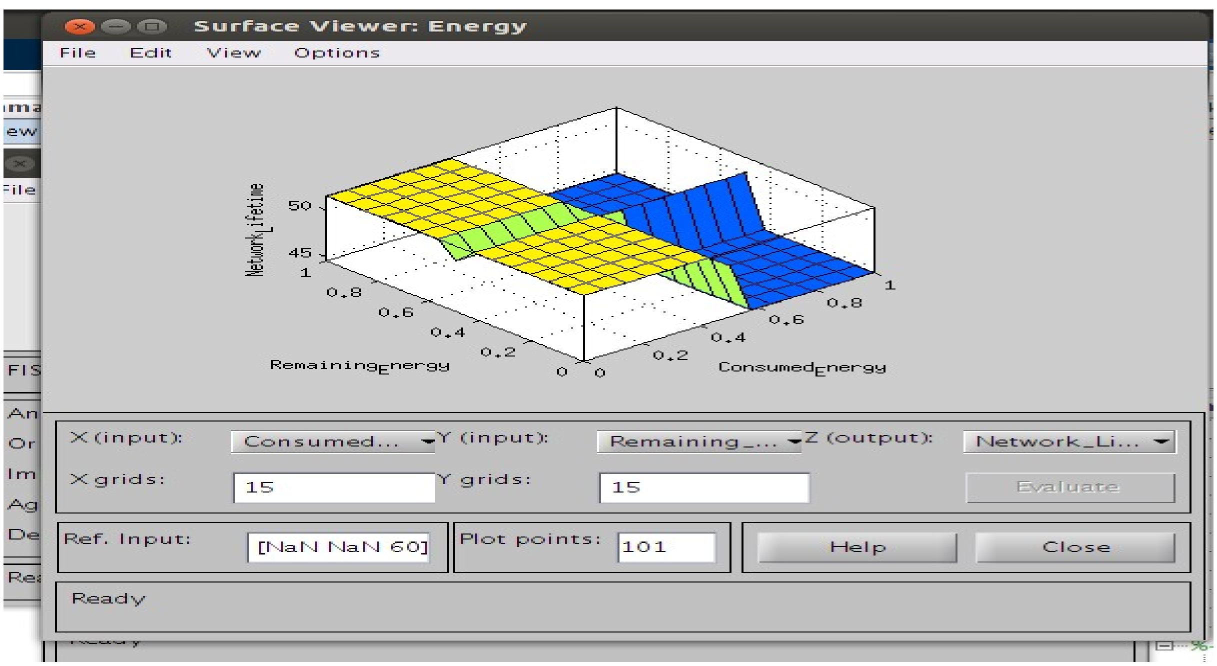

Figure 49: FLC deffuzifier for Network Lifetime - surface

4.7.3 FLC system for Reliability characteristic

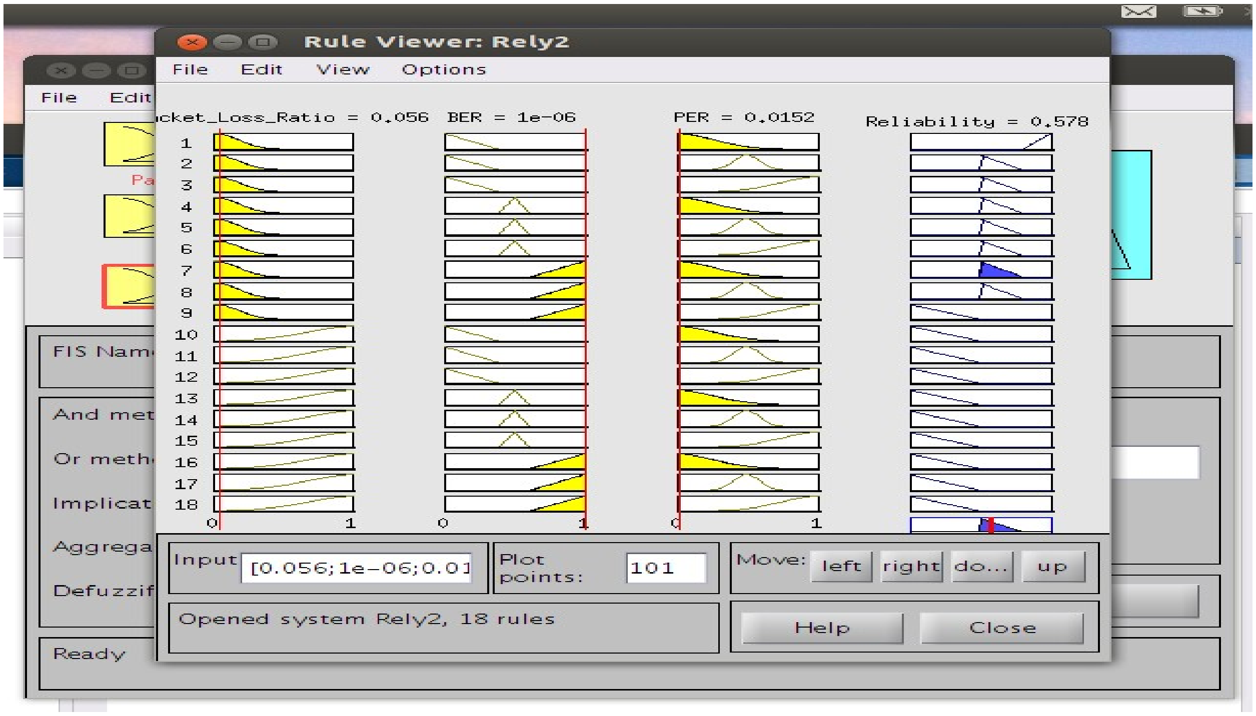

Figure 50: FLC deffuzifier for Reliability – rules

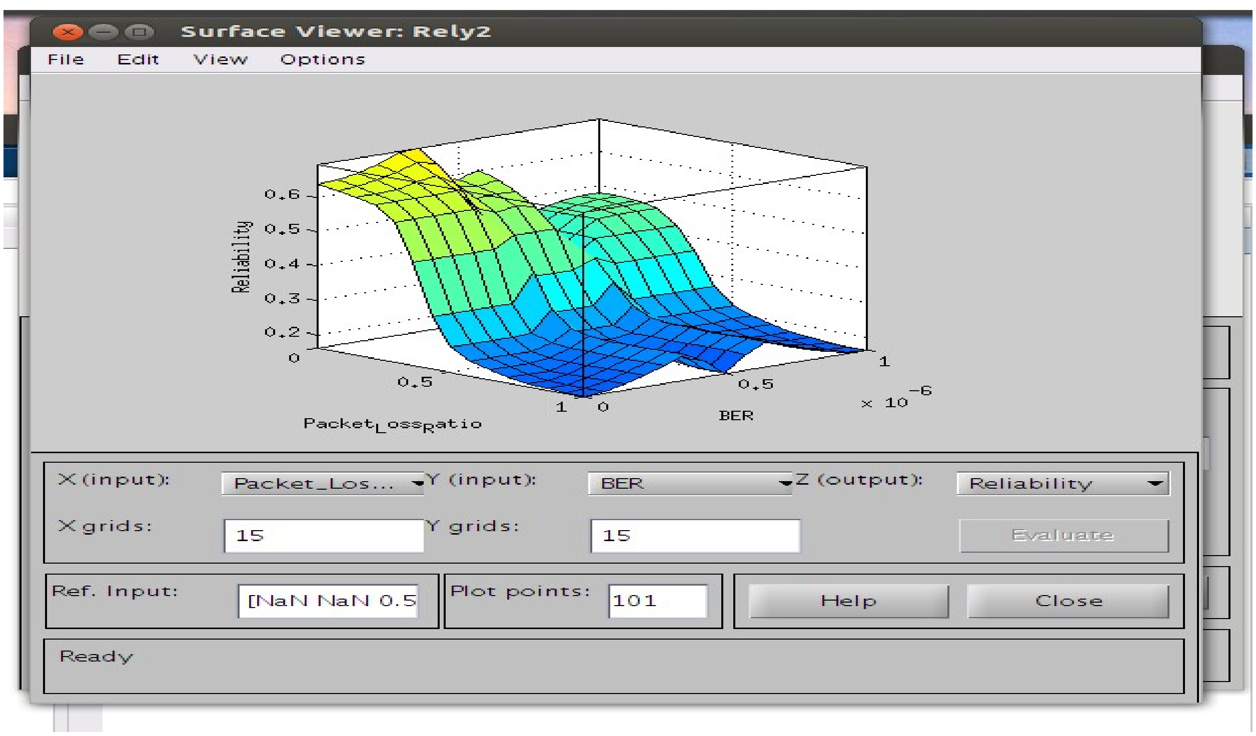

Figure 51: FLC deffuzifier for Reliability - surface

The results are tabulated for easier viewing as below;

QoS Characteristic	QoS Metric	Value	Deffuzified Value from FLC
Timeliness	Latency	39ms	0.603or 60.3%
	Response Time	0.121794s	
	Jitter	5.561ms	
Network Lifetime	Consumed Energy	0.03794	69.7days
	Remaining Energy	18719.967J	
	Estimated Lifetime	69.9354	
Reliability	Packet Loss Ratio	0.056	0.578 or 57.8%
	Bit Error Rate	7.65832E-06	
	Packet Error Rate	0.0152	

Table 17: Fuzzy Logic Results for QoS

4.8 Analytic Hierarchy Process

The AHP requires the Fuzzy QoS results to assist the decision maker in assignment of the importance weights. To formulate the pairwise comparisons, we use a relative scale to compare the QoS parameters against each other given that we want to transmit monitored temperature in an industrial park;

(i) Timeliness is at 68.6% and the network lifetime is 68.9 days out of a possible 70days. For the purpose at hand, timeliness should be far more important, currently it is just acceptable. We assign it a value of 7.

(ii) Compared to reliability, reliability is always much important but at 88.9% it is excellent thus not much weighty in this particular situation, thus we assign it a value of 3.

(iii) Comparing the network lifetime to reliability, we note both characteristics are at highly acceptable values even though reliability is more important than lifetime, thus we assign it a value of 3.

The comparison matrix for our scenario will therefore look as follows

	Timeliness	Lifetime	Reliability
Timeliness	1	7	1/3
Lifetime	1/7	1	1/3
Reliability	3	3	1

The next steps involve computing the weights for each criteria and alternatives. We also emphasize that our scenario is simplified for the purpose of demonstrating this stage of the evaluation model.

4.8.1 Calculating the criteria weights

With the comparison matrix in place, we now compute the priority vector, which is the normalized Eigen vector of the matrix. The method in use here is only an approximation of the Eigen vector (and Eigen value) of a reciprocal matrix which is enough for the task.

We sum each column of the reciprocal matrix to get

	Timeliness	Lifetime	Reliability
Timeliness	1	7	1/3
Lifetime	1/7	1	1/3
Reliability	3	3	1
	29/7	11	5/3

Then we divide each element of the matrix with the sum of its column, we will have the normalized relative weight. The sum of each column is now 1

	Timeliness	Lifetime	Reliability
Timeliness	7/29	7/11	1/5
Lifetime	1/29	1/11	1/5
Reliability	21/29	3/11	3
	1	1	1

The normalized principal Eigen vector is obtained by averaging across the rows

	Timeliness	Lifetime	Reliability	
Timeliness	7/29	7/11	1/5	1.077
Lifetime	1/29	1/11	1/5	0.325
Reliability	21/29	3/11	3	3.307
	1	1	1	

Thus, we can see that reliability followed by timeliness is the most highly ranked QoS characteristic. We now give priorities for the alternatives given what we know about the criteria ranking. This involves a procedure similar to the above for each of the alternatives.

Evaluating the alternatives with respect to Timeliness;

Timeliness (60.3% at 1.077)	Usable	Limited	Unusable
Usable	28/39	28/36	7/15
Limited	7/39	7/36	7/15
Unusable	4/39	1/36	1/15
	1	1	1

Normalizing the matrix

Timeliness (at 60.3% at 1.077)	Usable	Limited	Unusable	
Usable	28/39	28/36	7/15	1.96239
Limited	7/39	7/36	7/15	0.840598
Unusable	4/39	1/36	1/15	0.197008547
	1	1	1	

From the computation, the timeliness is usable, the highest ranking alternative.

Evaluating the alternatives with respect to Network Lifetime;

Network Lifetime (at 69.7 days and 0.325)	Usable	Limited	Unusable
Usable	1	6	9
Limited	1/6	1	7
Unusable	1/9	1/7	1
	70/54	50/7	17

Normalizing the matrix;

Network Lifetime (at 69.7 days)	Usable	Limited	Unusable	
Usable	54/70	42/50	9/17	2.14084
Limited	9/70	7/50	7/17	0.68034
Unusable	6/70	1/50	1/17	0.16454
	1	1	1	

Network Lifetime is also usable in the alternative weighting since usable has the highest weight

Evaluating the alternatives with respect to Reliability

Reliability (at 57.8% and 3.307)	Usable	Limited	Unusable
Usable	1	3	5
Limited	1/3	1	7
Unusable	1/5	1/7	1
	23/15	29/7	13

Normalizing the matrix;

Reliability (at 57.8% and 3.307)	Usable	Limited	Unusable	
Usable	15/23	21/29	5/13	0.3846
Limited	5/23	7/29	7/13	0.9972
Unusable	3/23	1/29	1/13	0.4982
	1	1	1	

Reliability is limited as the result from this matrix informs us. We know from the criteria ranking that reliability is a very important QoS requirement but has a sort of average performance for this particular application and therefore needs improvement. Tabulating these rankings in a simple format reveals that usable is the weightiest alternative.

	Usable	**Limited**	**Unusable**
Timeliness	1.96239	0.840598	0.197
Lifetime	2.14084	0.68034	0.16454
Reliability	0.3846	0.9972	0.4982
	4.48783	**2.518138**	**0.85974**

Table 18: AHP weights for QoS determination

This corresponds to an overall QoS of **good** for our Wireless Sensor Network. We note that although the timeliness and network lifetime are good, the reliability is limited, meaning that the BER, loss ratio and PER require to be monitored closely. This in turn might imply a poorly designed wireless channel, probably interference or such other causes.

4.9 Model Driven Scheme performance evaluation

This work has consistently kept alive the key concepts that form the foundations of the hybrid scheme; the Fuzzy Logic and the Analytic Hierarchy Process. It follows that a complete evaluation of the scheme would most suitably encompass comparisons with similar or close models and such constructions. There exists various research work that makes use of Fuzzy Logic for QoS evaluation, while AHP has also been incorporated in research for eventual QoS determination. In this respect, we will examine how our hybrid model driven scheme compares against each of two fuzzy logic based models, a model that employs fuzzy C means clustering and one that uses a AHP stage. These are (1), (10), (26) and (12)

4.9.1 Model Comparisons

A unique feature of our scheme is the incorporation of characterization to describe a given group of QoS metrics that describe a certain aspect of QoS. In (1), the QoS was determined using the measured values for Delay, Jitter (both in msec) and Loss as a percentage. The obtained values used the mean and standard deviation. There were nine rules used for deffuzzification. The final output is a single crisp value that gives the normalized overall QoS value of the application under consideration. The method is simple and does not require analytical and heavy mathematical models. However, this model has no aspect of any subjective assessment; it can be considered as the first component in our whole scheme.

In (26), a model has been devised for QoS evaluation in Physiological monitoring services within a M2M or Internet Of Things (IoT) setting with respect to the e-health domain. As in (1), three QoS parameters of Data Rate, Delay and Packet Losses are used to determine the QoS of Physiological monitoring service using a Fuzzy Logic system. This system consists of eighteen rules and a membership function for each parameter. The proposed method uses the Mamdani rules and the center of gravity for defuzzification. There is no aspect of subjective assessment, though the model is simple. Compared to our model driven scheme, it can be looked at as the first part of our scheme. However, we note here that this particular research was scant in actual implementation details.

The work (10) has used artificial intelligence and statistical methods to evaluate QoS of a hybrid wireless network. In the study, a network QoS evaluation system that used a combination of fuzzy C-means (FCM) and a regression model to analyze and assess the QoS in a simulated network was proposed. This network consisted of a wired LAN segment and a wireless IEEE802.11e segment. A defining feature was the choice of a FCM clustering algorithm whose robustness was well placed to deal with the imprecision of QoS patterns. The QoS parameters of delay, jitter and packet loss ratio were classified into three clusters. The regression model combined the centers generated from each cluster to produce a single QoS output that represented overall network QoS. This work approximates our hybrid model driven scheme in that it uses two stages each employing a well-known mathematical concept. However, it is more analytical in the FCM algorithm and involves heavy computational engagement in classifying the QoS clusters and statistically combining the obtained centers in a regression model. Furthermore, as in all previous models looked at, there is no aspect of subjective assessment and this is consistent with the main objective of the work which is to avoid network congestion.

The notion of the customer's perception of quality of service and its relation to customer satisfaction is briefly examined in (12). This work importantly marshals the critical tasks of identifying quality of experience (QoE) factors and the related QoS parameters, as well as constructing appropriate evaluation architecture, all in a subjective assessment domain. The research employs Fuzzy Analytic Hierarchy Process (FAHP) in the evaluation architecture as proposed, such that qualitative and quantificational methods are used to guarantee rationality and accuracy of the evaluation of service quality. The final outcome is a direct correlation between a customer satisfaction degree and the measured QoS parameters, such that if a Service Provider improves a given poor QoS measure, the satisfaction degree increases. In this research, descriptive characteristics have been elegantly utilised as Quality of Experience parameters. Thus, a mapping is done between layers represented by the customer satisfaction degree, a QoE layer and a QoS layer. This compares very well with the QoS characteristics adopted in our hybrid model driven scheme; in this research it was a technique to evolve from a high level customer perception to parametric QoS opinions, while in our research it was a way of smoothly transitioning between objective QoS assessment and subjective assessment.

CHAPTER FIVE: CONCLUSIONS AND RECOMMENDATIONS

5.1 Study Achievements

Revisiting our objectives, we note that this work has remarkably hewn to the motivation of the research thus far. The model driven evaluation scheme has taken shape in a simple and elegant manner. The scheme has also been demonstrated in the two stages of FLC and AHP successfully, therefore meeting our second objective in a satisfactory manner.

During the development and execution of the simulation, much went into review of various QoS evaluation models for different types of wireless networks. Several of the more conversant models are compared to our model driven scheme with a general conclusion emerging to the effect that our scheme is flexible and robust as desired. Furthermore, it fully accommodates subjective user assessment from an early stage. The flexibility is achieved through the introduction of QoS characteristics. These linguistic descriptors introduce intermediacy at a critical stage in the model, which allows the idea of user assessment to be introduced smoothly in the next AHP stage. This is the real defining feature and differentiator of our scheme.

The comparisons, apart from achieving our third objective, also place this study in context and help in revealing the wider scope envisioned in this kind of QoS evaluation in future. This lays the ground for recommendations on future work and possible opportunities in Machine 2 Machine quality of service monitoring and evaluation.

5.2 Study assumptions

In most of the literature, researchers who have simulated wireless sensor networks have left no doubt that simulation involves a lot of assumptions that mostly attempt to apply ideal situations, something that is never possible in reality. It is no different in this work, we have made various assumptions that are mentioned here for completeness;

(i) While dealing with QoS parameters, we have assumed that all parameters directly reported by Castalia are primary parameters, although strictly speaking some of these parameters are also derivations calculated internally by Castalia.

(ii) We have assumed that packets received despite interference make up the populations that give our BER and thus PER measurements. The reality is more complex with other unaccounted for reasons being causes

(iii) We have also generalized the environment in our factory set-up to only take a simply made temporal model that may not be an exact approximation of that environment.

5.3 Research Contributions

While reviewing literature, many facets of sensor networks were studied. We have variously indicated the lack of existing research on QoS evaluation in a M2M setting, which is our final destination in the scheme of things. Therefore, our research contributes specifically in QoS evaluation in sensor networks for M2M and generally in the area of network QoS evaluation.

The research also touches on some WSN aspects in a peripheral manner. These include routing protocols, power efficiency and wireless channel management. Suffice to say, the research will add onto the existing body of literature.

5.4 Recommendations and Future Work.

Measurement of QoS metrics has always been carried out through network management systems for most types of networks. This has led to the evolution of different QoS management methods, mostly involving empirically obtained data that is massive in nature. However, the evaluation of QoS for sensor type networks and indeed machine type communications has been very minimal and is now becoming mainstream with the growth of consumer electronics and smart wearable in the Internet of Things. Such evaluation requires the consumer to be closely involved and accommodated as much as possible in determining the kind of service they would want (Du et. al 2009). In this regard, consumer opinion needs to be integrated as much as possible within QoS evaluation and management systems. It is therefore our recommendation that future methodologies for QoS evaluation and management within Internet of Things technologies be consumer centric in their basic constructions. By this we mean for researchers and developers to include the subjective aspect which will ultimately lead to an improvement in the final end user's satisfaction index. M2M networks are generating a lot of data in measured QoS metrics that easily fit into the big data realm. A possible future area of study is the use of data mining functions in combination with decision theory techniques to design scalable QoS evaluation models that can effectively assess customer satisfaction as a final objective. It is our belief that as machine type networks proliferate, more and more performance data will become available and more complex in nature. It will require these kinds of models to manage the consumer impacting QoS issues.

REFERENCES

1. Al-Sbou, Y., 2010. Fuzzy Logic Estimation System of Quality of Service for Multimedia Transmission, International Journal of QoS Issues in Networking (IJQoSIN), Vol. 1, No. 1, pp. 1-9.

2. Bochmann G.V., Hafid, A., 1997. Some Principles for Quality of Service Management, Distributed Systems Engineering, Volume 4 Number 1.

3. Bonneau, V., 2005, Machine-to-Machine Strong Growth of Wireless M2M and Impact of RFID, COMMUNICATIONS & STRATEGIES, no. 59, 3rd quarter, p. 175.

4. Brazell, J.B.,Donoho, L.,Dexheimer, J.,Hanneman,R.,Langdon,G. 2005. M2M: The wireless Revolution - A technology forecast M2m the wireless revolution a technology forecast. IC2 Institute, University of Texas

5. Bulbul, M. A. 2010. A Short Fuzzy Logic Tutorial

6. Cha, I.,Shah, Y.,Schimidt,A.U.,Leicher, A., Meyerstein,M. 2009. Security and Trust for M2M Communications, Technical Paper, 22nd Wireless World Research Forum

7. Chalmers, D., Sloman, M. 1999. A survey of quality of service in mobile computing environments. Communications Surveys & Tutorials, IEEE, 2(2), pp. 2-10.

8. Chen, D., & Varshney, P. K., 2004. QoS Support in Wireless Sensor Networks: A Survey. In International Conference on Wireless Networks (Vol. 233) pp. 227-233

9. Dimou,K., 2013.Senior Research Engineer, Ericsson Research MACHINE TO MACHINE (M2M) COMMUNICATIONS-PART I BASICS & CHALLENGES

10. Dogman, A., Saatchi, R., & Al-Khayatt, S. 2012. Quality of Service Evaluation using a Combination of Fuzzy C-Means and Regression Model. In Proceedings of World Academy of Science, Engineering and Technology (No. 61). World Academy of Science, Engineering and Technology.

11. Dohler, M., Watteyne, T., Alonso-Zárate, J. 2010. Machine-to-Machine: An Emerging Communication Paradigm.

12. Du, Y., Zhou, W., Chen, B., & Song, J. 2009. A QoE based evaluation of service quality on Wap in wireless network. Networking and Mobile Computing in Wireless Communications, 2009. WiCom'09.

13. Ericsson AB, 2011. More than 50 billion connected devices, White Paper.

14. ETSI, Machine to Machine Communications,[online] Available at http://www.etsi.org/technologies-clusters/technologies/m2m

15. Evans, D., 2011. The Internet of Things: How the Next Evolution of the Internet Is Changing Everything. White Paper, Cisco Internet Business Solutions Group (IBSG).

16. Forman, E. H., with Gass, S I, 2001 “The Analytic Hierarchy Process – An Exposition”, Operations Research, vol.49, no. 4, July-Aug, pp. 469-486.

17. Galetić, V., Bojić, I., Kušek, M., Ježić, G., Dešić, S., Huljenić, D. 2011. Basic principles of Machine-to-Machine communication and its impact on telecommunications industry, MIPRO, 2011 Proceedings of the 34th International Convention , vol., no., pp.380-38

18. GRIERA, M., JIMÉNEZ, M., MARTÍNEZ, J. A. ca. 2000 QOS evaluation model for a Campus-Wide Network: an alternative approach.

19. Guillemin, P.,Friess, P., 2009. Internet of Things Strategic Research Roadmap, Cluster of European Research Projects on the Internet of Things (CERP-IoT)

20. Gupta S., Hirdesh. A., 200-. OVERVIEW of M2M, Department of Computer and Information Science and Engineering.

21. Haas, R and Meixner, O. 2005. An Illustrated Guide to the ANALYTIC HIERARCHY PROCESS, Institute of Marketing & Innovation University of Natural Resources and Applied Life Sciences, Vienna

22. Hardy, W. C., 2001. QoS: Measurement and Evaluation of Telecommunications Quality of Service, John Wiley & Sons, Ltd. ISBNs: 0-470-84591-0 (Electronic).

ITU-T, 2008. Recommendation E.800 (09/08).

23. Jäger, S., Jungebloud, T., Maschotta, R., & Zimmermann, A. 2013. Model-based QoS evaluation for embedded wireless sensor networks. In Proceedings IEEE International Systems Conference (SYSCON’13), Orlando, Florida USA.

24. Jager, S., Jungebloud, T., Maschotta, R., Zimmermann, A., 2013. Model-based QoS evaluation for embedded wireless sensor networks, Systems Conference (SysCon), 2013 IEEE International , vol., no., pp.195,199.

25. Juniper Networks, 2011. Machine -to-Machine (M2M)— The Rise of the Machine's, White paper

26. Koucheryavy, A., Al-Naggar, Y. 2013. The QoS Estimation for Physiological Monitoring Service in the M2M network, The INTHITEN (INternet of THings and ITs ENablers) conference,
27. Lien, S-Y., Chen K-C., 2011. Massive Access Management for QoS Guarantees in 3GPP Machine-to-Machine Communications, Communications Letters, IEEE , vol.15, no.3, pp.311-313
28. Liu, Chun-Chu, 2008. Analytic Hierarchy Process (What is AHP).
29. Lollini, P., Bondavalli, A., Di Giandomenico, F. 2005. QoS evaluation in a UMTS cell.
30. Marshal, A., 2009. Quality of Service in Wireless IP Networks: Trends & Challenges Trends Challenges, Advanced Telecommunications Group. School of Electrical & Electronic Engineering
31. Mendel, J. 1995. Fuzzy logic systems for engineering: a tutorial. Proceedings of the IEEE, 83(3) pp 345–377.
32. Nedeljkovic, I. 2004. Image classification based on fuzzy logic. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 34(Part XXX). 5th International Conference on (pp. 1-4). IEEE.
33. Nef, M-A.,Karagiorgou,G.I.S.,Kikiras,P.K., 2012. Enabling QoS in the Internet of Things, CTRQ 2012 : The Fifth International Conference on Communication Theory, Reliability, and Quality of Service
34. Numerex Corp, Numerex defines M2M, [online] Available at http://www.numerex.com/
30. OECD, 2012. Machine-to-Machine Communications: Connecting Billions of Devices, OECD Digital Economy Papers, No. 192, OECD Publishing.
35. Song, J., Zhang, S., Gong, Y., & Dai, B. 2012. A QoS Evaluation Model for Test-Bed in the Cloud Computing Environment. IEEE Ninth International Conference on e-Business Engineering (ICEBE), (pp. 292-295)
36. Wang, J., Liu, H., & Song, M. 2011. A novel QoS evaluation scheme based on support vector machine. 2011 Seventh International Conference on Natural Computation (ICNC) (Vol. 2, pp. 724-727). IEEE.
37. Weber, Mathew L., 2012. A Survey of Quality of Service (QoS) Considerations in 802.15.4 Zigbee Networks. IOWA STATE UNIVERSITY - CPRE543, 2012FALL.

38. Wu,G., Talwar,S., Johnsson,K., Himayat,N., Johnson,K.D. 2011. M2M: From Mobile to Embedded Internet, RECENT PROGRESS IN MACHINE-TO-MACHINE COMMUNICATIONS, IEEE Communications Magazine, pp. 36-43

39. Younis, M., Akkaya, K., Eltoweissy, M., & Wadaa, A. 2004. On handling QoS traffic in wireless sensor networks. In System Sciences, Proceedings of the 37th Annual Hawaii International Conference on Information System Sciences(pp. 10-pp). IEEE.

40. Zadeh, L. A. 1965. Fuzzy sets. Information and control, 8(3), pp. 338-353.

41. Zakaria, A. ca 2005. Quality of Service in Wireless Sensor Networks.