University of Strathclyde
Department of
Electronic and Electrical Engineering

Cognitive Wireless Sensor Networks (CogWSNs)

by

Hock Guan Goh

A Thesis presented in fulfilment of the requirements for
the degree of Doctor of Philosophy

February 2014

Copyright

Signed:

Date: 18 February 2014

Acknowledgement

I thank my great teacher, the Buddha, for his teachings on the Dharma.

I would like to express my sincere appreciation to my supervisor, Prof Ivan Andonovic and my co-supervisors, Dr. Craig Michie and Dr. Kae Hsiang Kwong for their great guidance, advice, and support. I am indebted to them for giving me an opportunity to work and study at University of Strathclyde, Glasgow, Scotland. I would also like to thank them for their countless stimulating discussions and numerous insights.

I would like to express my gratitude to my colleagues Swee Keow Goo, Konstantinos Sasloglou, Tsung-Ta Wu, Yen Kaow Ng, Kok Lim Yau, Christian Andrae, Di Cao, Gavin Tsao, and Fan Yu Meng for discussions and for their support during my research. I wish them all the best in their future careers.

Special acknowledgement is also extended to Prof. Hong Tat Ewe and Dr. Soung Yue Liew from the Universiti Tunku Abdul Rahman (UTAR), Malaysia for offering me an environment to continue my experiments and writing up of the Thesis.

Lastly, I would like to thank my parents, my wife, Hooi Ping, and my child, Jyie Shann for their constant love and encouragement throughout my study.

Abstract

Cognitive Wireless Sensor Networks (CogWSNs) are an adaptive learning based wireless sensor network relying on cognitive computational processes to provide a dynamic capability in configuring the network. The network is formed by sensor nodes equipped with cognitive modules with awareness of their operating environment. If the performance of the sensor network does not meet requirements during operation, a corrective action is derived from stored network knowledge to improve performance. After the action is implemented, feedback on the action taken is evaluated to determine the level of improvement.

Example functions within CogWSNs can be as simple as to provide robust connectivity or as complex as to negotiate additional resources from neighbouring network groups with the goal of forwarding application-critical data. In this work, the concept of CogWSNs is defined and its decision processes and supporting architecture proposed. The decision role combines the Problem Solving cognitive process from A Layered Reference Model of the Brain and Polya Concept, consisting of Observe, Plan, Implement, and Evaluate phases. The architecture comprises a Transceiver, Transducer, and Power Supply virtual modules coordinated by the CogWSN's decision process together with intervention, if necessary, by a user.

Three types of CogWSN modules are designed based on different implementation considerations: Rule-based CogWSN, Supervised CogWSN, and Reinforcement CogWSN. Verification and comparison for these modules are executed through case studies with focus on power transmission and communication slot allocation. Results show that all three modules are able to achieve targeted connectivity and maintain utilisation of slots at acceptable data rates.

List of Abbreviations

ADC	-	Analog-to-Digital Converter
ADS	-	Advanced Decision Systems
AIMRP	-	Address-light, Integrated MAC and Routing Protocol
ANN	-	Artificial Neural Network
ANMP	-	Ad hoc Network Management Protocol
API	-	Application Programming Interface
ARPANET	-	Advanced Research Projects Agency Network
AWACS	-	Airborne Warning and Control System
BS	-	Base Station
B-MAC	-	Berkeley-MAC
CEC	-	Cooperative Engagement Capability
CMU	-	Carnegie Mellon University
CogWSN	-	Cognitive Wireless Sensor Network
CPU	-	Central Processing Unit
CSI	-	Channel-State Information
CSMA	-	Carrier Sense Multiple Access
DARPA	-	Defence Advanced Research Projects Agency
DLL	-	Data Link Layer
DSN	-	Distributed Sensor Networks
ESRT	-	Event-to-sink Reliable Transport
FDS	-	Fixed Distributed System
FSPL	-	Free Space Path Loss
GPIO	-	General Purpose Input/Output
GPS	-	Global Positioning System
I^2C	-	Inter-Integrated Circuit
IoT	-	Internet of Things
IPS	-	instructions per second
ISI	-	Inter-Symbol Interference
ISM	-	Industrial, Scientific and Medical

ITU-R	-	International Telecommunications Union-Radio
LMS	-	Least Mean Square
LRMB	-	Layered Reference Model of the Brain
MAC	-	Media Access Control
MEMS	-	Micro-Electro-Mechanical Systems
MIMO	-	Multiple-Input and Multiple-Output
MIT	-	Massachusetts Institute of Technology
NOAA	-	National Oceanographic and Atmospheric Administration
OSI	-	Open Systems Interconnection
PAN	-	Personal Area Network
PSFQ	-	Pump-Slowly, Fetch-Quickly
PWM	-	Pulse Wave Modulation
QoS	-	Quality-of-Service
RAM	-	Random Access Memory
RBA	-	Rule-based CogWSN
RBL	-	Rule-based CogWSN with Greedy Scoring
REMBASS	-	Remote Battlefield Sensor System
RKRL	-	Radio Knowledge Representation Language
RL	-	Reinforcement CogWSN
RMST	-	Reliable Multi-Segment Transport
ROM	-	Read-Only Memory
RSSI	-	Received Signal Strength Indicator
SAN	-	Software Adaptable Network
SDR	-	Software Defined Radio
SI	-	State Information
SL	-	Supervised CogWSN
SensIT	-	Sensor Information Technology
SMP	-	Sensor Management Protocol
SNMP	-	Simple Network Management Protocol
SOSUS	-	Sound Surveillance System
SPI	-	Serial Peripheral Interface
SQDDP	-	Sender Query and Data Dissemination Protocol

STEM	-	Sparse Topology and Energy Management
STEM-B	-	STEM Beacon
STEM-T	-	STEM Tone
S-MAC	-	Sensor-MAC
TADAP	-	Task Assignment and Data Advertisement Protocol
TASS	-	Tactical Automated Security System
TF	-	Tuneable Function
TRAMA	-	Traffic-Adaptive Medium Access Protocol
TRSS	-	Tactical Remote Sensor System
UART	-	Universal Asynchronous Receiver/Transmitter
T-MAC	-	Timeout-MAC
USART	-	Universal Synchronous/Asynchronous Receiver/Transmitter
UWB	-	Ultra-Wide Band
VMS	-	Virtual Memory System
WSN	-	Wireless Sensor Network
XLP	-	Cross-Layer Protocol
Z-MAC	-	Zebra MAC

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction

The Chapter is an overview of the research starting with the motivation for the work and identifying the challenges to be addressed. Objectives are then defined and main contributions arising out of the research highlighted. The organisation of the Thesis is outlined. Finally, a list of related publications is recorded.

1.1 Motivation

The requirements and demands of monitoring physical environments with low cost devices have stimulated the development of Wireless Sensor Networks (WSNs) [1, 2], a technology able to be deployed at large scale and widely used in various applications such as military [3], agriculture [4, 5], health care [6], home [7], and commercial [8].

In general, a WSN is formed by a group of sensor nodes equipped with a short range communication capability. Each of the sensor nodes consists of at least a transceiver, a transducer, a processing unit, and a power unit [9]. The sensor node is small in size comprising inexpensive components; the size of the sensor node often restricts the power supply capacity. Due to the cost of manufacturing and market demand, low data rate transceivers operating in license-exempt Industrial, Scientific and Medical (ISM) frequency bands are most often used [10]. The nodes are also capable of forming self-organised networks using multi-hop communications [11].

The limited radio capability and uncertainty in physical operating environments pose fundamental constraints in optimizing WSN connectivity [12]. Currently, a WSN lacks the capability of fine tuning its radio configuration dynamically to meet the challenges of a changing operating environment. As a result, degradation in radio link performance and unreliable network connectivity can be expected [13].

Applying artificial intelligence, especially cognitive approaches, has been identified as a viable solution to circumvent the above challenges. Solutions such as Cognitive Radio [14] and Cognitive Networks [15] exploit tuneable parameters on the hardware to create and maintain optimum network communications.

The Thesis addresses the intersection of WSN technology and cognitive computational intelligence. Cognition has been used in communication [16], sensing [17], or system application [18] of WSNs. As WSNs comprise transceivers, transducers, processors, and power units, cognition could be applied across all core elements.

Here, the concept of Cognitive Wireless Sensor Network (CogWSN) is defined, and its decision processes and architecture proposed. The decision process methodology consists of four phases; Observe, Plan, Implement, and Evaluate. The architecture comprises three core virtual modules; Transceiver, Transducer, and Power Supply. These virtual modules are coordinated by the CogWSN's decision process but have the option for intervention by the user. Each virtual module contains 'State Information', the storing of information about the operating conditions and a 'Tuneable Function', defining the actuating function.

Three types of CogWSN are designed based on different implementation considerations: Rule-based CogWSN [19], Supervised CogWSN [20], and Reinforcement CogWSN [21]. Verification of each method is carried out to evaluate performance in terms of power transmission and communication slot allocation. Comparison of the three options is executed to quantify the performance in different, repeated, representative scenarios.

1.2 Objectives

The performance of WSNs is often compromised by dynamic changes in their operating environment [22]. The environment in this context refers to the communication medium, hardware, energy resource and data traffic flows. When these changes occur post deployment, human intervention is needed in order to diagnose their impact and to resolve problems through manual configuration. Several questions arise related to these scenarios:

1. Can human intervention in tuning a WSN configuration during deployment be eliminated?
2. What modifications or additional elements are required in order to support the proposed solutions?
3. How to embed, within the solution the capability to be aware of the configuration that it needs to tune?
4. How well does the proposed solution perform?

In order to provide answers to the above questions, the following objectives form the focus of the study:

1. To define the concept of CogWSN providing a solution to reduce human intervention in tuning WSN configurations during deployment.
2. To define appropriate decision processes and architectures in order to support the methodology.
3. To equip the decision processes with three learning approaches: rule-based learning, supervised learning, and reinforcement learning.
4. To verify and compare the performance of the three decision strategies in terms of power transmission and communication slot allocation.

1.3 Main Contributions

The main contributions of the research are:

1. Proposed a concept and determined the elements of CogWSNs. The term ‘CogWSN’ is defined as a networked group of sensor nodes capable of sustaining performance in a dynamic environment using an embedded cognitive capability, whilst meeting user requirements. CogWSN is supported by an architecture which comprises Transceiver, Transducer, and Power Supply modules. Each module contains ‘State Information’, the

storing of information about the module's operating condition and a 'Tuneable Function' defining the actuating function.

2. Proposed and defined the CogWSN decision process and design of the architecture merging cognitive computational intelligence with sensor network technologies. The CogWSN architecture comprises Transceiver, Transducer, and Power Supply modules coordinated by a decision process accepting intervention, if necessary, by a user. The decision role combines the Problem Solving cognitive process from A Layered Reference Model of the Brain and Polya Concept [23], comprising Observe, Plan, Implement, and Evaluate phases. During the Observe phase, the desired parameters are monitored closely, and if a parameter is detected to be out with the controlled range, it is identified as a problem. In the Plan phase, a plan is derived to solve the identified problem. The solution is implemented according to the plan in the Implement phase. Finally in the Evaluate phase, an evaluation is carried out to determine how well the problem is solved. The evaluation result is stored as a reference for similar problems in the future.
3. Implemented CogWSN operation using rule-based learning, supervised learning, and reinforcement learning. Rule-based CogWSN requires full *a priori* knowledge of the target goals to be established; Rule-based CogWSN with Greedy Scoring requires all possible actions with parameters to be defined but not the decision; Supervised CogWSN only requires partial trained knowledge; and Reinforcement CogWSN does not need any knowledge to be installed at the outset.
4. Validated the performance of CogWSN. The proposed CogWSN performance in terms of transmission power and communication slots allocation has been evaluated. The solutions are benchmarked with algorithms drawn from reported research.

1.4 Organisation of the Thesis

The Thesis is organised as follows. A review of the literature on the history, evolution, architecture, and application of WSNs is presented in Chapter 2. The Chapter also discusses cross-layer design, machine learning, cognitive techniques applied to WSNs, and standardisation in relation to WSNs. In Chapter 3, the principles underpinning CogWSN are defined together with a proposal for its decision process and architecture. Three types of CogWSN learning approaches are introduced and designs presented in the following three Chapters; in Chapter 4, Rule-based CogWSN; in Chapter 5, Supervised CogWSN; and in Chapter 6, Reinforcement CogWSN. Case studies are established, their performance characterised and results presented in all three Chapters. Relative performance among these three implementations is documented in Chapter 6. Finally, conclusions on the research and suggestions for future work are presented in Chapter 7.

1.5 List of Publications

1. K. H. Kwong, T. T. Wu, **H. G. Goh**, K. Sasloglou, B. Stephen, I. Glover, C. Shen, W. Du, C. Michie, and I. Andonovic, "Practical Considerations for Wireless Sensor Networks in Cattle Monitoring Applications," *Computers and Electronics in Agriculture*, vol. 81, pp. 33-44, Feb. 2012.

2. **H. G. Goh**, S. Y. Liew, K. H. Kwong, C. Michie, and I. Andonovic, "Abstract Reporting and Reformation Schemes for Wireless Sensor Networks", *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering (LNICST) Vol. 72, The 1st International Conference on Wireless Communications and Applications (ICWCA 2011)*, Hainan Island, China, pp. 69-74, 1-3 Aug. 2011.

3. K. H. Kwong, T. T. Wu, **H. G. Goh**, K. Sasloglou, B. Stephen, I. Glover, C. Shen, W. Du, C. Michie, and I. Andonovic, "Implementation of Herd Management Systems with Wireless Sensor Networks," *IET Wireless Sensor Systems*, vol. 1, no. 2, pp. 55-65, Jun. 2011.

4. I. Andonovic, C. Michie, M. Gilroy, **H. G. Goh**, K. H. Kwong, K. Sasloglou, and T. Wu, "Wireless Sensor Networks for Cattle Health Monitoring," invited chapter of a book entitled *ICT Innovations 2009 (Ed.: D. Davcev and J. M. Gomez)*, Springer-Verlag Berlin Heidelberg, ISBN: 978-3-642-10780-1, e-ISBN: 978-3-642-10781-8, 2010, pp. 21-31.

5. **H. G. Goh**, S. Y. Liew, K. H. Kwong, C. Michie, and I. Andonovic, "Cognitive Wireless Sensor Network," *International Conference on Advanced Infocomm Technology (ICAIT 2010)*, Hainan, China, 20-23 Jul. 2010. (abstract)

6. T. -T. Wu, D. Cao, B. Stephen, **H. G. Goh**, K. H. Kwong, W. Du, C. Shen, C. Michie, and I. Andonovic, "A Practical Data Reporting Solution for Free-ranging Cattle Monitoring Applications Using Wireless Sensor Networks," *The 2010 American Society of Agricultural and Biological Engineers (ASABE 2010)*, Pittsburgh, Pennsylvania, United States, 20-23 Jun. 2010.

7. D. Cao, T.-T. Wu, **H. G. Goh**, B. Stephen, K. H. Kwong, C. Michie, and I. Andonovic, "Exploitation of Wireless Telemetry for Livestock Condition Monitoring," *The XVIIth World Congress of the International Commission of Agricultural and Biosystems Engineering (CIGR 2010)*, Québec City, Canada, 13-17 Jun. 2010.

8. **H. G. Goh**, K. H. Kwong, C. Shen, C. Michie, and I. Andonovic, "CogSeNet: A Concept of Cognitive Wireless Sensor Network", *IEEE Consumer Communications and Networking Conference (CCNC 2010)*, Las Vegas, Nevada, United States, 9-12 Jan. 2010. (short paper)

9. K. Sasloglou, I. A. Glover, **H. G. Goh**, K. H. Kwong, M. P. Gilroy, C. Tachtatzis, C. Michie, and I. Andonovic, "Antenna and Base-station Diversity for WSN Livestock Monitoring," *Wireless Sensor Network, Scientific Research Publishing*, vol. 1, no. 5, pp. 383-396, Dec. 2009.

10. K. H. Kwong, T. -T. Wu, K. Sasloglou, B. Stephen, D. Cao, **H. G. Goh**, S. K. Goo, M. Gilroy, C. Tachtatzis, I. A. Glover, C. Michie, and I. Andonovic,

"Implementation of Herd Management System with Wireless Sensor Networks," *Joint International Agricultural Conference (JIAC 2009)*, Wageningen, Netherlands, 6-8 Jul. 2009.

11. K. H. Kwong, **H. G. Goh**, T. -T. Wu, B. Stephen, C. Michie, I. Andonovic, D. Ross, J. Hyslop, and C. X. Liu, "Wireless Telemetry for Livestock Condition Monitoring," *The 2009 American Society of Agricultural and Biological Engineers (ASABE 2009), The 7th World Congress of Computers in Agriculture (WCCA 2009)*, Reno, Nevada, United States, 22-24 Jun. 2009.

12. D. Cao, F. Y. Meng, **H. G. Goh**, K. H. Kwong, C. Michie, and I. Andonovic, "An Evaluation of Positioning System for Wireless Sensor Networks in an Indoor Environment," *Proceedings of the 10th Annual PostGraduate Symposium on The Convergence of Telecommunications, Networking and Broadcasting (PGNET 2009)*, Liverpool John Moores University, Liverpool, United Kingdom, pp. 10-13, 22-23 Jun. 2009.

13. K. H. Kwong, K. Sasloglou, **H. G. Goh**, T. -T. Wu, B. Stephen, M. Gilroy, C. Tachtatzis, I. A. Glover, C. Michie, and I. Andonovic, "Adaptation of Wireless Sensor Network for Farming Industries," *The 6th International Conference on Networked Sensing Systems (INSS 2009)*, Carnegie Mellon University, Pittsburgh, United States, pp. 1-4, 17-19 Jun. 2009.

14. D. Cao, **H. G. Goh**, K. H. Kwong, C. Michie, and I. Andonovic, "Positioning System for Wireless Sensor Networks with Location Fingerprinting," *Progress In Electromagnetics Research Symposium (PIERS 2009)*, Beijing, China, 23-27 Mar. 2009. (abstract)

15. K. H. Kwong, T. -T. Wu, **H. G. Goh**, B. Stephen, M. Gilroy, C. Michie, and I. Andonovic, "Wireless Sensor Networks in Agriculture: Cattle Monitoring for Farming Industries," *Progress In Electromagnetics Research Symposium (PIERS 2009)*, Beijing, China, 23-27 Mar. 2009. (invited paper)

16. **H. G. Goh**, K. H. Kwong, C. Michie, and I. Andonovic, "Performance Evaluation of Priority Packet for Wireless Sensor Network", *The 2nd IARIA International Conference on Sensor Technologies and Applications (SENSORCOMM 2008)*, Cap Esterel, France, 25-31 Aug. 2008.

17. K. H. Kwong, **H. G. Goh**, C. Michie, and I. Andonovic, "Quality of Service Control in Wireless Sensor Network with Priority Access," *The 2nd International Workshop on Connecting Health: Using IT to Assist Patient Recovery (ICST QSHINE, ConnectingHealth 2008)*, Hong Kong, 31 Jul. 2008.

18. K. H. Kwong, **H. G. Goh**, C. Michie, I. Andonovic, B. Stephen, T. Mottram, and D. Ross, "Wireless Sensor Networks for Beef and Dairy Herd Management," *The 2008 American Society of Agricultural and Biological Engineers (ASABE 2008)*, Providence, Rhode Island, United States, 29 Jun.-2 Jul. 2008.

19. **H. G. Goh**, K. H. Kwong, C. Michie, and I. Andonovic, "Performance Evaluation of Priority Packet for Wireless Sensor Network," *Proceedings of the 9th Annual PostGraduate Symposium on The Convergence of Telecommunications, Networking and Broadcasting (PGNET 2008)*, Liverpool John Moores University, Liverpool, United Kingdom, pp. 15-19, 23-24 Jun. 2008.

20. K. H. Kwong, **H. G. Goh**, C. Michie, and I. Andonovic, "Performance Evaluation of Priority Medium Access Control (P-MAC) for Wireless Sensor Network," *Proceedings of the 5th IASTED Asian Conference on Communication Systems and Networks (AsiaCSN 2008)*, Langkawi, Malaysia, 2-4 April 2008.

Chapter 2: Review

The Chapter presents a review of the literature relating to the research undertaken. The review begins with the history of WSNs, and although origins dating back to the start of the ‘Cold War’ can be identified [24], the area still remains subject to extensive research. The evolution of WSNs is then related to an introduction to architectures, performance limitations and challenges. Related research in cross-layer design [25], machine learning [26], cognitive techniques applied in WSNs [27], and standardisation [28] are discussed. The Chapter ends with a summary and analysis of the state-of-the-art in WSNs.

2.1 History and Evolution of WSN

Research on sensor networks has its roots as far back as the ‘Cold War’ period [29]; similar to many other networking technologies, military applications have primarily driven WSN research progress and development. During the Cold War, a network of sensors on the seabed – the Sound Surveillance System (SOSUS) [30] - was deployed at strategic locations to provide warning of Soviet submarines approaching the continental United States. SOSUS is a network of acoustic sensors (hydrophones [31]) placed at the bottom of the ocean; even more sophisticated acoustic sensor networks have been developed for submarine surveillance over more recent years [32]. SOSUS is currently being used by the National Oceanographic and Atmospheric Administration (NOAA) [33] for monitoring events, such as seismic activity. Networks of air defence radar installations were also developed and deployed to defend the United States and Canada during the Cold War. The system has evolved over the years to include aerostats as sensors [34] and Airborne Warning and Control System (AWACS) aeroplanes [35], and is also used for the purpose of detecting illegal drugs carried by aircraft [36].

The start of modern research on sensor networks was stimulated by the ‘Distributed Sensor Networks (DSN)’ programme at the Defence Advanced Research Projects Agency (DARPA) [24]. Kahn [37], the co-inventor of the TCP/IP protocol and responsible for originating DARPA's Internet Program, wanted the Advanced Research Projects Agency Network (ARPANET) [38] (predecessor of the Internet) extended to support sensor networks. Initial testing was executed on minicomputers, such as PDP-11, VAX machines running UNIX and Virtual Memory Systems (VMS) with Modems operating at 300 Baud to 9600 Baud [24].

In 1978, DSN was identified as a technology component within the Distributed Sensor Nets Workshop [39] held in Carnegie Mellon University (CMU). Distributed acoustic tracking was chosen as the target for demonstration; this foundation spawned several streams of research summarised in Table 2.1.

Table 2.1. Research carried out on the DSN test-bed.

Research Institute	**Research Focus Area [references]**
Carnegie Mellon University (CMU), Pittsburgh, Pennsylvania	Provided a network operating system that allows flexible, transparent access to distributed resources needed for fault-tolerant DSNs [40].
Massachusetts Institute of Technology (MIT), Cambridge, Massachusetts	Implemented knowledge-based signal processing techniques for tracking helicopters using a distributed array of acoustic microphones by means of signal abstractions and matching techniques [41].
Advanced Decision Systems (ADS), Mountain View, California	Developed a multiple-hypothesis tracking algorithm [42, 43].
MIT Lincoln Laboratory, Lexington, Massachusetts	Developed a real time test-bed for acoustic tracking of low-flying aircraft [44].
University of Massachusetts, Amherst, Massachusetts	Developed a test-bed for distributed vehicle monitoring [45].

Although early research on sensor networks projected that large numbers of small form factor sensors were needed for any deployment, the technology for manufacturing these families of sensors was immature and hence costly. However, planners of military systems recognized the benefits of these sensor networks and their potential to become a crucial component of network-centric warfare [46]. In network-centric warfare applications, sensors and weapons are mounted with and controlled by separate platforms that operate independently [46]; sensors and weapons collaborate over a communication network. Examples are Cooperative Engagement Capability (CEC) [47] using multiple radar antennas to collect data on air targets, Fixed Distributed System (FDS) [48], Advanced Deployable System (ADS) [49] using acoustic sensors arrays for anti-submarine warfare, Remote Battlefield Sensor System (REMBASS) [50] and Tactical Remote Sensor System (TRSS) [51] using unattended ground sensors across battlefield terrain.

Latterly, advances in computing and communication technologies have stimulated a significant shift in sensor networks research and progressed the implementations nearer to that of the original vision. Small and inexpensive sensors based on Micro-Electro-Mechanical Systems (MEMS) [52], cost effective wireless networking chipsets [53], and inexpensive low-power processors allow large deployments of wireless sensor networks for various applications. Again, DARPA initiated a research program on sensor networks to leverage the latest technological advancements through the 'Sensor Information Technology (SensIT)' [54] program. There were two core themes within SensIT; to develop new networking techniques for ad-hoc operation within highly dynamic environments and to deploy networked information processing to extract useful, reliable, and timely information from the deployed sensor network. SensIT networks would exhibit new capabilities such as interactivity and programmability with dynamic tasking and querying; one early example of this concept is the Tactical Automated Security System (TASS) [55].

Today, WSNs represent a new generation of real-time embedded systems with significantly different communication constraints from traditional networked systems

[11]. WSNs can self-organize [56], achieve multi-hop connections and have become the foundation for the potential realization of the Pentagon-inspired "smart-dust" concept [57] proposed by researchers at DARPA. The principle is to sprinkle thousands of tiny wireless sensors across a battlefield to monitor enemy movements without alerting the enemy to their presence. A "smart dust" optical mote [58, 59] uses MEMS to realise sub-millimetre-sized mirrors for establishing communication paths. By self-organizing into a sensor network, 'smart dust' filters raw data for relevance before relaying only the important findings to central command.

From the outset, to maintain the stringent principles underpinning the operating strategy and value of WSNs, the nodes/motes have to be low cost and of small form factor. Many of the early WSN deployments are best viewed as "proof of concept" [60, 61]. The transceivers were built using simple radio chips, only allowing single channel transmission at a time. Frequencies such as 315MHz, 433MHz, 868MHz, 915MHz, and 2.4GHz are chosen simply since they all fall under unlicensed ISM bands [62, 63]. The data rate was low, usually less than 1Mbps for a point-to-point link. The communication layers only consisted of the application, transport, network, data link and physical layers [1]. The sensor node utilised the most basic low computing power such as 8-bit or 16-bit microcontrollers [64]. Overall, node power consumption had to be low and power saving strategies formed one of the main design considerations. Since limited power sources were available, most routinely a battery, complex transducers that require external power sources were not considered in early deployments [65].

The desire to derive more information from deployed sensors was the seed to the development of the next stage of the WSN design evolution, with particular emphasis on WSNs comprising a large number of nodes [66, 67]. More sophisticated techniques were proposed such as Ultra-Wide Band (UWB) [68, 69] and implementing diversity techniques like Multiple-Input and Multiple-Output (MIMO) [70] operation in order to provision enhanced communication. The direct impact of these advanced designs was increased cost when compared to more traditional WSN implementations. As more communication capabilities were provided, the point-to-point data rate exceeded 1Mbps [71]. More sophisticated network protocols [72, 73] are being considered for implementations of next generation WSNs for the realisation of the Internet of Things (IoT) [74]. The sensor node is expected to offer more computing power, providing more memory and be able to execute more instructions per second (IPS). Consequently it is expected that the sensor node will consume more of the limited energy resource, which prompts the integration of sensor nodes with energy harvesting or energy recharging modules [75].

2.2 WSN Architecture

A WSN is a wireless *ad-hoc* network formed by a group of sensor/actuator devices (nodes). A minimal WSN node consists of a transceiver, a transducer, a processor, and a power unit. It has a small form factor and is implemented with inexpensive components (Figure 2.1). WSN nodes are capable of forming a self-organised multi-hop network.

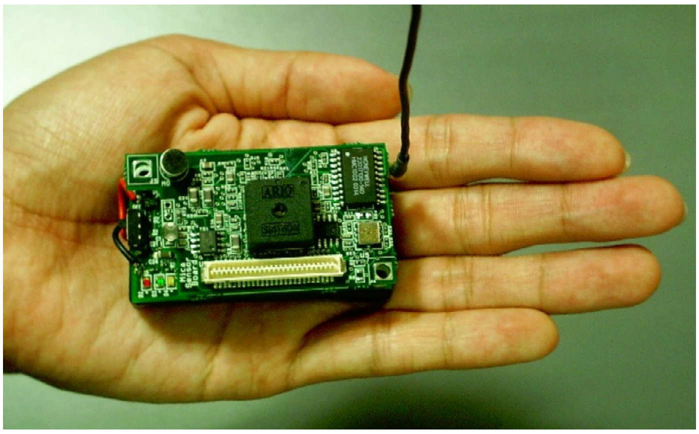

Figure 2.1. A WSN development platform (MICAz with MTS310 sensor board) [76].

In more detail, there are several core components that comprise a WSN node; a sensor and/or actuator, microcontroller or microprocessor, unique identifier chip, external flash memory, radio chip, antenna, and batteries (Figure 2.2). A sensor is a component that detects a parameter in one form and presents/transduces that measurement into an electrical signal, while an actuator converts electrical control signals into non-electrical energy such as kinetic energy; both of are often referred to as transducers. A microcontroller with Read-Only Memory (ROM) and Random Access Memory (RAM) is effectively a modest computer on a single integrated device that can either store data and/or execute software routines/programs [77]. It consists of a relatively simple central processing unit (CPU) combined with support functions such as a crystal oscillator, timers, interrupts and interfaces [78]. The variety of interfaces can be Analog-to-Digital Converter (ADC), General Purpose Input/Output (GPIO), Pulse Wave Modulation (PWM), Universal Asynchronous Receiver/Transmitter (UART), Universal Synchronous/Asynchronous Receiver/Transmitter (USART), Inter-Integrated Circuit (I^2C), and/or Serial Peripheral Interface (SPI) bus [78]. An identity is provided on each chip (can be a unique identifier chip and/or radio chip), imparting a unique identification to each sensor node. A Flash Memory [79] may be used as external storage for storing data locally on each node; Flash can be optional. A Radio chip [80] is responsible for the transmission of data signals wirelessly within the Radio Frequency segments of the electromagnetic spectrum, frequencies well below visible light, employing a range of modulation techniques [81]. An antenna is used to transmit or receive electromagnetic waves. A recent evolution of the platform [82] offers an integrated microcontroller and radio on a single chip.

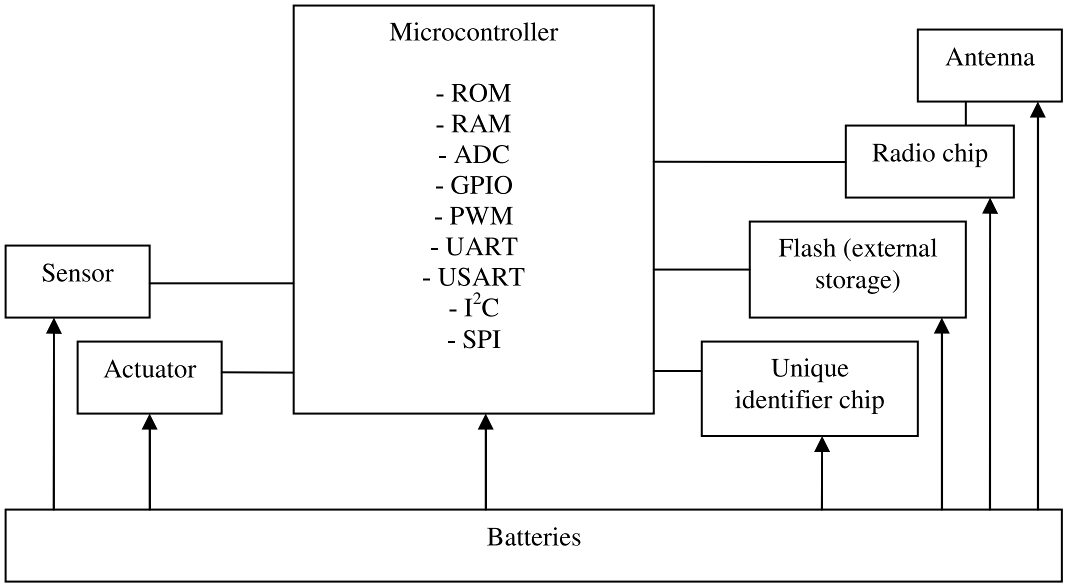

Figure 2.2. A general hardware block diagram of a typical WSN node [76].

WSN architectures can be represented in a simple block diagram as shown in Figure 2.3. The Transducer(s) interacts with the physical world; the Processor manages the signal transfer between transducer and transceiver; the Transceiver manages the transmit/receive function of the radio signal. All components are powered by a power unit.

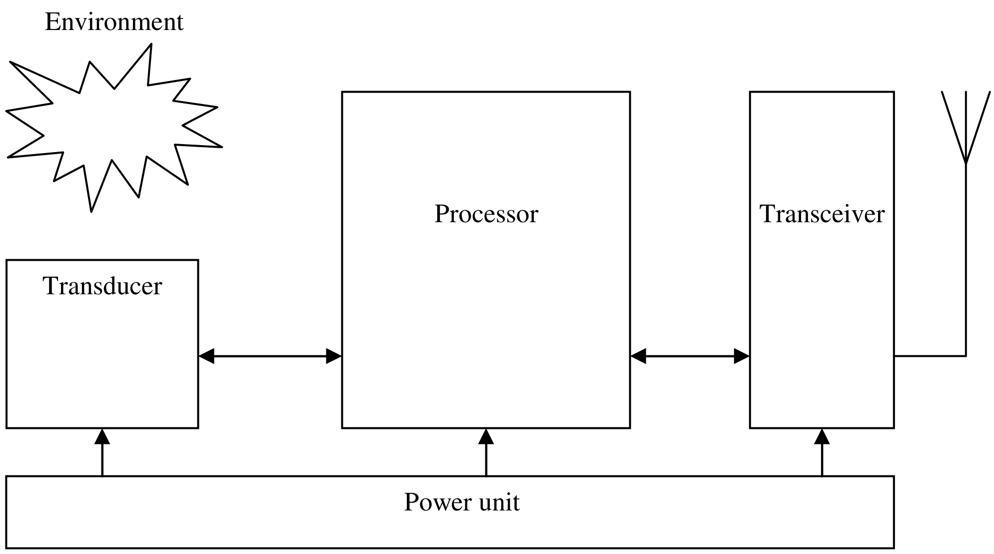

Figure 2.3. A bock diagram of a generic WSN architecture.

Sensor nodes are usually scattered across an environment/process/structure (Figure 2.4). Each node has the capability to acquire and route data back to a sink node [83]; the sink node is usually connected to existing networks which can backhaul that data to another appropriate location through the Internet, a cellular network, or a satellite network. After appropriate levels of data processing, the information is disseminated to end users through personal computers or mobile devices [84].

The roles most often implemented by WSNs include data gathering in continuous sensing environments and data reporting in event-triggered sensing. Data gathering for continuous sensing is used in applications that closely monitor a process or asset for analysis purposes [85, 86]. Data reporting for event-triggered applications only report when a pre-specified event occurs [87]. The application must define the exact function a WSN is designed to provide.

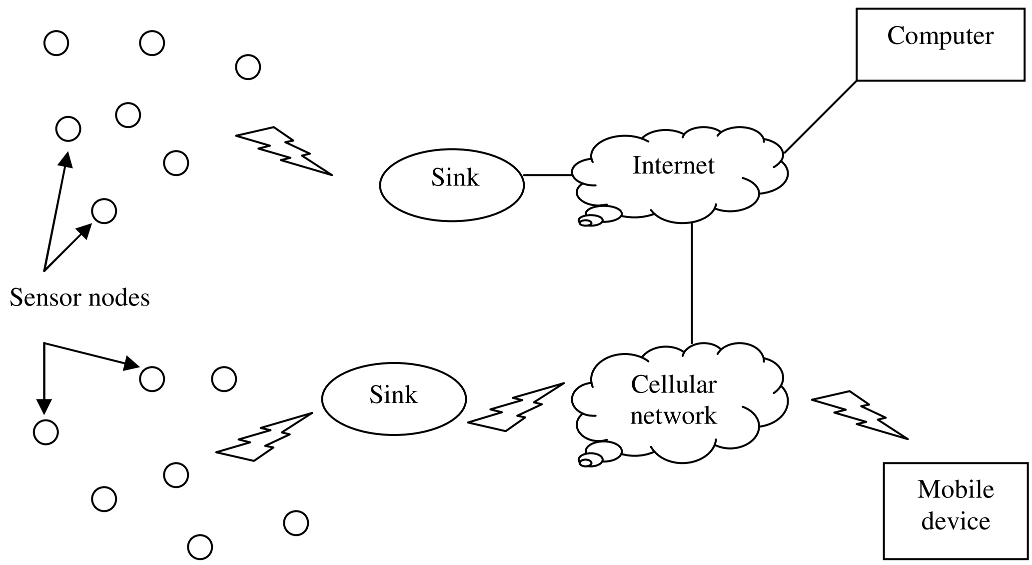

Figure 2.4. An example of communication routes from a sensor network to end users.

2.2.1 Transceiver

Due to the cost and market demand, WSN transceivers use low data rate chipsets and operate in ISM unlicensed bands. Many transceivers are equipped to form a self-organised network using multi-hop routes [9].

Although there is no doubt that advanced radio chip designs comprising multiple RF interfaces/antennas are useful in WSNs in order to achieve enhanced data rates, energy consumption renders their use questionable; a trade-off exists between power consumption and communication capability [88, 89, 90]. A bigger power resource can be used but the size of the sensor node and the ease of deployment becomes a major deployment constraint.

The WSN architecture can be referenced to the 5-layer profile through the OSI Reference Model as shown in Table 2.2; Physical, Data Link, Network, Transport, and Application layers [1].

Table 2.2. A 5-layer profile through the OSI Reference Model for WSN.

Layer	Data unit	Function
Application	Data	Network process to application
Transport	Segment	End-to-end connections and reliability
Network	Packet	Path determination and logical addressing
Data Link	Frame	Physical addressing and access control
Physical	Bit	Media, signal and binary transmission

2.2.1.1 Physical Layer

The Physical Layer is responsible for bit-level transmission between nodes, comprising basic hardware network transmission technologies. The layer defines the means of transmitting raw bits rather than logical data packets over physical links inter-connecting nodes. The bit stream may be grouped into code words or symbols and converted to a physical signal transported over a transmission medium.

The performance of this layer is affected by three major factors; hardware, software, and the medium of propagation. The hardware relates to factors such as the shape of antenna, antenna gain, and operating frequency; the software relates to modulation, bit synchronization, and transmission power. The wireless environment is however dynamic, uncontrollable and poses multiple challenges:

1. Free Space Path Loss (FSPL) [91]: the loss in signal strength of a radio wave owing to line-of-sight path transmission through free space. The loss depends on the frequency and the distance between transmitter/receiver and can be calculated as;

$$FSPL = 20\log_{10}(d) + 20\log_{10}(f) + 20\log_{10}(\frac{4\pi}{c}) \tag{2.1}$$

where $FSPL$ is the loss measured in dB, d is the distance between transmitter and receiver in metres, and f is the selected frequency in Hz. As the distance between transmitter/receiver increases, the path loss also increases. For the same transmission distance, the higher frequencies suffer a higher path loss (Figure 2.5).

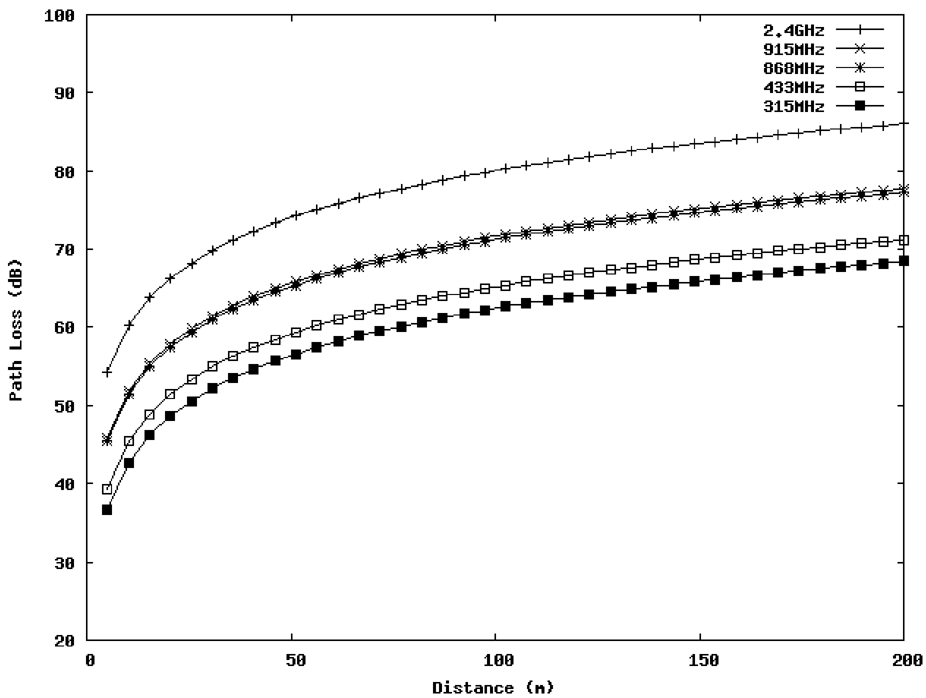

Figure 2.5. FSPL comparison at 315MHz, 433MHz, 868MHz, 915MHz, and 2.4GHz.

2. Attenuation [91]: represents any reduction in signal strength of a radio wave when the signal penetrates through solid objects, due for example to the absorption of the signal power. Attenuation can vary depending upon the structure of the object e.g. metal greatly increases the attenuation. Object thickness is also a contributory factor.
3. Scattering: is the diffusion of the radio wave when incident on a rough surface (Figure 2.6). Scattering is often most detrimental when the object size is on the order of the wavelength of the signal or less [91].

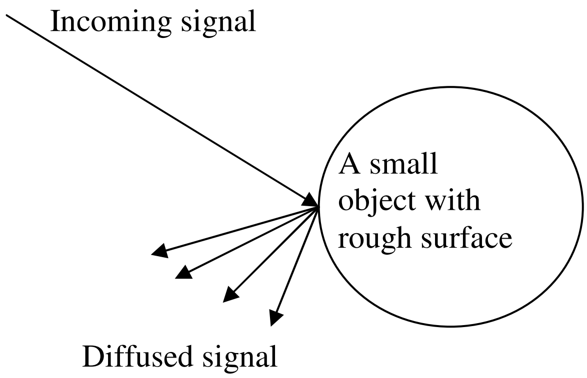

Figure 2.6. A schematic representation of scattering.

4. Reflection: is the change in direction of a radio wave when the signal encounters a surface relatively large to the wavelength of the signal. As shown in Figure 2.7, the radio wave may be reflected from various substances or objects as it traverses along the path between transmitting/receiving sites. The level of reflection depends on the material encountered. Smooth metal surfaces have good electrical conductivity and are efficient reflectors of radio waves [91]. When a radio wave is reflected from a flat surface, it suffers a phase shift; the shift in the phase of reflected radio waves is one of the major causes of signal fading [92].

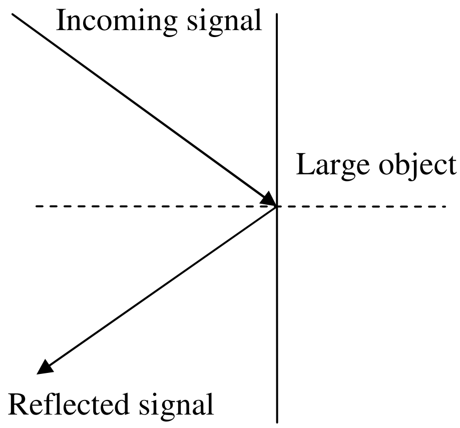

Figure 2.7. A schematic representation of reflection.

5. Refraction: is the change in direction of a radio wave due to a change in its speed occurring when a wave passes from one type of medium to another type of medium [93]. As shown in Figure 2.8, when a radio wave passes from a less dense to a more dense medium, the direction of the wave is changed, when $\theta_1 > \theta_2$.

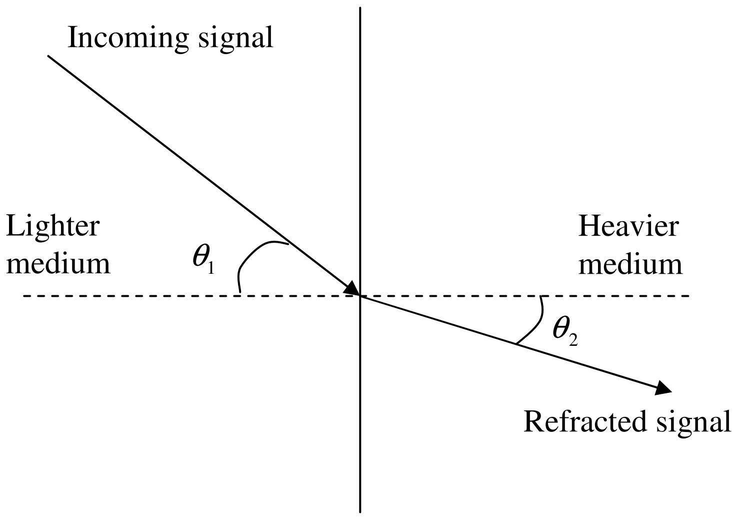

Figure 2.8. A schematic representation of refraction.

6. Diffraction: occurs when a radio wave encounters an obstacle and effectively 'bends' around that obstacle (Figure 2.9). The resultant change

in direction of part of the wave from the normal line-of-sight path makes it possible to receive energy around the edge of the obstacle [91].

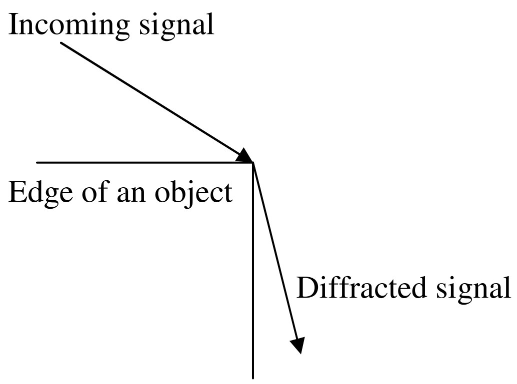

Figure 2.9. A schematic representation of diffraction.

7. Frequency Dispersion (Doppler Spread): motion of a node produces Doppler shifts of incoming received signals (Figure 2.10) [91]. This also occurs when a transmitter node is moving whilst transmitting; this phenomenon causes, amongst other effects [91], Inter-symbol interference (ISI) [93].

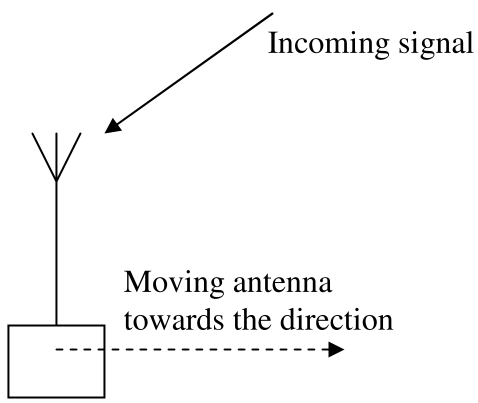

Figure 2.10. A schematic representation of Frequency Dispersion.

8. Ducting: confines a signal along a narrow "pipe". When a radio wave enters a duct, it can travel with low loss over great distances [93], acting in

a manner similar to a large optical fibre, trapping the radio wave within a high refractive index layer [92].

9. Interference: is the interaction of waves that are correlated or coherent with each other, either because they originate from the same source and/or because they are at the same or nearly the same frequency (same node characteristic from other network domains). Radio signals based on the prevailing conditions, may be in phase (constructive interference) or out of phase (destructive interference) [92].
10. Noise: are unwanted random signals impairing the quality of the wanted signal. A number of noise terms must be considered to obtain network performance, predominately generated on transmission and reception but can also originate from nodes/devices within the network [81].

2.2.1.2 Data Link Layer (DLL)

The Data Link Layer ensures that an initial connection has been established, segments output data into data frames, and handles acknowledgements from receivers confirming the data has been received successfully by analyzing bit patterns at standardised locations in the frames.

In WSN Media Access Control (MAC) design, energy-efficiency is always a main design criterion with other performance metrics considered as secondary objectives [94, 95]. Energy could be wasted due to several mechanisms at the MAC stage through packet collision, overhearing, control packet overhead and idle listening [94, 96].

Sensor-MAC (S-MAC) [97] is one of the earliest protocols proposed to solve the above issues at the expense of sacrificing per-hop fairness and latency. It operates at a pre-defined low duty cycle in a multi-hop networking environment. Nodes form virtual clusters based on common sleep schedules to reduce the control overhead and enable traffic-adaptive wake-up cycles. S-MAC uses in-channel signalling to avoid ‘overhearing’ unnecessary traffic [97] and instigates message (the collection of meaningful, interrelated units of data) passing to reduce contention latency for applications that require in-network data processing. Some studies [98, 99] highlight that S-MAC with fixed sleep and wake periods does not perform well under variable traffic loads. Solutions such as Timeout-MAC (T-MAC) [98] and the Traffic-Adaptive Medium Access Protocol (TRAMA) [99] improve on S-MAC by introducing an adaptive duty cycle; T-MAC ends an active period if no traffic occurs for a TA duration (the minimal amount of idle listening per frame) as shown in Figure 2.11 [98]; TRAMA uses an adaptive, dynamic approach based on current traffic patterns to switch nodes to low power mode [99].

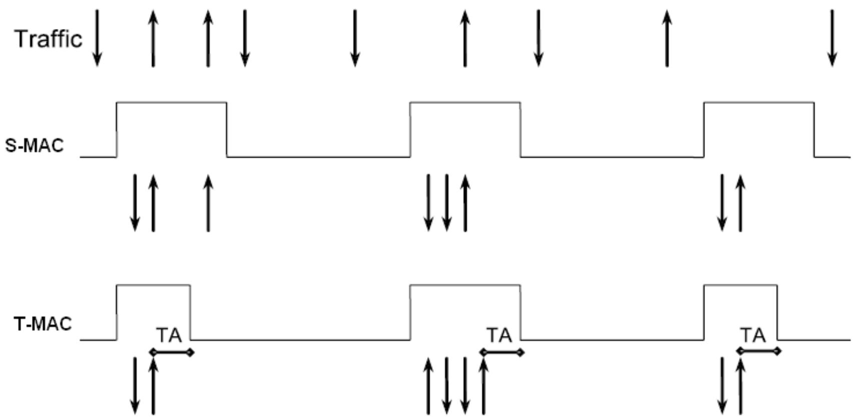

Figure 2.11. A comparison between S-MAC and T-MAC duty cycles.

In terms of simplicity, variants of Carrier Sense Multiple Access (CSMA) [11, 100], such as Berkeley-MAC (B-MAC) [100] and Sparse Topology and Energy Management (STEM) [101], are strong options. B-MAC uses a mechanism based on outlier detection to improve the quality of the communication channel. If a node detects an outlier during channel sampling, it declares the channel clear to send. If the node does not locate an outlier within five samples, it declares the channel is busy [100]. STEM separates the data transmission from the wake-up channel by using two different radios/channels. There are 2 types of STEM schemes: STEM Tone (STEM-T) and STEM Beacon (STEM-B) [101]. STEM-T uses a bit stream while STEM-B sends a series of beacon packets as the preamble. When there is no beacon collision, STEM-B has the advantage of providing lower setup latency and better energy efficiency. However, if frequent data transmission is not required, the channel sampling period for STEM-B must be longer than the inter-beacon interval; a shorter time is required for STEM-T to detect a wake-up tone.

Zebra MAC (Z-MAC) [102] is a hybrid MAC protocol combining TDMA [91] and CSMA principles to offset the weaknesses of each individual approach. Under low contention, Z-MAC behaves like CSMA and under high contention, like TDMA. The protocol uses knowledge of topology and loosely synchronized clocks to improve MAC performance under high contention. Under low contention, and when these hints are not reliable, the protocol behaves like CSMA. Z-MAC is also robust to dynamic topology changes and time synchronization failures that commonly plague WSNs [102].

2.2.1.3 Network Layer

The Network Layer is responsible for establishing paths for data transfer through a network, extending the DLL beyond the local network into an inter-network by providing the routing and forwarding of packets mechanisms.

In WSNs, routing protocols can be categorised into four main types; network structure, communication model, topology based, and reliability. Each type can be further classified as shown in Table 2.3 [83, 103].

Table 2.3. Categorisation of routing protocols and its further classification.

Types of routing	**Further classification**
Network Structure	- Flat-based - Hierarchical-based
Communication Model	- Query-based - Negotiation-based - Non-coherent-based - Coherent-based
Topology Based	- Location-based - Mobile agent-based
Reliability	- QoS-based - Multipath-based

A number of routing strategies have been reported; in flat-based routing, all nodes are homogeneous and provide the same functionality [104, 105, 106, 107]. In hierarchical-based routing, the nodes execute different roles distributed throughout the network [108, 109]. In query-based routing [110, 111], destination nodes propagate a query for data (e.g. sensing task) from a node throughout the network, and a node with the correct data sends the data matching the query back to the node that initiated the query. Usually these queries are described in natural language or high-level query languages [110]. Negotiation-based routing [112] uses high-level data descriptors in order to eliminate redundant data transmissions through negotiation. In non-coherent data processing routing [113], nodes will locally process the raw data before transmitting to other nodes for further processing. The nodes that perform further processing are referred to as aggregators [56]. In coherent routing [56], data are forwarded to aggregators after minimal processing, typically tasks like time-stamping and duplicate suppression. Location-based routing [114] requires location information to determine the data forwarding route; location information could be obtained through the Received Signal Strength Indicator (RSSI) [115], relative coordinates of neighbouring nodes [116], or external devices such as the Global Positioning System (GPS) [117]. Mobile agent-based routing [118] uses agents to explore and create routing paths. When agents discover shorter or more efficient paths, the paths in routing tables are updated accordingly. Quality-of-Service (QoS)-based routing [119] requires the network to balance energy consumption and data quality. In particular, the network has to satisfy certain QoS metrics such as delay, energy consumption, bandwidth during the delivery of data. Multipath-based routing [120] utilises multiple paths rather than a single path in order to enhance network performance. Maintaining multiple paths between source and destination increases energy consumption and an overhead is generated in the level of traffic [121]. A balance between network reliability and path maintenance overhead is required for this kind of routing.

In general, two main operating scenarios drive the selection of an appropriate WSN routing strategy - static and mobile topologies. For static topologies, the routing path can be optimised [122] while for mobile topologies, frequent broadcasting of beacons is required to discover the state of the network and its constituent nodes in order to implement effective routing [123].

2.2.1.4 Transport Layer

The Transport Layer is a group of protocols responsible for encapsulating application data blocks into data units referred to as datagrams or segments suitable for transfer through the network to the destination host, or managing the reverse transaction by abstracting network datagrams and delivering their payload to an application. Thus the Transport Layer protocols establish a direct and virtual host-to-host transport capability for applications [124].

The Reliable Multi-Segment Transport (RMST) [125] is one of the earliest transport protocols designed for WSNs. RMST is developed to operate in conjunction with Directed Diffusion Routing [110], acting as a filter to any diffusion node improving reliability but with no real time guarantees.

Pump-Slowly, Fetch-Quickly (PSFQ) [126] is another transport protocol designed for WSNs aimed at improving network reliability. The protocol distributes data from a source node by pacing data at a relatively slow speed (so-called pump slowly) but allowing nodes that experience data loss to fetch any missing segments from their local immediate neighbours aggressively (so-called fetch quickly). Lost messages are detected when a higher sequence number than expected is received at a node, triggering the fetch operation. The solution is able to achieve loose delay bounds while minimising the lost recovery cost by using localised recovery of data amongst immediate neighbours.

Event-to-Sink Reliable Transport (ESRT) [127] is a transport protocol not only developed to achieve reliable event detection but also to enable congestion control. ESRT runs on the sink, with sensor nodes subject to resource constraints. Protocol operation is governed by the prevailing network state based on the reliability achieved and the congestion condition of the network. If the 'event-to-sink' reliability is lower than required, ESRT adjusts the reporting frequency of source nodes aggressively in order to reach the target reliability level as soon as possible. If the reliability is higher than required, then ESRT reduces the reporting frequency conservatively in order to conserve energy while still maintaining reliability.

In summary, WSN transport protocols are usually classified into three categories: reliability support, congestion control, and a combination of both.

2.2.1.5 Application Layer

WSN application environments are often inhospitable or difficult to access. Executing local maintenance tasks performed by technicians or users is challenging and on occasion almost impossible; therefore, a pressing need to manage the deployed nodes exists. A study [128] highlights that traditional management protocols, such as the Simple Network Management Protocol (SNMP) [129] and Ad-hoc Network Management Protocol (ANMP) [130] are not suitable implementations for WSNs due to the energy, hardware, and software restrictions.

Sensor Management Protocol (SMP) [131] is a general management protocol for WSNs. It provides functions needed to perform administrative tasks such as introducing procedures related to data collection, exchanging data related to location finding algorithms, time synchronisation, node mobility, controlling node sleep cycles, data querying, network reconfiguration, and security [131].

Task Assignment and Data Advertisement Protocol (TADAP) is a management protocol for task assignment and data dissemination [132]. The protocol allows users to post their interest to a sensor node, a subset of nodes, or the entire network. This interest can be a certain attribute of the phenomenon or a triggering event. Another application for this protocol is the advertisement of available data, in which sensor nodes advertise available data to users, and the users query the data of interest.

Sender Query and Data Dissemination Protocol (SQDDP) enables user applications by providing interfaces to issue queries, respond to queries, and collect incoming replies [110]. These queries are attribute-based such as “the places that sense temperature higher than 40ºC” or location-based naming such as “temperatures read by the nodes in area N1”.

Other examples of application specific protocols developed to fulfil the requirement of various application needs for WSN have been object tracking [133], security [134], and multimedia services [135].

2.2.2 Transducers

WSN implementations are not restricted to sensors such as light, temperature, humidity, and accelerators but can be embedded with actuators such as servo motors for controlling purposes [136, 137]. Although the cost of microcontrollers and radio chipsets are low, adding sensors and actuators bring about a cost penalty. Transducer interfacing is not restricted to using GPIO but can be integrated through ADC, PWM, UART, USART, I^2C, or SPI. Some transducers require higher voltages to operate and need to be powered using a separate source, compromising the overall size of the sensor node. Therefore, the selection of the most appropriate transducer is a major design criterion.

2.2.3 Processors

WSN nodes are essentially low-power computing platforms, the core being 8-bit or 16-bit microcontrollers. Since WSNs are limited by the available power and are low cost distributed systems, the use of advanced and complex processors is not aligned with these principles. Recent designs combine the processor and radio units into a single chip, such as the CC1110/CC1111 [82].

2.2.4 Power Units

WSN nodes are most often powered by batteries such as non-rechargeable AA format, coin lithium, or a rechargeable pack. Although sensor nodes are amenable to energy harvesting options, improper management of the power consumption will result in compromised network functionality as the time the node takes to harvest energy may be longer than the time over which the node dissipates that energy [138]. An additional device is normally needed to generate the harvested energy e.g. solar panel, and this translates into both a cost penalty and an increase in form factor [75]. Therefore, the selection of the most appropriate power unit is a key consideration in WSN designs.

2.3 Cross-Layer Design

Cross layer design [139] allows direct connection between the layers of the system through sharing of key information between non-adjacent or adjacent layers [140]. One of the main drivers for cross-layer approaches is the limitations of the layered architecture which although serves well for wired system development, is not wholly appropriate for wireless networks [139]. There are several approaches to cross layer architecture design: creation of new interfaces (upward information flow [139], download information flow [141, 142], back-and-forth information flow [143, 144]), merging of adjacent layers [123], design coupling without new interfaces [145], and vertical calibration across layers [146].

In general, cross layer design is motivated by three main drivers: the unique problems created by wireless links, the possibility of opportunistic communication on wireless links, and new modalities of communication offered by the wireless medium. The risk of cross layer design is it may result in the degradation of overall system or connection performance [147]. It also creates inseparable coupling between layers at the expense of performance improvement [147].

In sensor networks, cross layer design has been shown to improve data traffic flow performance e.g. Cross-Layer Protocol (XLP) [148] integrates the functionalities of all layers from physical to transport through a protocol in order to achieve congestion control, routing, and MAC in a WSN. This approach significantly improves performance, outperforming traditional layered protocol architectures in terms of both network performance and implementation complexity. The Address-light, Integrated MAC and Routing Protocol (AIMRP) [123] is another protocol that combines the MAC and network layers to achieve energy savings in mobile environments. The direct connection from the MAC to the application layer in order to achieve excellent performance for certain applications has been reported by [149].

2.4 Machine Learning

Machine Learning [26, 150] is a sub research area of Artificial Intelligence [151]. It is normally classified based on problem domains into 3 major categories; classification and regression, acting and planning, and interpretation and understanding [152].

Classification determines how to assign a test case to one of a finite set of classes. Regression is used to predict a case value based on partial or all historical data inputs. There are 3 types of learning methods used to solve the issues of classification and regression; supervised learning [153], unsupervised learning [154], and semi-supervised learning [155]. Supervised learning determines an output based on a given training case with associated classes or values for the attribute to be predicted. Unsupervised learning decides an output based on a given training case without any associated class information or any specific attribute singled out for prediction. Semi-supervised learning falls between these two approaches; in this learning approach, partial of the training instances come with associated classes or values for predicted attributes.

Acting and planning approaches are used to optimise problem solving, planning, and scheduling tasks. These approaches address the selection of actions or plans based on an agent with knowledge in a given world state. Several approaches such as learning apprentice [156], adaptive interface [157], programming by demonstration [158], and behavioural cloning [159] can be used to address issues related to formulation of action learning. For planning, methods such as reinforcement learning [160] and learning from problem solving and mental search [161] can be considered.

Lastly, interpretation and understanding approaches aim to interpret and understand situations, scenarios, events, or environments. For interpretation approaches, more constructive observations are needed by combining a number of separate knowledge elements to explain the data. Therefore, models are required to explain the data in terms of deeper structures. Methods such as induction over explanations [162], constructive induction [163], and explanation-based generalisation [164] can be useful for interpretation. Understanding approaches induce its own explanatory structures from regularities in the data and then utilised to clarify new test instances. Approaches such as natural language [165] and theory revision [166] are the examples to be considered.

2.5 Cognitive Technique Applied in WSNs

The term "cognitive" in the Cambridge dictionary is "an adjective related to thinking or conscious mental processes" [167]. From the networking perspective [168, 169], cognition is used in association with a technology that operates inside a complex environment (for example the congested radio frequency spectrum), observes it, makes behaviour choices, and receives feedback from it, all the while learning i.e. assembling a data set that will help determine future behaviours based on past and current feedback.

The continually rising number of users and capacity requirements of radio systems are fuelling an ever increasing demand for spectrum. Cognitive Radio [14] offers a tempting solution to this problem by proposing opportunistic usage of frequency spectrum bands not occupied by licensed users. Cognitive radio architectures were first proposed by Mitola [170] in 1995 addressing the organization of the knowledge of the radio domain into data structures process able in real-time that integrate machine learning and natural language processing technology into software radio. The features embedded in the architecture are derived from cognitive radio use cases, such as inferring user communications context, shaping access-network demand, and realizing a management protocol for real-time radio spectrum. This architecture is based on the set-theoretic ontology of radio knowledge defined in the Radio Knowledge Representation Language (RKRL) [171] layered on top of Software Defined Radio (SDR) [172, 173]. Three on-line tasks are required to be executed in cognitive radio: 1) radio-scene analysis (estimation of interference of the radio environment and detection of spectrum holes), 2) channel identification (estimation of Channel-State Information (CSI) and prediction of channel capacity for use by the transmitter), and 3) transmit-power control and dynamic spectrum management. Cognitive radio is expected to meet 4G requirements at up to 1Gbps throughput with multiple asynchronous concurrent data streams on mobile handsets, base stations and small cells [174].

Cognitive networks [15] extend the idea of cognitive radio to improve resource management, QoS, security, access control, and many other network goals. In a network, a cognitive process can perceive current network conditions, plan, decide, and act on the basis of those conditions. The network is able to learn to make future decisions taking into the consideration the end-to-end goals. The implementation of the mechanism is achieved by interfacing the cognitive process on top of the Software Adaptable Network (SAN), similar to cognitive radio [168]; the entire cognitive network framework comprises End-to-end Goals, Cognitive Processes, and SAN. The goals or requirements determine cognitive behaviours by identifying, prioritizing, and weighting the user requirements of the network. The Cognitive

Process consists of three components: the specification language, cognition layer, and network input [168], the SAN containing Network Application Programming Interface (API) and Modifiable Network Elements. A Network API is an interface implemented through middleware between the user or application and the network elements, including the network stack. Modifiable Network Elements [168] include any object or element used in a network able to be modified for control purpose.

Cognition in WSNs has been cited in a number of papers [175, 176]. However the implementation of cognition in WSNs remains limited and the opportunity to develop solutions to key design issues such as network lifetime maximization, energy efficient routing, the reliability of event detection and transfer, optimization of multiple or conflicting objects, and application-specific design exists. Early reports on potential cognitive frameworks for WSNs simply extend a spectrum sensing capability on to the existing architectures [177, 178], while others introduced a cognitive feature into a specific scenario, similar to context-aware applications [18, 179, 180].

In general, there are two high level categories of cognition in networking;

- "cognitive radio", focusing on efficient bandwidth usage, sharing the spectrum among primary and secondary users, through the use of cost effective spectrum sensing algorithms [181].
- all the layers of the network together optimize the application objectives [19, 182], referred to as "cognitive networking". In the Thesis, the term "cognition in networking" is best aligned to this category.

2.6 Standardisation

Most standards in wireless communications are defined by the IEEE [183]. For the wireless Internet infrastructure, there are three on-going streams under the IEEE 802 LAN MAN Standards committee [184], namely IEEE 802.11 [185], IEEE 802.15 [186, 187, 188, 189], and IEEE 802.16 [190].

ZigBee is an industrial standard designed for a series of high level communication protocols from network to application layers using small, low-power digital radios based on the IEEE 802.15.4 standard for Personal Area Networks (PANs) [191, 192]. ZigBee devices are able to form mesh networks to transport data over longer distances through ad-hoc multi-hop communications with decentralised control. ZigBee is purposely designed to be much simpler and less expensive than other existing WPANs, such as Bluetooth. The current ZigBee protocols support beacon and beaconless networking.

6LoWPAN is a standard defined to allow IPv6 packets to be transmitted and received over IEEE 802.15.4 based networks [193, 194]. The 6LoWPAN concept originated from the philosophy that "the Internet Protocol could and should be applied even to the smallest devices" [195]. The standard has wide applications such as automation, home entertainment, office, and factory environments. One of the popular applications of 6LoWPAN is in Smart Grids [196], enabling energy meters and other devices to form micro mesh networking prior to transmitting data to the billing system using the IPv6 backbone.

Standardisation is also being pursued for Cognitive Radio and Networks, Dynamic Spectrum Access, and Coexistence [197]. IEEE 802.22 [198] is the first cognitive radio-based standard with tangible operating frequency bands. Several standardization organizations such as the SDR Forum [199] and the International Telecommunications Union-Radio Sector (ITU-R) [200] are developing standards in this area. The IEEE 802.22 is an extension of IEEE 802.16 and was initiated in 2004 to specify the air interface, including MAC and PHY layers, of fixed point-to-multipoint wireless regional area networks operating in the VHF/UHF TV broadcast bands between 54MHz and 862MHz. The unique requirements of operating on a strict non-interference basis in spectrum assigned to, but unused by, the incumbent licensed services requires a new approach using purpose-designed cognitive radio techniques that permeate the PHY and MAC layers.

Another standard aimed at defining Cognitive Radio operation is IEEE SCC41 [201]. IEEE SCC41 addresses issues related to the deployment of next generation radio systems and advanced spectrum management. IEEE SCC41 was preceded by the IEEE 1900 task force jointly established in 2005 by the IEEE Communications Society and the IEEE Electromagnetic Compatibility Society. Finally in 2007, the IEEE created a new governing body for all IEEE 1900 standards and named it SCC41 on Dynamic Spectrum Access Networks.

Although rudimentary cognitive capabilities (detection of other signals with application of dynamic frequency assignment, power control, and other techniques in response) already exist, the view remains that existing standards have not yet matured to the point of being truly cognitive. But the promise and potential value of such techniques is clearly recognized, and almost all existing and future wireless standards are attempting to incorporate cognitive radio, dynamic spectrum access, and coexistence dimensions.

2.7 Summary

Wireless Sensor Networks have been the subject of development for many years motivated initially by military applications. Over the last twenty years, as technology options evolved in terms of storage, processing and wireless transmission cost, the range of applications blossomed; today a number of WSN motivated products are bringing significant commercial benefits to key industry sectors.

Modern WSNs provide a rich mix of functionalities such as UWB, OFDM, MIMO, self-healing, to note but a few. WSNs are thus able to operate with more options driven by the application. The evolution of WSNs in terms of functionality and applications is summarised as in Table 2.4.

Table 2.4. Evolution of WSNs in terms of functionality and applications.

Evolution Stage	Functionality	Application
Initial stage	Low processing power, small storage, short range optical communication, self-organised	Realisation of “smart-dust” concept in military applications
Early stage	Low processing power, small storage, longer communication range using simple radio transceiver, self-organised, simple sensing capability	“Proof of concept” and small scale deployment in various monitoring applications
Modern stage	Advanced processing power, higher storage, advanced radio communication to support higher data traffic, self-organised, self-healing, complex sensing capability	Realisation of IoT and large scale deployment

The rapid progress in WSN technologies - which have provided increased processing power/storage capacity at no cost penalty - has stimulated the next phase of WSN research, that of embedding intelligence within the network. Cognitive techniques have the potential to enhance WSN functionality facilitating extensions of the applications spectrum to encompass operation in complex environments, observation of network states, executing behaviour choices based on the prevailing state, and providing feedback whilst formulating future behaviours. Cognitive behaviour should not be confined at the spectrum level but should enable the network to achieve end-to-end goals through efficient resource management. The evolution of cognition in WSNs in terms of functionality and application is summarised as in Table 2.5.

Table 2.5. Evolution of cognition in WSNs in terms of functionality and applications.

Evolution Stage	Functionality	Application
Cognitive Radio (from 1995)	Radio-scene analysis, channel identification, transmit-power control and dynamic spectrum management	Multiple asynchronous concurrent data streams on mobile handsets, base stations and small cells
Cognitive Networks (from 2005)	Resource management, QoS, security, access control, and other network goals	Identifying, prioritizing, and weighting the user requirements of the network
Cognitive Sensor Networks (from 2008)	Spectrum sensing and context-aware	Network lifetime maximization, energy efficient routing, the reliability of event detection and transfer, optimization of multiple or conflicting objects, application-specific design

In general, there are two categories of cognition in networking; “cognitive radio”, focusing on efficient bandwidth usage through sharing of the spectrum, and cost effective spectrum sensing. In the Thesis, the term “cognition in networking” is more appropriate and considers all layers of the network to optimize the application objectives.

Chapter 3: CogWSN Architecture and Decision Process

3.1 Definition of Cognitive Wireless Sensor Network (CogWSN)

The term “cognitive” is an adjective related to thinking or conscious mental processes [167]. From the networking perspective [182], cognition is used in association with a technology that operates inside a complex environment (for example the congested radio frequency spectrum), observes it, makes behaviour choices, and receives feedback from it, all the while learning i.e. assembling a data set that will help determine future behaviours based on past and current feedback.

In the Thesis, CogWSN is proposed through the integration of two technologies: cognitive artificial intelligence and wireless sensor networks. Thus, CogWSN is defined as a networked group of sensor nodes capable of sustaining performance in a dynamic environment using an embedded cognitive capability, while meeting user requirements.

3.2 Hardware; Issues and Limitations

Since the founding principles of WSNs are built upon harnessing small and inexpensive components, bounds in the degree of operation are inherent. The limitations are [1, 2, 9, 10, 11, 64]:

1. A WSN is a packet-radio network and its radio interface cannot normally transmit and receive at the same time.
2. Data are aggregated at a Base Station (BS) which also issues commands.
3. The BS is connected to a PC using a wired cable and effectively enjoys unlimited power.
4. By default, a WSN device is normally powered using a small capacity, non-rechargeable battery.
5. The radio operates in the ISM band.
6. A single, low gain antenna is normally used.

7. Normally several PANs are in operation, not coordinated.

3.3 CogWSN's Decision Process

A Layered Reference Model of the Brain (LRMB) [23] has been proposed as the basis for an integrated framework for modelling the behaviour of the human mind. LRMB encompasses 37 cognitive processes classified in six layers known as sensation, memory, perception, action, meta-cognition, and higher cognitive layers from the bottom-up. The higher layers enable more complicated and diversified life functions to be implemented, totalling 16 cognitive processes. In the CogWSN research presented here, the Problem Solving cognitive process is selected as the decision process.

According to Polya [202, 203], 4 steps must be followed when solving a mathematical problem;

1. Understand the problem
2. Devise a plan
3. Carry out the plan
4. Look back

Problem Solving [204], as defined in LRMB, is a higher layer cognitive process of the brain that seeks a solution for a given problem or finds a path to reach a given goal. Within the CogWSN concept, a Problem Solving cognitive process enhances the functionality of the WSN to have the ability to observe the environment/process, derive a plan, execute the plan, and provide feedback after execution.

For CogWSN, the decision process following the Problem Solving cognition from LRMB embodies the Polya principle. Hence the CogWSN decision process is formulated through 4 phases, illustrated in Figure 3.1.

1. Observe; monitor and identify if there is a problem
2. Plan; derive a plan to solve the problem

3. Implement; implement according to the plan
4. Evaluate ; feedback whether the problem has been solved

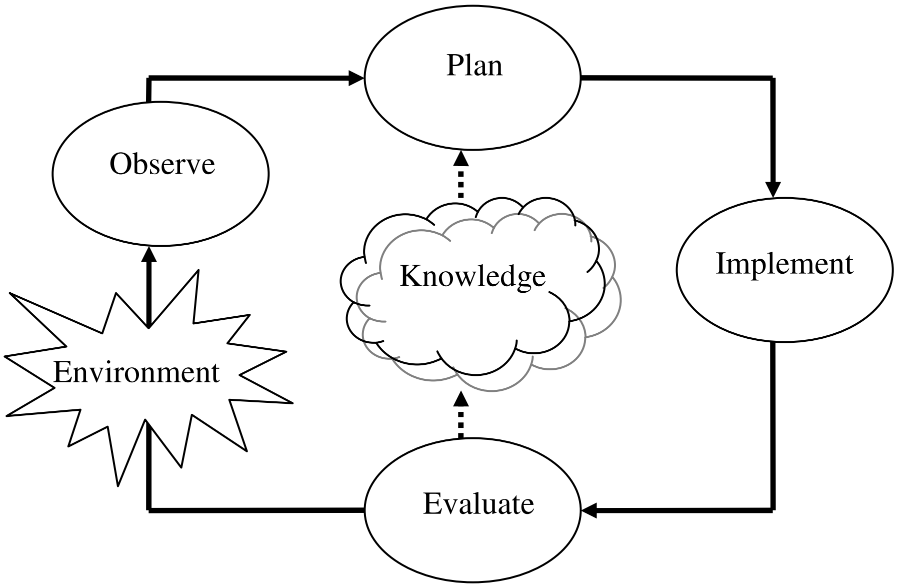

Figure 3.1. CogWSN decision process.

In the process, observation is performed to identify any potential issue in the monitored environment. If an issue is found, a plan is derived based on existing knowledge; however initial knowledge can be pre-installed before the deployment. Then an action is taken according to the derived plan. Finally, an evaluation is carried out by monitoring the concomitant changes in the environment for a short period to determine whether the action taken yields benefits. The result of the evaluation is recorded into the knowledge system for future reference. It should be noted that the situation may be improved or degraded as a consequence of the action taken, and the process continues to loop by reverting to the Observe phase.

3.4 CogWSN Architecture

Each component in the WSN other than the processor unit (holds the required operation and all virtual modules) contains information about its operating status and actuating function. All information is embedded into virtual modules integrated to inter-operate with the required task specified by the user. CogWSN operate co-operatively between the transceiver, transducer, and power supply virtual modules. Each module contains two elements defined as State Information (SI) which stores the information on the operating state and Tuneable Function (TF) which defines the actuating function (Figure 3.2).

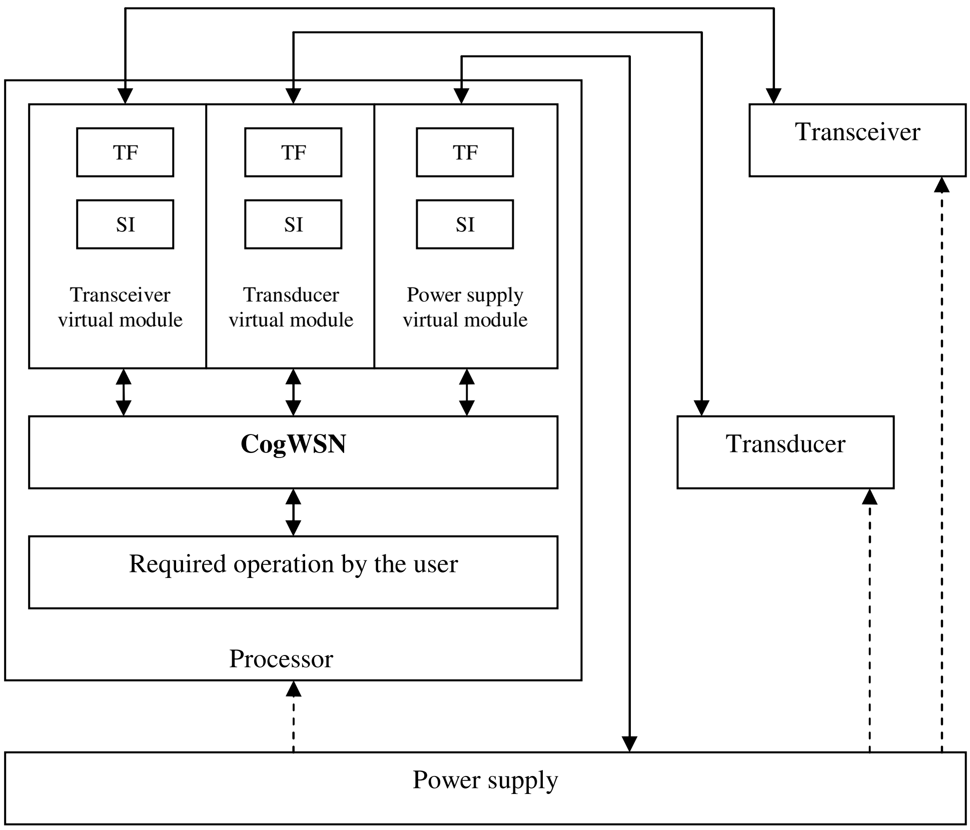

Figure 3.2. CogWSN architecture.

SI and TF subcomponents in the virtual modules can be defined as shown in Table 3.1. The CogWSN module coordinates all virtual modules to the required operational goal specified by users.

Table 3.1. Definition of the virtual modules.

Virtual Module	State Information (SI)	Tuneable Function (TF)
Transceiver	RSSI, link quality, data error rate, data throughput, channel congestion, beacon rate, parent node communication slot, nodes communication slot, routing information, data buffer, priority packet information, data transfer rate	Power transmission, antenna selection [205], channel switching, radio sleep/wake control, beacon rate control, parent node communication slot control, children nodes communication slot control, routing control, priority packet control [206], data transfer rate control
Transducer	Data sampling rate, threshold check, sensor/actuator status	Data sampling rate, sensor sleep/wake control, sensor offset, actuator sleep/wake control, actuator control
Power supply	Voltage of the power supply, duty cycle of activities	Tune to secondary power supply

3.4.1 Transceiver Module

Transceiver module captures the performance of the communications in terms of reliable connectivity, energy-efficiency, channel utilisation optimisation, error rate, and data throughput. To demonstrate the functionality of this module within CogWSN, a case study is presented where the SI is the RSSI and TF is the power transmitted [19].

The goal is to fine tune the transmission power so that a packet can be received successfully at the receiver with signal strength between -85dBm to -65dBm but not at the expense of additional energy. For example, if a packet can be received successfully at a destination node using 0.1mW, transmitting the same packet using 1mW is wasteful of power, impacting negatively on overall network lifetime. For this scenario, a test bed is established to evaluate the performance of the module. It should be noted that the receiver sensitivity does not exceed -94dBm [207].

The foundation of the decision process is triggered by observing the operating environment and making decisions based on rule-based knowledge. When a decision has been derived, a series of actions is taken in an attempt to manage prevailing network inefficiencies. It is highly desirable to have the problem resolved at the first attempt but commonly, a progression of subsequent actions is required to provide further enhancements; subsequent decisions are made through feedback from previous actions. In each cycle, the degree of improvement can be monitored from the feedback. The cognitive module assesses the changes in conditions and a score is given to each pairing of 'action and its result'. Therefore, when a similar condition comes to pass in the future, the module naturally applies the action with the highest score.

The experiment is conducted using a MICAz platform [208]. The cognitive module is coded into a MICAz node at the transmitter to adjust its transmission power so that signal strength is maintained at an acceptable level for successful reception. The receiver is required to acknowledge every packet received with an ACK, containing the signal strength of the received packet. This forms the basis of the observation phase with which radio link quality can be assessed. To verify the correctness of the performance, two experiments are conducted, shown in Figure 3.3 and Figure 3.4.

In Figure 3.3, 2 nodes are placed close to each other 50cm apart. A node is configured as transmitter and another node acts as receiver. It is observed that the transmission power of the transmitter has adjusted 3 times. After the first adjustment, the transmission power is reduced from 1mW to 0.316mW whereupon the received signal strength drops from -56dBm to -60dBm; the signal strength is still above -65dBm, indicating unnecessary power overuse. Hence, further fine tuning is conducted and the decision process eventually stops when the transmission power is reduced to 0.0316mW and received signal strength of -70dBm. In this case study, the benefit of embedded cognition is clear as it reduces the power consumption by 96.8% whilst maintaining acceptable link quality.

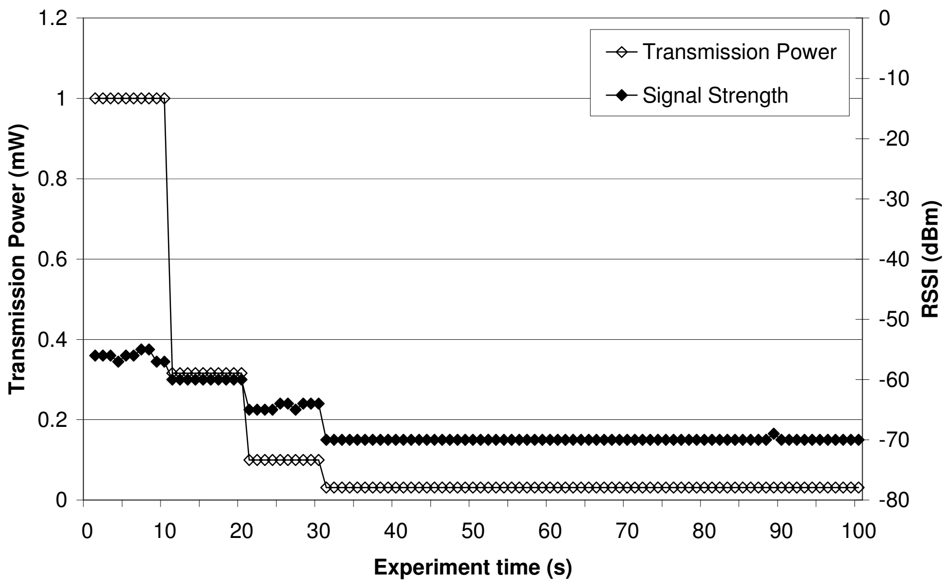

Figure 3.3. Transmission power adjustment to prevent energy overuse.

In Figure 3.4, an experiment is established to evaluate the network response to a bad radio link; 2 nodes, the transmitter and the receiver, are separated at 15m. Here, the transmitter is initially set to very low transmission power (0.00316mW). After several retries, the transmitter autonomously increases the transmission power so that packets reach the receiver at an appropriate signal strength.

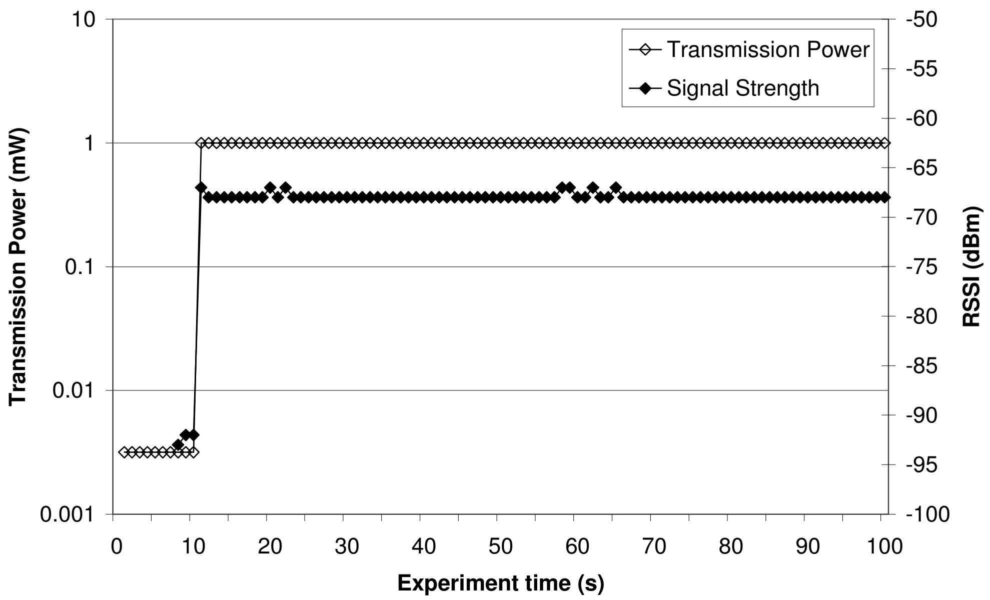

Figure 3.4. Transmission power adjustment to improve link quality.

The experiments in Figure 3.3 and Figure 3.4 verify the correctness of the performance of the transceiver virtual module where the goal is achieved after some power transmission adjustments.

3.4.2 Transducer Module

Sensor nodes may be furnished with a function that controls when to report data at a rate determined by the prevailing network condition e.g. available power unless over-ridden by a request from the user. In limited power resource scenarios, it is important to ensure data are transported in an energy-efficient manner; at the transducer module, data can be abstracted for transport and reformed on arrival at the base station [209].

An experiment is conducted to demonstrate the operation of this module within CogWSN; SI in this case is the data reporting rate and TF is the parameter representative of the degree of sensor sensitivity. Sensor sensitivity is determined through the Arithmetic Progression Threshold [210] and Geometric Progression Threshold algorithms [211].

In practice, configuring the sampling rate for data capture directly under user control is non-trivial. Thus to circumvent this, the approach adopted is to apply some intelligence at the sensor node that governs the reporting of a processed stream of original data without compromising the application. In this example, the amount of data to be reported is reduced by transporting modified, abstracted samples only when the difference between the current and the previous reported data samples exceeds pre-defined thresholds. The thresholds are set according to Equation 3.1 and Equation 3.2.

$$th_n = a + (n-1)s \tag{3.1}$$

$$th_n = ar^{n-1} \tag{3.2}$$

where a is an assigned value based on the degree of sensor sensitivity, s is the sensor sensitivity value obtained from its datasheet and n is a variable incremented by 1 after each threshold comparison until a maximum value is reached. After a data sample is reported, n is reset to 0. Assume that the previous reported sample has a value of x_0 and after i-1 unreported sampling phases, the current reading of the sensed data is x_i. The algorithm illustrated in Figure 3.5 is applied to determine whether or not to report the data sample.

```
increase i by 1
n ← min(i, n_max)
if arithmetic progression is selected, then
  th_n ← a + (n – 1) × d
if geometric progression is selected, then
  th_n ← a × r^(n-1)
if |x_i – x_0| ≥ th_n, then
  report the values of i and x_i
  x_0 ← x_i
  i reset to 0
```

Figure 3.5. Arithmetic Progression and Geometric Progression Threshold algorithms.

For comparison purposes, another three thresholding methods are considered; fixed threshold value [212], difference between current sample and previous sample values [213], and difference between current sample and previous reported values [214]. The experiments are designated as:

1. Threshold methodology 1; Complete data
2. Threshold methodology 2; Threshold value >= 25
3. Threshold methodology 3; Difference between current sample value and previous sample value by 0.1
4. Threshold methodology 4; Difference between current sample value and previous report value by 0.1
5. Threshold methodology 5; Arithmetic Progression Threshold with $a = 0.1$, $d = 0.1$, $n_{max} = 10$
6. Threshold methodology 6; Arithmetic Progression Threshold with $a = 0.5$, $d = 0.05$, $n_{max} = 10$
7. Threshold methodology 7; Geometric Progression Threshold with $a = 0.1$, $r = 1.3$, $n_{max} = 10$
8. Threshold methodology 8; Geometric Progression Threshold with $a = 0.5$, $r = 1.1$, $n_{max} = 10$

Figure 3.6, Figure 3.7, and Figure 3.8 show the abstraction with burst data, slowly incremented and decremented data, and randomly generated data, respectively.

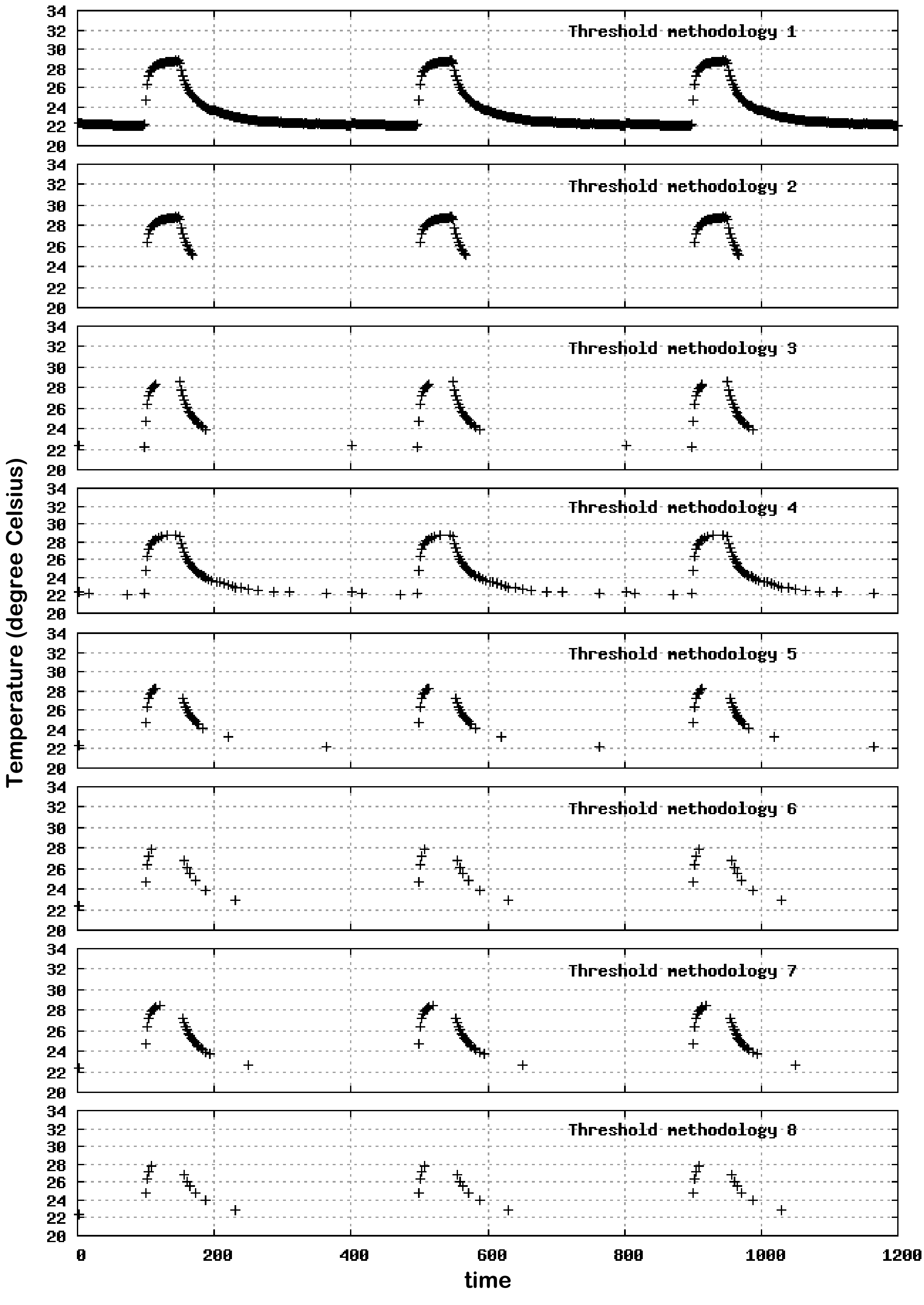

Figure 3.6. Data abstraction with burst data.

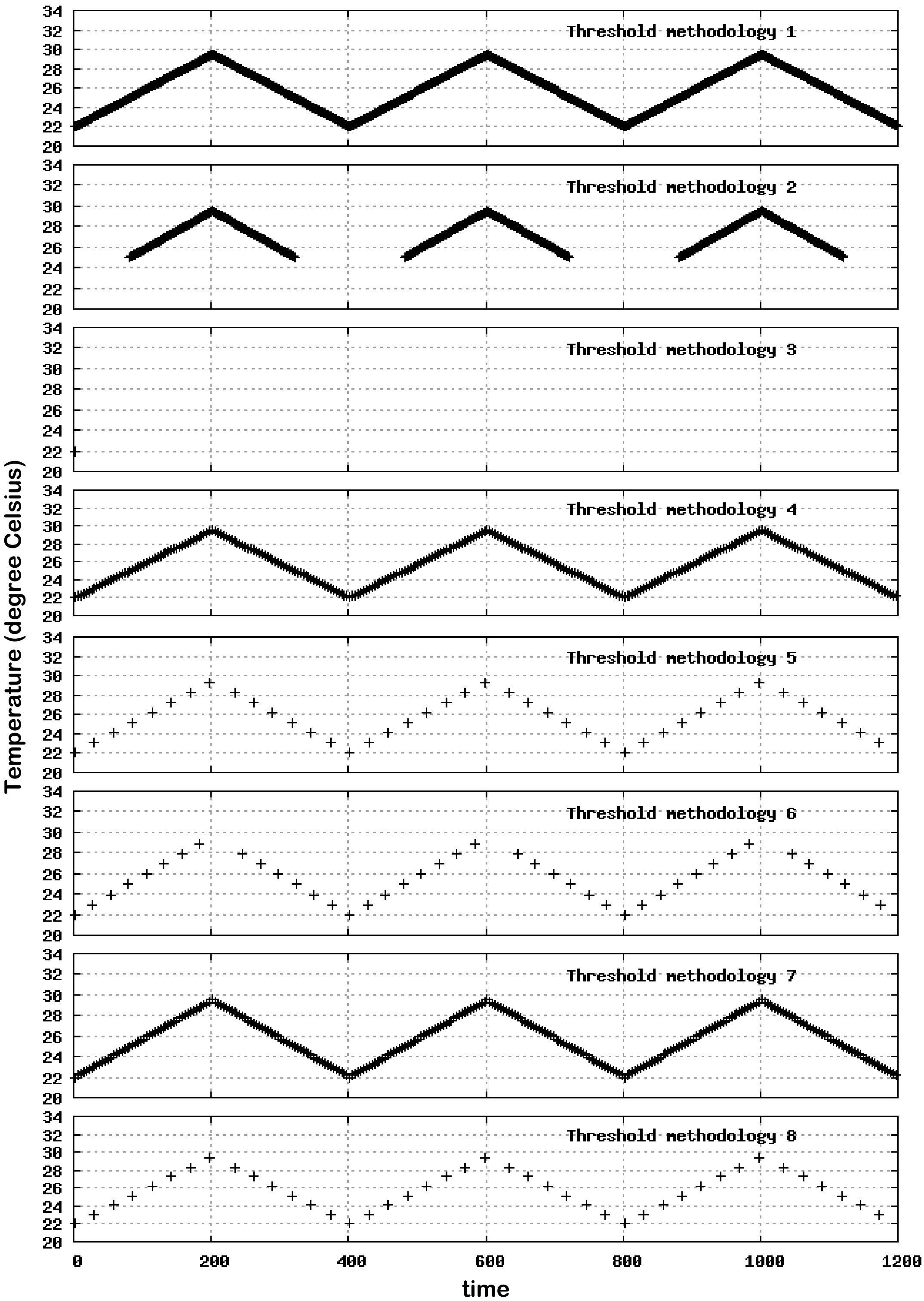

Figure 3.7. Data abstraction with slowly incremented and decremented data.

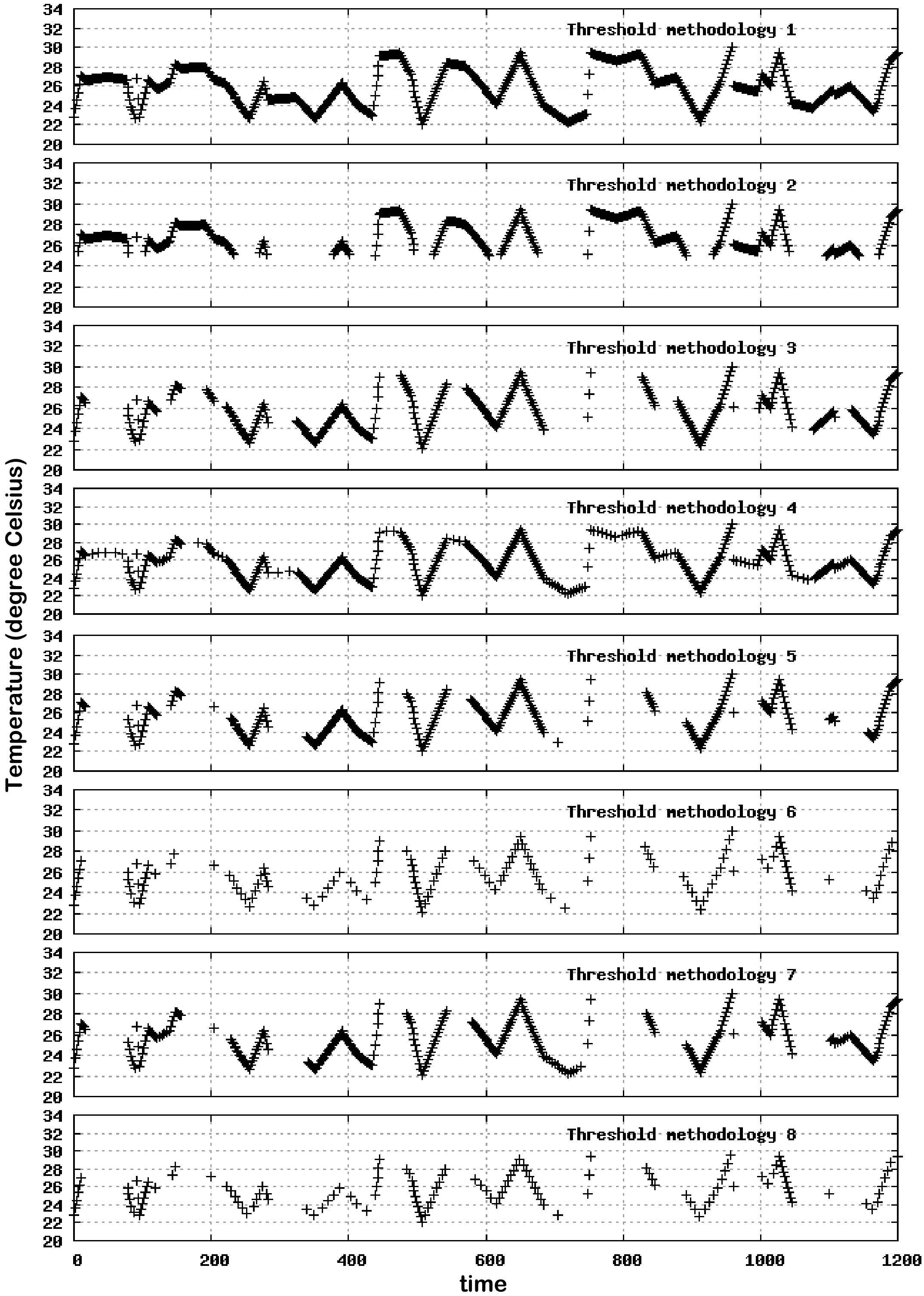

Figure 3.8. Data abstraction with randomly generated data.

Figure 3.9 shows the total number of transmitted packets with the assumption that one packet is required for each sensor reading. Threshold methodology 6 and

Threshold methodology 8 are able to reduce the number of packets transmitted by more than 50%.

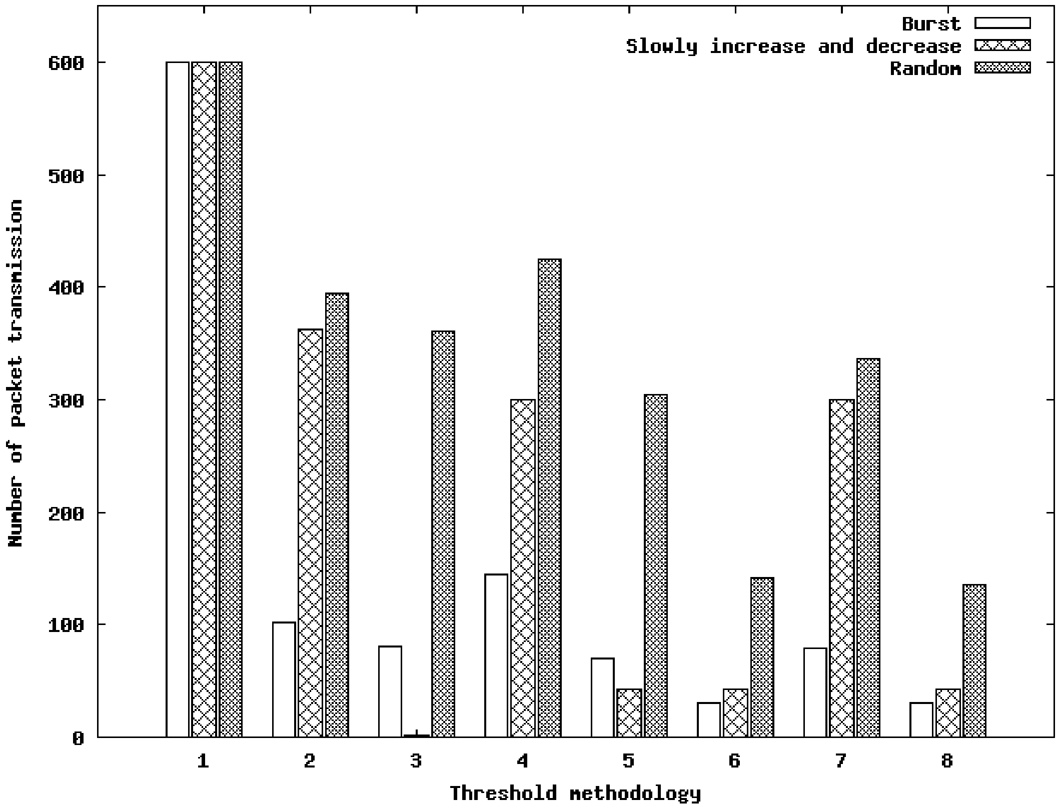

Figure 3.9. Number of packet transported to the base station for different traffic types.

When the abstracted data reaches the sink, the unreported data is re-established using a recovery scheme as in [209]. This scheme merges the unreported data with the previous updated data, thus generating new reported data. Figure 3.10, Figure 3.11, and Figure 3.12 show complete data after the application of the recovery scheme for burst, slowly incremented and decremented, and randomly generated data respectively.

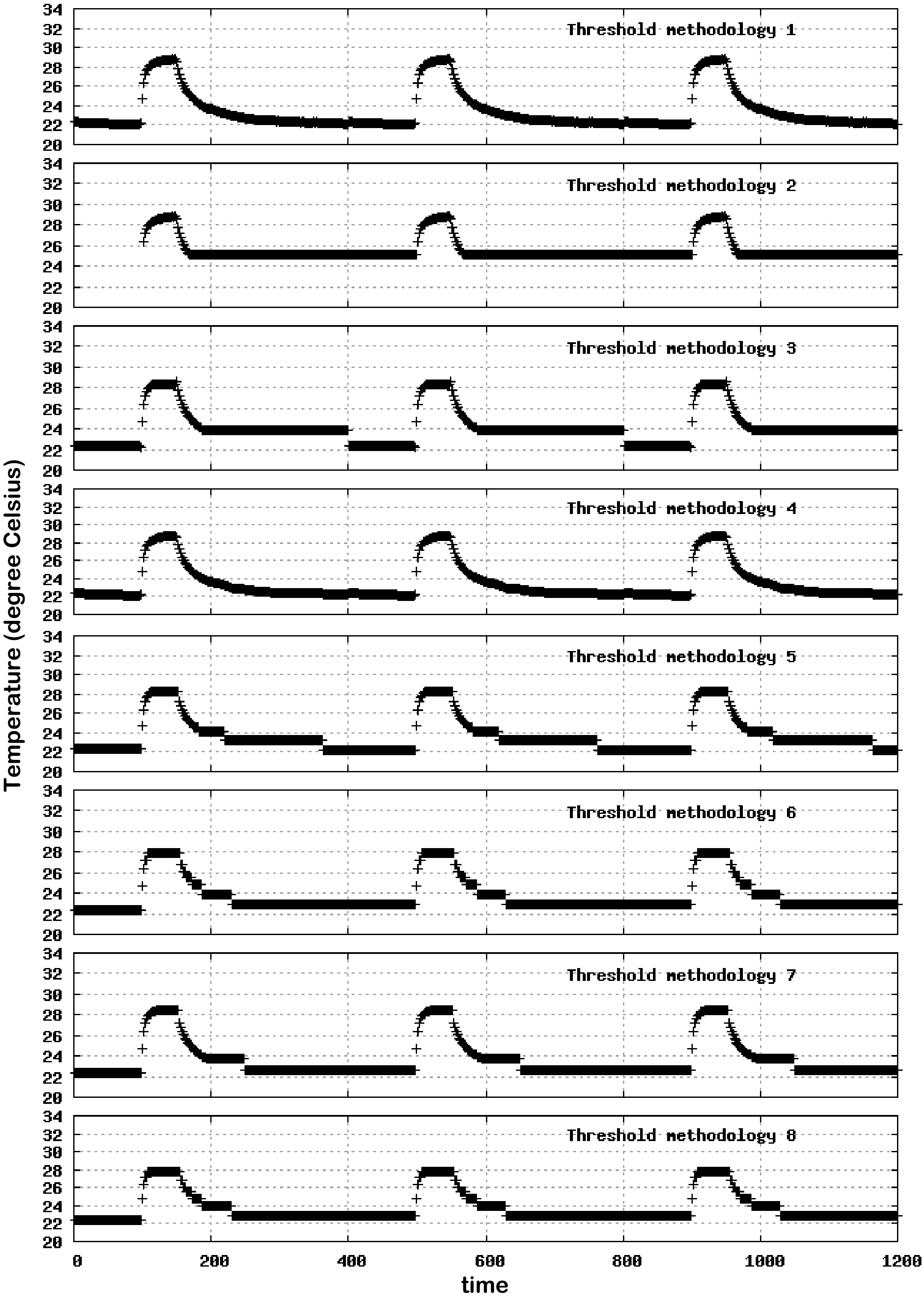

Figure 3.10. Complete data after the application of the data recovery scheme for burst data.

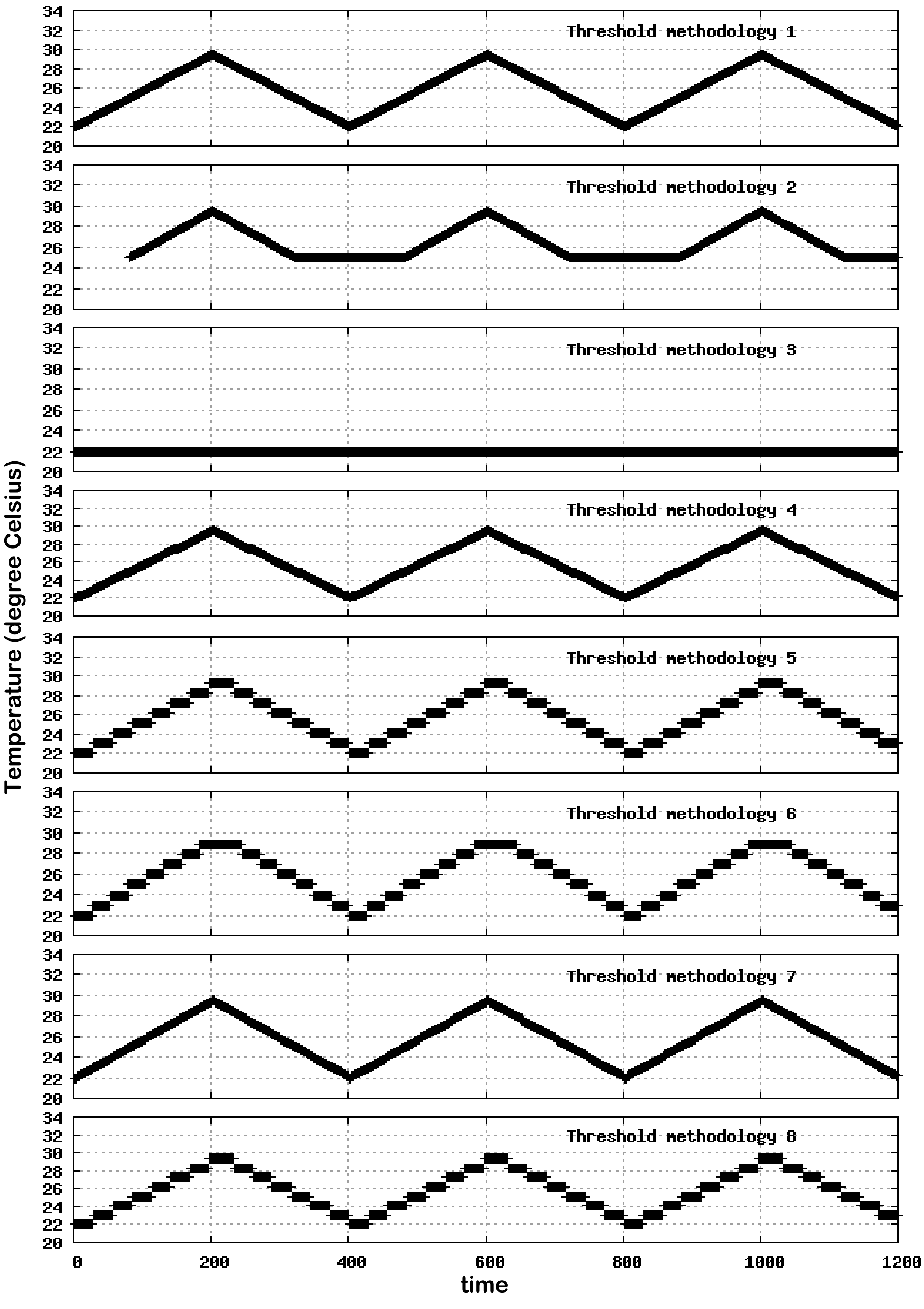

Figure 3.11. Complete data after application of the the data recovery scheme for slowly incremented and decremented data.

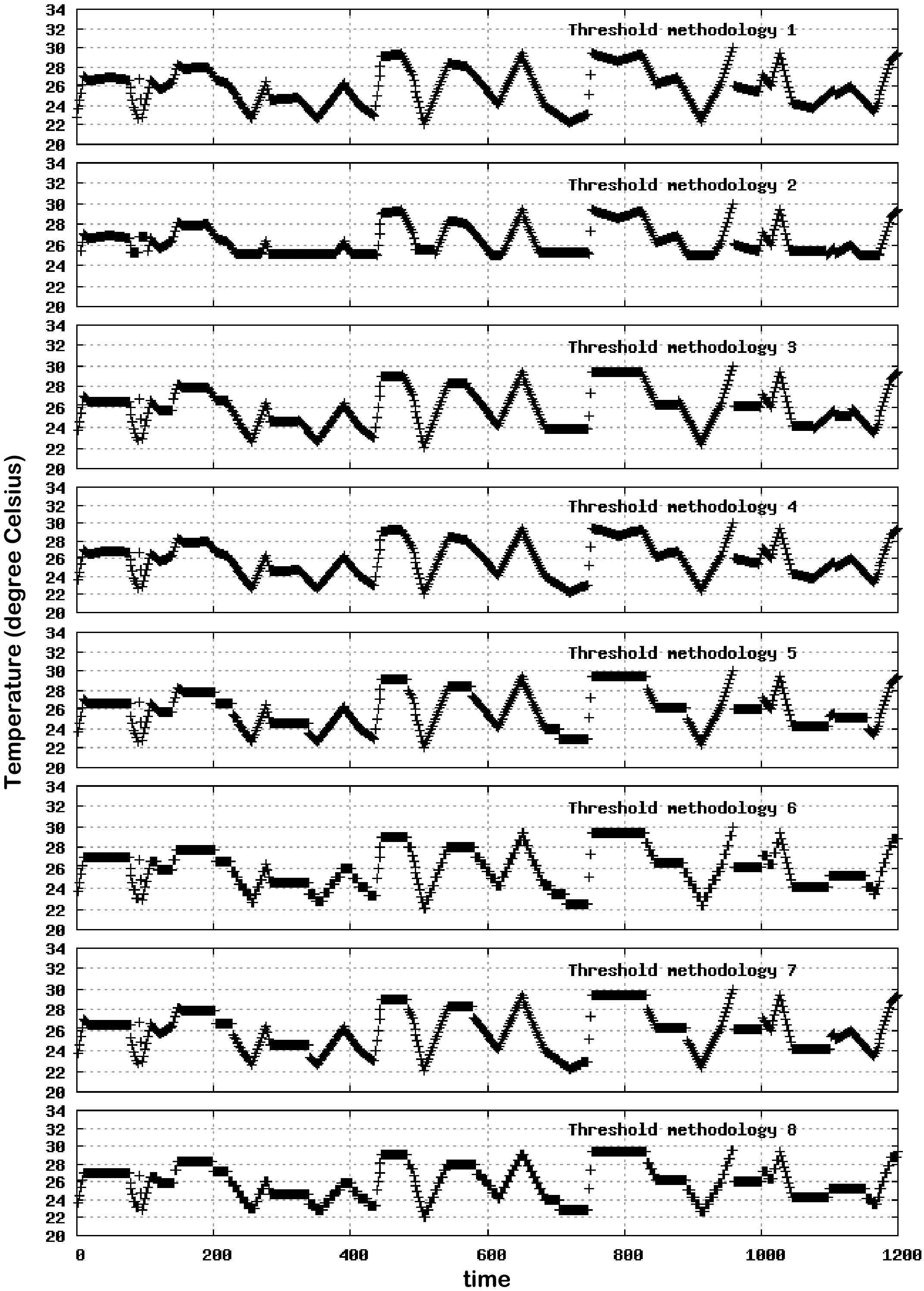

Figure 3.12. Complete data after the application of the data recovery scheme for randomly generated data.

In Figure 3.13, after the data is reformed, the means of the temperature measurements following Threshold methodologies 4 to 8 are less than 0.5. For these five Threshold methodologies, the average values after the recovery scheme are less significant as compared to the means obtained from full sensor readings.

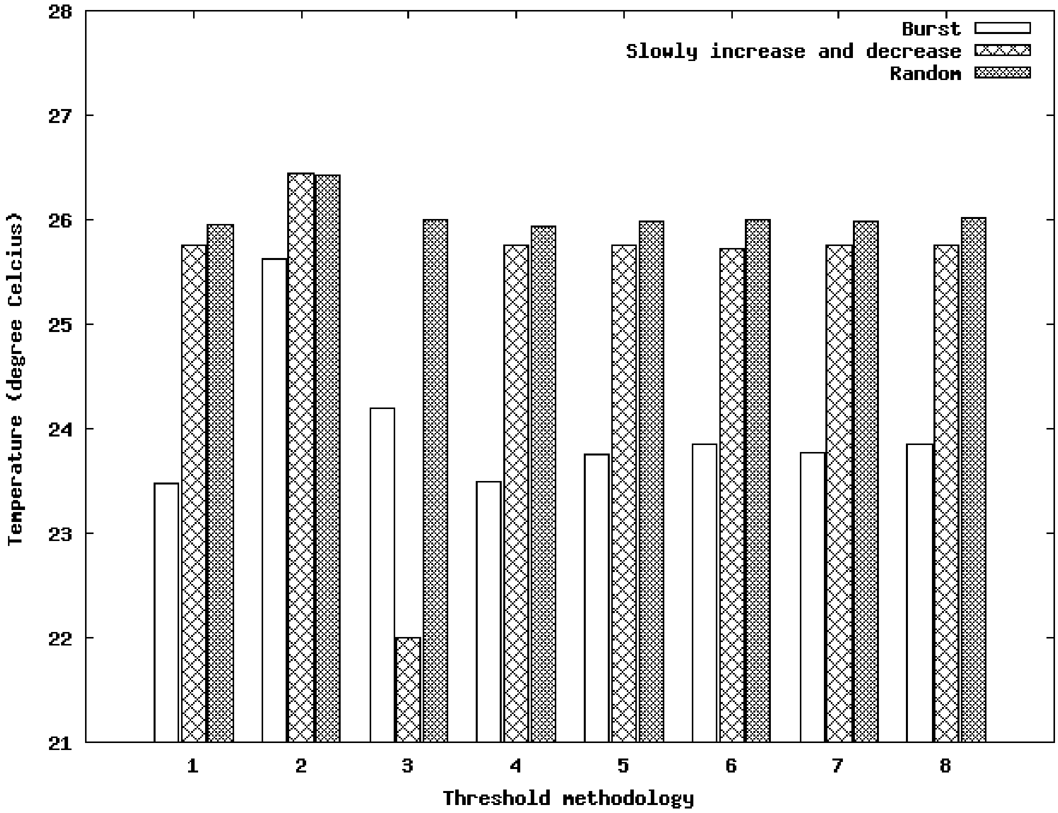

Figure 3.13. Mean of the temperature after the data recovery scheme.

In Figure 3.14, the sum of differences for measurements after data recovery is presented. Equation 3.3 is used to evaluate the sum of differences where d is the recovered data and d_0 is the data obtained from full sensor readings;

$$sum\_d = \sum \left| d - d_0 \right| \tag{3.3}$$

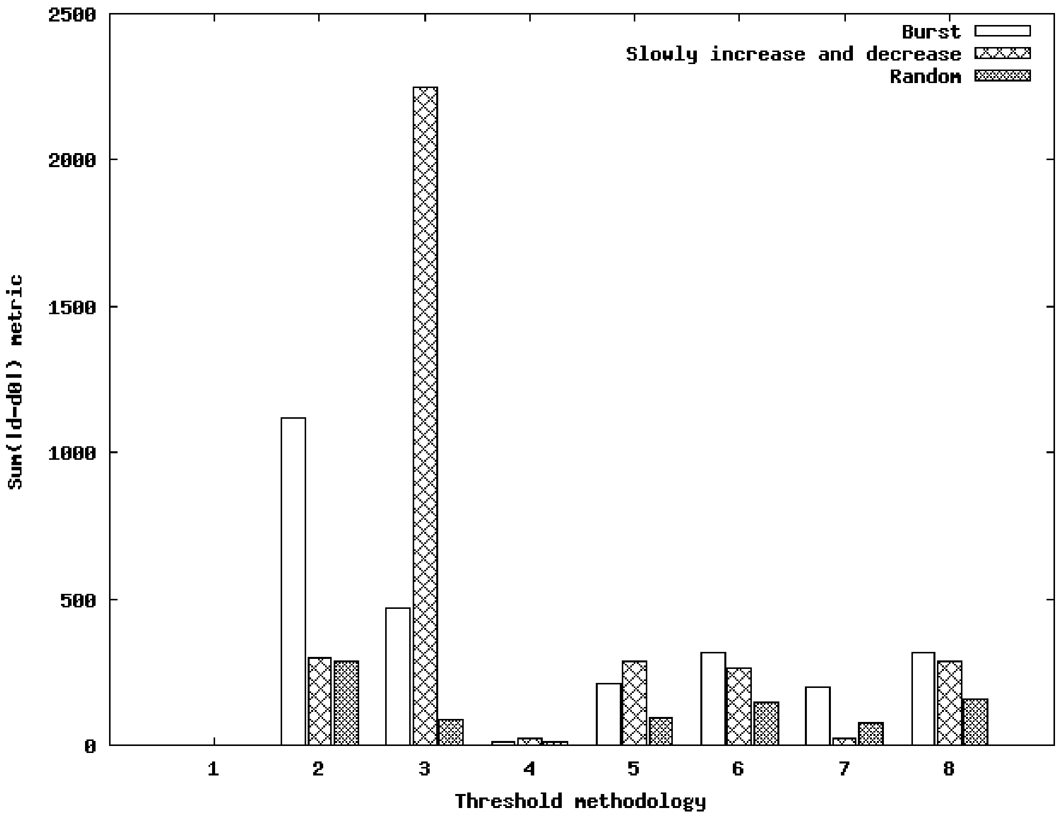

Figure 3.14. The sum of differences after the data recovery scheme.

For Threshold methodologies 4 to 8, the sums of differences are on average less than 320 for each methodology. This implies that each reading has a maximum error of 0.267 degree Celsius, acceptable since according to the datasheet [215], the error of the sensor reading is 0.2 degree Celsius.

Abstract Reporting and Reformation schemes [209] can be further enhanced into Zeroth-, First- and Second-order Data Abstraction and Reformation algorithms [216] to manage linear and non-linear data patterns.

3.4.3 Power Supply Module

It is important to always monitor the condition of the battery in WSNs. Usually the operating voltage ranges from 2.7V to 3.3V [217], and when the voltage drops below 2.7V, the device moves into the 'critical battery condition' (Equation 3.4). In this case, any further operation moves into a limited power resource scheme such as discussed in Section 3.4.2 using Threshold methodology 6 and 8.

$$BattCond = \begin{cases} normal, & f(v) \geq 2.7V \\ critical, & f(v) < 2.7V \end{cases} \tag{3.4}$$

A model can be established to estimate the power consumption and hence network lifetime [218]. The model can define an estimated current consumption per related activity as shown in Table 3.2, allowing the WSN to self-monitor its duty cycle over the operational timeline.

Table 3.2. Current consumption for a number of key functions.

Activity	Current consumption (mA) [217]
Processor in sleep	0.008
Processor in operation	8
Radio in sleep	0.002
Radio in transmit	12
Radio in receive	8
Logger memory in sleep	0.002
Logger memory in write	15
Logger memory in read	4
Sensor in sleep	0.005
Sensor in operation	5

The estimated power consumption can be calculated through Equation 3.5, where a is the current consumption of the related activity, c is the duty cycle on the activity, and i is the activity number;

$$Consumption = \sum_{i}^{n} a_i c_i \tag{3.5}$$

3.5 CogWSN Operation

A CogWSN is expected to self-organise, implying an automatic multi-hop network configuration capability. Link establishment starts at a base station with a beacon broadcast with an interval, for example every 5sec. A node that receives the beacon requests to join as a 'child' node. Upon the successful establishment of the link between parent and child, the node continues to broadcast the same beacon. This process is repeated until the entire network is formed. It should be noted that upon reception of each beacon, child nodes may reselect its parent based on certain metrics such as lesser number of hops or better link quality.

A time slot cycle is defined as shown in Figure 3.15, showing nodes operating on a TDMA basis; which assumes synchronized clocks amongst nodes so that all are awake at the same time in order to receive the broadcast. The time slot is divided into 'Wake Early and Receive Parent's Beacon', 150ms (WB), 'Broadcast Beacon', 30ms (B), 'Allow a Child to Join', 150ms (J), 'Slot Communication to Parent', to be determined (P), 'Slot Communication to Children', to be determined (C), and 'Sleep and Perform Decision Process', to be determined (SC). For each data packet transmitted by a child node, an acknowledgement packet from the parent node is expected in return.

A flow diagram of the operation of the CogWSN is shown in Figure 3.16.

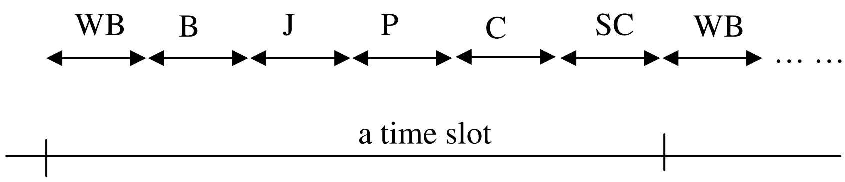

Figure 3.15. Time cycle for CogWSN operation.

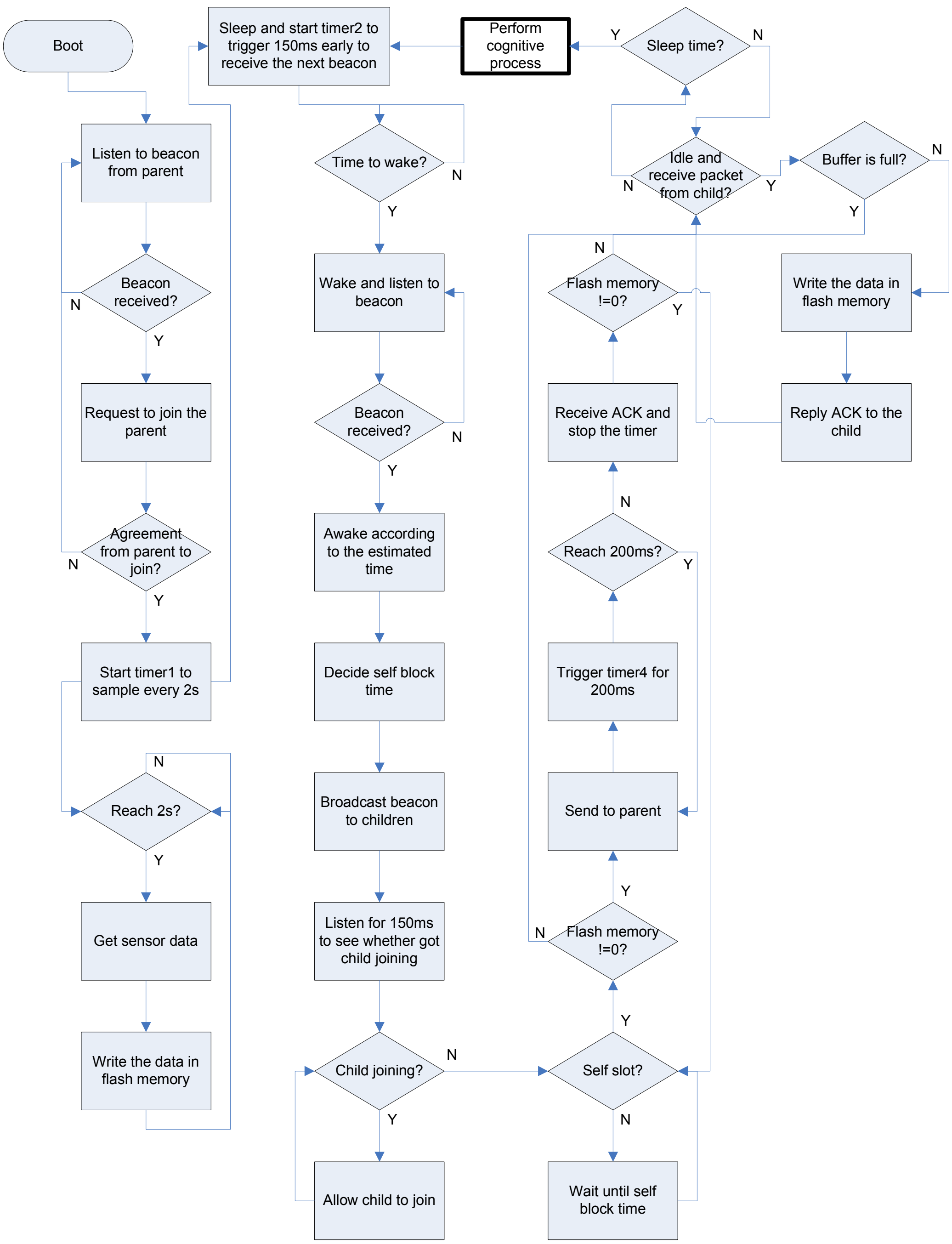

Figure 3.16. Flow diagram for CogWSN operation.

The proposed CogWSN operation is fully implemented into the MICAz platform. The operation allows multi-hop networks to be formed rooted at a base station. Several learning modules as discussed in Chapters 4 to 6 can be adopted to perform the cognitive process.

3.6 Conclusions

CogWSN has been defined as a networked group of sensor nodes capable of sustaining performance in dynamic environments using a cognitive capability, while meeting user requirements. The particular approach adopted centres on a decision process founded in the Problem Solving cognitive process from LRMB in combination with the algorithm reported by Polya. The approach is designed drawing from principles in Cognitive Radio and Cognitive Networks (Section 2.4), which also rely on cognitive decision processes.

The CogWSN decision process is formulated through 4 phases: Observe, Plan, Implement, and Evaluate. A CogWSN architecture has been proposed based on the creation of transceiver, transducer, and power supply virtual modules executing on the cognitive processes through software implementations. Each module contains two elements defined as the State Information (SI), which stores information on the device operating condition, and the Tuneable Function (TF), which defines the actuating function, as mentioned in Section 3.4.

It is essential for all virtual modules to participate in realising energy-efficient operation. In Section 3.4.1, the Transceiver Module has the capability that the transmitter monitors the signal strength received at the receiver and adjusts its power transmission so that packets can be sent within an optimal RSSI range. In Section 3.4.2, the Transducer Module has a similar goal where it reduces the number of transmissions through a controllable abstraction method and reconstitutes all data through a reformation scheme. The Power Supply Module in Section 3.4.3 records its battery operation status by estimating its power consumption through an established model. When the battery falls in the 'critical battery condition', it alerts other modules to move any operation into the minimum operation mode.

Chapter 4: Rule-based CogWSN

Problem Solving as defined in LRMB is searching for a solution to a given problem or finding a path to reach a given goal. The representation of the problem is central and must include a description of the given situation, pre-defined operations for changing the situation, and assessment criteria to determine whether the goal has been achieved or not [219]. Rules are used to represent transitions between states of the problem. The Chapter starts with an overview of the rule-based approach and rule-based learning and illustrates how these can be used within a CogWSN decision process. The detail of each phase of the decision process is discussed.

4.1 Rule-Based Learning

Rule-based learning has roots in cognitive psychology [219, 220, 221] and early computer models of learning implemented through a high level computer language with computational statements such as `if: then` production rules [219]. The design of rule-based approaches is based on a strict and static predefined set of policies hard-coded, generating responses accordingly. Figure 4.1 shows an example of a traditional rule-based approach.

```
if (state S1) then (action A1);
elseif (state S2) then (action A2);
elseif (state S3) then (action A3);
... ...
else (state Sn) then (action An);
end if;
```

Figure 4.1. An example of a traditional Rule-Based approach.

Rule-based approaches can be enhanced to exhibit a learning capability for example, by adding some information that allows the best rule to be selected for the action based on that information. A decision made from the rule-based learning is based on properties alone and relies on simple criteria that do not require a significant amount of memory. Figure 4.2 shows an example of the application of rule-based learning using greedy scoring [222].

```
if (state S1) then (action A1 with the highest score
scr1 and then the greatest tuning);
elseif (state S2) then (action A2 with the highest
score scr2 and then the greatest tuning);
elseif (state S3) then (action A3 with the highest
score scr3 and then the greatest tuning);
... ...
else (state Sn) then (action An with the highest
score scrn and then the greatest tuning);
end if;
if (action is correct) then (increase the score by
1);
else (action is incorrect) then (decrease the score
by 1);
end if;
```

Figure 4.2. An example of Rule-based learning using greedy scoring.

In this work, the rule-based approach and rule-based learning using greedy scoring are considered in the implementation of a CogWSN. Neither approach consumes significant computation and memory resources; however, the limitation is that all correct inputs and outputs must be matched before the deployment. Any input not matched properly will not produce the correct output.

4.2 Observe Phase

This phase requires continuous observation to identify any potential issue in the monitored conditions. The status of the monitored conditions is captured to determine whether there is an issue; if any issue is identified, the next phase, the Plan Phase, is triggered.

Normally, a summary is compiled for each attribute as in Figure 4.3, where *u* is the type of condition and *v* is the value assigned to the status. An example of a summary is shown in Figure 4.4 for the observation of RSSI, slot utilisation to parent node, and slot utilisation from child nodes.

```
Condition(u₁): status(v₁)
Condition(u₂): status(v₂)
… …
Condition(uₙ): status(vₙ)
```

Figure 4.3. A summary created from observation.

```
Condition(RSSI): status(-62dBm)
Condition(Slot utilisation to parent node): status(6
packets)
Condition(Slot utilisation from children nodes):
status(9 packets)
```

Figure 4.4. An example of a summary from observation.

To determine whether to trigger the Plan Phase, a mapping between the observed conditions and the pre-defined goals is executed as shown in Table 4.1. If any pre-defined goal is not achieved, the Plan Phase is triggered.

Table 4.1. A mapping between observed conditions and pre-defined goals.

Observed Conditions	**Pre-Defined Goals**
Condition(RSSI): status(-62dBm)	Target 1: Received power for data packet between -85dBm to -75dBm.
Condition(Slot utilisation to parent node): status(6 packets)	Target 2: Slot utilisation to parent node, x is $\frac{1}{2}s_1 \leq x \leq s_1$, where slot allocation to communicate with parent node, s_1 = 2, 4, 8, 16, 32, or 64.
Condition(Slot utilisation from children nodes): status(9 packets)	Target 3: Slot utilisation from child node, y is $\frac{1}{2}s_2 \leq y \leq s_2$, where slot allocation to communicate with child node, s_2 = 2, 4, 8, 16, 32, or 64.

Observation is triggered under two scenarios; event-based and timer-based [14, 15, 19]. For event-based observation, when the monitored condition experiences critical changes, a report is compiled and the Plan Phase is triggered. For timer-based observation, conditions are monitored at a fixed interval. If any condition requires further attention, the Plan Phase is triggered. Based on CogWSN operation as implemented in Section 3.5, the event-based observation is most suitable as the Observe Phase; when the wake cycle ends, the communication unit is turned off and the resource can be used to perform the cognitive decision process. Timer-based observation may trigger the cognitive decision process when all components in the WSN are busy with their own tasks.

4.3 Plan Phase

This phase is invoked on any condition requiring further action to resolve outstanding issues from the Observe Phase. There are two ways to derive a plan, based on the first detected symptom in order or based on all detected symptoms.

For example, based on Table 4.1, a scenario as shown in Figure 4.5 is detected. For a plan to be derived based on the first detected symptom in order, only condition

1 is considered; for a plan based on all detected symptoms, all conditions are taken into consideration.

```
Condition 1: above the expected range
Condition 2: below the expected range
Condition 3: in the expected range
```

Figure 4.5. An example of a summary of detected symptoms.

For rule-based approaches, the solution is pre-determined according the conditions based on the `if: then` statement. For this approach, a plan is derived based on the first detected symptom in order, where a high priority symptom such as radio link connectivity is arranged at the beginning of the order. If a plan is derived for all detected symptoms, all recommended solutions for the states require to be crafted manually through a significant amount of the multiplication of monitored conditions, dramatically increasing the size of the ROM and the RAM in order to accommodate all possible solutions in the degree of p^n, where n is the number of condition and p is the possible status for a particular condition.

The same recommendation is appropriate for rule-based learning using greedy scoring. This solution is determined by selecting the highest score in the action list. If there is more than one solution with the same highest score, the action with the greatest tuning is selected as in Figure 4.2, allowing the node to solve the problem more rapidly. However it must be noted that this strategy may cause the function to be over-tuned in some cases [223]. A look-up table can be added to act as a filter in order to prevent over-tuning, discussed in Section 5.5.

4.4 Implement Phase

In this phase, an action is performed according to the derived solution. The solution could be in two forms viz. 'level-up' or 'level-down' a setting [12] or 'tune a setting' with a parameter [19]. The advantage of the 'level-up' or 'level down' a setting is the action is adjusted step by step; however this approach suffers in that the adjustment is slow and may require several cognitive cycles to achieve the goal. The advantage of 'tuning a setting' with a parameter is the rapidity of adjustment but the selection of the correct parameter is difficult without *a priori* knowledge, which has to be embedded into the system at the Plan Phase. For the Rule-based CogWSN, the option of 'level up' or 'level down' a setting is chosen for the implementation, and for the Rule-based CogWSN with Greedy Scoring, 'tune a setting' with a parameter is preferred since the setting can be tagged with the score information.

4.5 Evaluation Phase

The Evaluation Phase creates feedback on the action taken to check the accuracy and validity of the derived solution. The evaluation result is stored so that it can be used as 'experience' for future Plan Phases. The Rule-based approach is crafted as an ideal solution and as such feedback is not needed and no learning is involved [224]. However, feedback is required for rule-based learning as shown in Figure 4.6 as each rule is assigned a score.

```
if (action is correct) then (increase the score by
1);
else (action is incorrect) then (decrease the score
by 1);
end if;
```

Figure 4.6. Feedback in Rule-based CogWSN with Greedy Scoring.

4.6 Verification

Several case studies are carried out to verify the performance of Rule-based CogWSN and Rule-based CogWSN with Greedy Scoring. The objective of the case studies is to drive the CogWSN to achieve optimum performance in terms of point-to-point connectivity and slot utilisation. The case studies focus on three domains; power transmission, slot allocation to the parent node, and slot allocation to the child node. Efficient use of power for transmission and slot allocation conserves energy and prolongs network lifetime. For Rule-based CogWSN, the targets are framed according to the order, monitored conditions, and derived actions for each domain, as shown in Table 4.2. For Rule-based CogWSN with Greedy Scoring, the targets are framed according to the monitored conditions, and derived actions for each domain are shown in Table 4.3. By default, the score for each rule for Rule-based CogWSN with Greedy Scoring is set to 127, where the minimum score is 0 and the maximum score is 255.

Table 4.2. The targets to achieve, monitored conditions, and derived actions for Rule-based CogWSN.

Test Metrics	**Targets**	**Monitored Conditions**	**Derived Actions**
Power transmission	Received power between -85dBm to -75dBm	-85dBm ≤ RSSI ≤ -75dBm	None
		RSSI < -85dBm	Increase transmit power by 1 level up
		RSSI > -75dBm	Decrease transmit power by 1 level down
Slot utilisation to parent node	Slot utilisation to parent node, x is $\frac{1}{2}s_1 \leq x \leq s_1$, where slot allocation to communicate with parent node, s_1 = 2, 4, 8, 16, 32, or 64	$\frac{1}{2}s_1 \leq x \leq s_1$	None
		$x > s_1$	Increase slot allocation by $2s_1$
		$x < \frac{1}{2}s_1$	Reduce slot allocation by $s_1/2$
Slot utilisation from child node	Slot utilisation from children nodes, y is $\frac{1}{2}s_2 \leq y \leq s_2$, where slot allocation to communicate with child node, s_2 = 2, 4, 8, 16, 32, or 64	$\frac{1}{2}s_2 \leq y \leq s_2$	None
		$y > s_2$	Increase slot allocation by $2s_2$
		$y < \frac{1}{2}s_2$	Reduce slot allocation by $s_2/2$

Table 4.3. The targets to achieve, monitored conditions, and derived actions for Rule-based CogWSN with Greedy Scoring.

Test metrics	**Targets**	**Monitored conditions**	**Derived Actions**
Power transmission	Received power between -85dBm to -75dBm	-85dBm $\leq$ RSSI $\leq$ -75dBm	None
		RSSI < -85dBm	Increase transmit power by 1, 2, 3, or 4 levels up
		RSSI > -75dBm	Decrease transmit power by 1, 2, 3, or 4 levels down
Slot allocation to parent	Slot utilisation to parent node, x is $\frac{1}{2}s_1 \leq x \leq s_1$, where s_1 = 2, 4, 8, 16, 32, or 64	$\frac{1}{2}s_1 \leq x \leq s_1$	None
		$x > s_1$	Increase slot allocation by $2s_1$, $4s_1$, $8s_1$, $16s_1$, or $32s_1$
		$x < \frac{1}{2}s_1$	Reduce slot allocation by $s_1/2$, $s_1/4$, $s_1/8$, $s_1/16$, or $s_1/32$
Slot allocation for child node	Slot utilisation from children nodes, y is $\frac{1}{2}s_2 \leq y \leq s_2$, where s_2 = 2, 4, 8, 16, 32, or 64	$\frac{1}{2}s_2 \leq y \leq s_2$	None
		$y > s_2$	Increase slot allocation by $2s_2$, $4s_2$, $8s_2$, $16s_2$, or $32s_2$
		$y < \frac{1}{2}s_2$	Reduce slot allocation by $s_2/2$, $s_2/4$, $s_2/8$, $s_2/16$, or $s_2/32$

Six experiments were conducted using the MICAz platform [76]. Experiments 1 and 2 are to verify the connectivity performance of CogWSN; Experiments 3 and 4 are to ensure the slot allocation for parent node can be dynamically adjusted according to the requirement; lastly, Experiments 5 and 6 are to determine that the slot allocation for a child node can be adjusted according to the child node's data traffic. A base station is directly connected to a PC through a USB-serial cable and each node is either coded with rules for Rule-based CogWSN or with rules and scores for Rule-based CogWSN with Greedy Scoring. CogWSN operation is implemented as defined in Section 3.5; the series of experiments, setup and the initial parameters are shown in Table 4.4.

Table 4.4. Experimental setup for 1 to 6 with initial parameters for Rule-based CogWSN and Rule-based CogWSN with Greedy Scoring.

Experiments and Set-up	**Initial Parameters**
Experiment 1 Setup: Node is placed close to the base station.	Power transmission, 0dBm; slot allocation for parent node, 2; slot allocation for child, 2; sampling rate, 0.5Hz
Experiment 2 Setup: After the node is placed close to the base station and is stable, the node is moved far from the base station but still within the communication range.	Power transmission, -25dBm; slot allocation for parent node, 2; slot allocation for child, 2; sampling rate, 0.5Hz
Experiment 3 Setup: Node is placed close to the base station.	Power transmission, -25dBm; slot allocation for parent node, 64; slot allocation for child, 2; sampling rate, 0.5Hz
Experiment 4 Setup: Node is placed close to the base station.	Power transmission, -25dBm; slot allocation for parent node, 2; slot allocation for child, 2; sampling rate, 5Hz
Experiment 5 Setup: Node is placed close to the base station. Child node is placed near to the parent node.	Power transmission, -25dBm; slot allocation for parent node, 2; slot allocation for child, 64; sampling rate, 0.5Hz; its' child sampling rate, 0.5Hz
Experiment 6 Setup: Node is placed close to the base station. Child node is placed near to the parent node.	Power transmission, -25dBm; slot allocation for parent node, 2; slot allocation for child, 2; sampling rate, 0.5Hz; its' child sampling rate, 0.5Hz

In Experiment 1, a node is placed initially in close proximity to the base station; the power transmission by default is set to the maximum, 0dBm and sensor data is sampled at 0.5Hz. The slot allocation to communicate with both the parent node and child node are both set to 2. For Rule-based CogWSN, (Figure 4.7) owing to the relatively short distance between node and base station, a received power above -75dBm at the beginning of the experiment results. After 4 cycles, the received power is adjusted step by step and successfully maintained between -85dBm and -75dBm. The adjustment continues for slot allocation to the parent node; the slot allocation is adjusted to 4 (Figure 4.8) clearing the buffer in each wake up cycle. The data in the buffer indicates the number of packets pending in the node to be transmitted to the parent node. Slot allocation is maintained from the 6^{th} cycle onwards, at which time slot utilisation is almost constant.

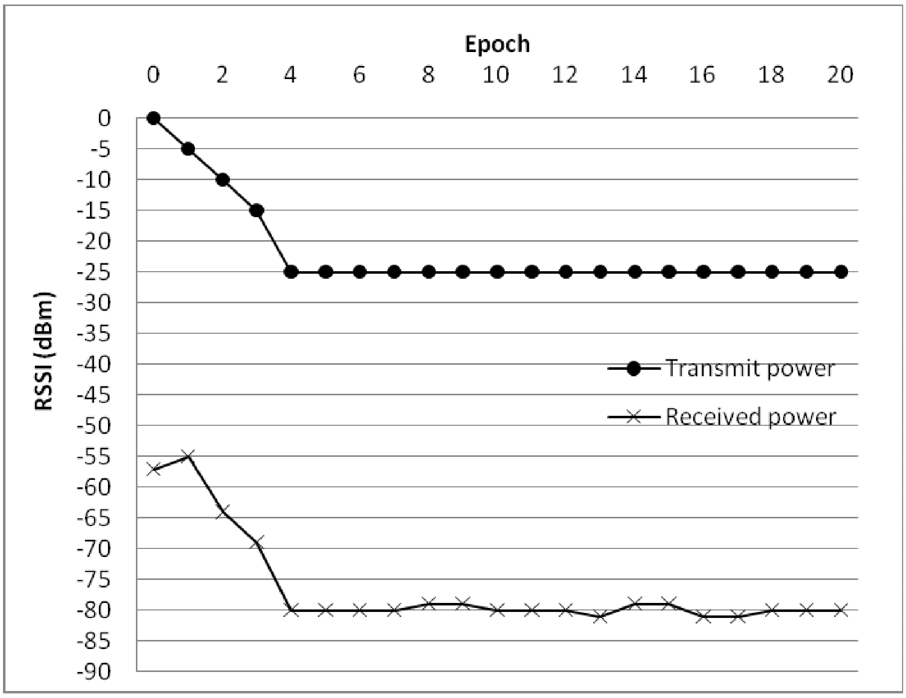

Figure 4.7. RSSI as a function of the number of cycles for Rule-based CogWSN in Experiment 1.

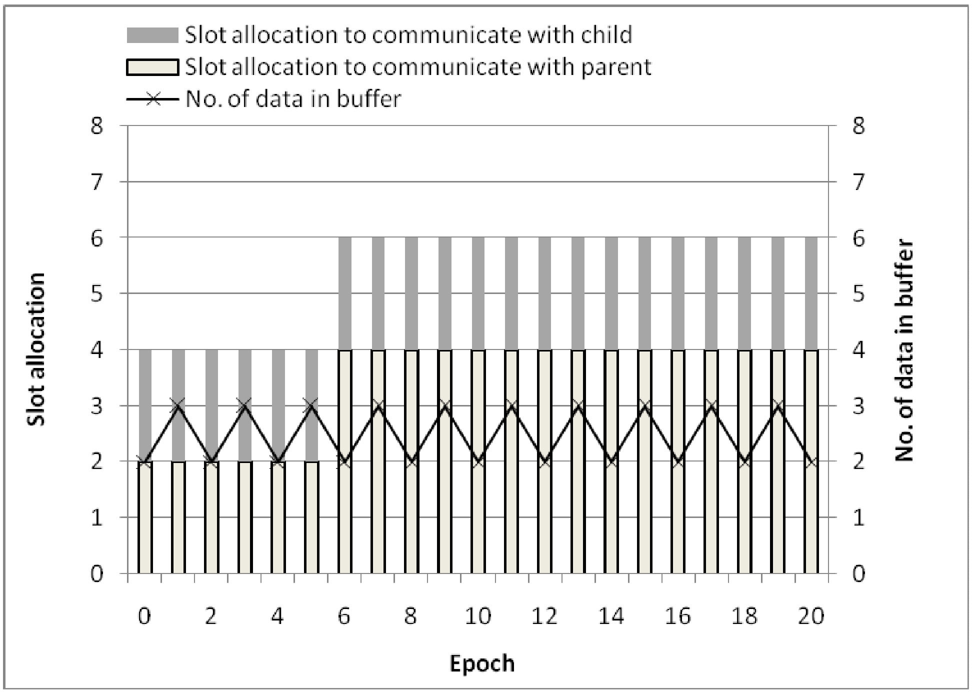

Figure 4.8. Slot allocation and buffer condition as a function of the number of cycles for Rule-based CogWSN in Experiment 1.

For Rule-based CogWSN with Greedy Scoring, at the outset, since the RSSI is above -75dBm, adjustment is needed to reduce the power for transmission. At the first cycle, the action of decreasing transmission power by 1, 2, 3, and 4 levels down are assigned the same score. The most significant adjustment is selected, reducing the transmission power requirement gradually. In this case, only one cycle of adjustment is needed to maintain the transmission power between -85dBm to -75dBm (Figure 4.9). The power adjustment is followed by adjustment of the slot communication with the parent (Figure 4.10). The first increment in slot allocation is performed with the most significant adjustment but since the allocated slot utilisation is low, a decrement is executed in the next cycle; the decrement is implemented with the most significant level of adjustment. The high utilisation causes the adjustment to be repeated. Feedback on the outputs from the first increment alerts the system that the rule was incorrectly selected; therefore the next increment is performed with the next most significant adjustment and so on. At the 6^{th} cycle, the adjustment is tuned and the slot allocation to communicate with the parent node is maintained at 4.

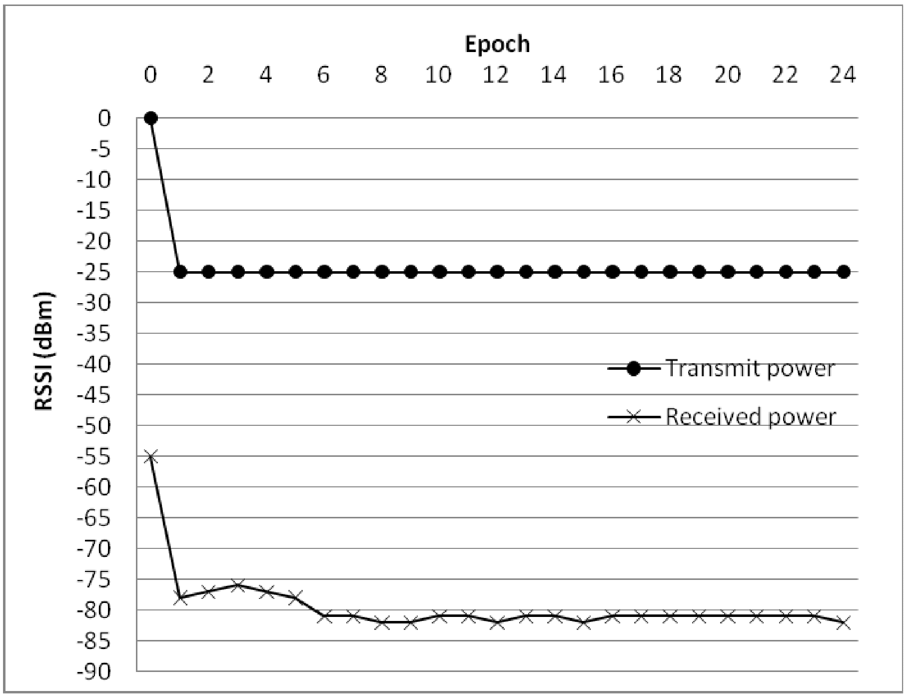

Figure 4.9. RSSI as a function of the number of cycles for Rule-based CogWSN with Greedy Scoring in Experiment 1.

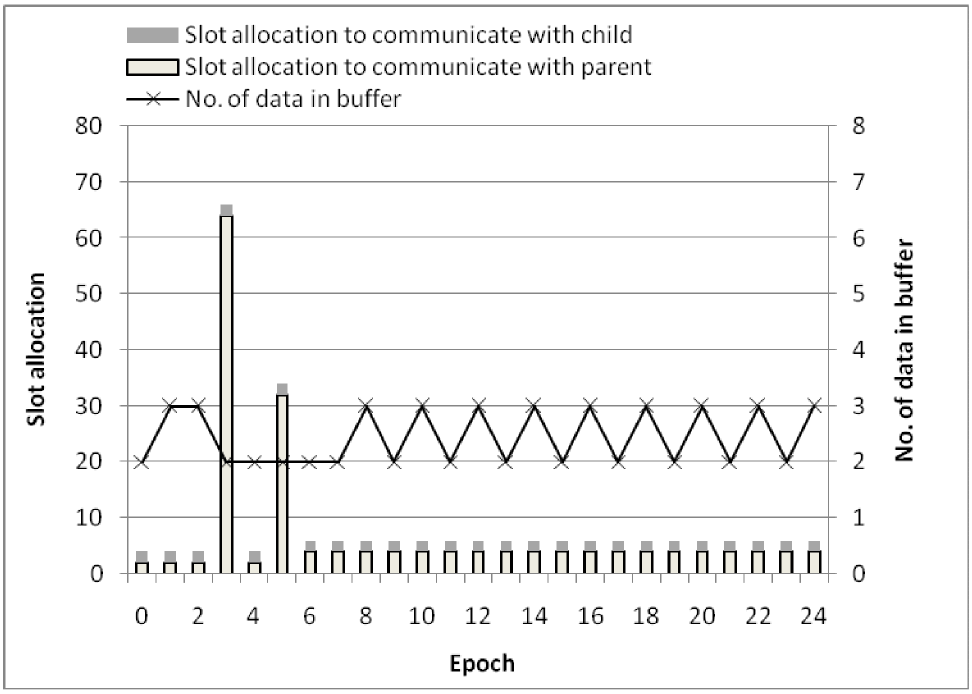

Figure 4.10. Slot allocation and buffer condition as a function of the number of cycles for Rule-based CogWSN with Greedy Scoring in Experiment 1.

In Experiment 2, the transmission power is set to the minimum -25dBm by default and sensor data is sampled at 0.5Hz. Slot allocation to communicate with the parent and child nodes are both set to 2 maintaining the same setting as in Experiment 1. At the outset, a node is placed in close proximity to a base station until the goal is achieved. At the 10^{th} cycle, the node is moved at walking speed away from the base station but is still within communication range. When the node stops receiving an acknowledgement packet, it increases its transmission power towards the maximum. The consequence is that the node receives feedback from the base station on the overuse of the necessary transmission power. For Rule-based CogWSN, after 2 cycles, the received power is reduced and successfully maintained between -85dBm to -75dBm (Figure 4.11). No appreciable adjustment is carried out for slot allocations (Figure 4.12) since connectivity is maintained and thus slot utilisation is almost constant.

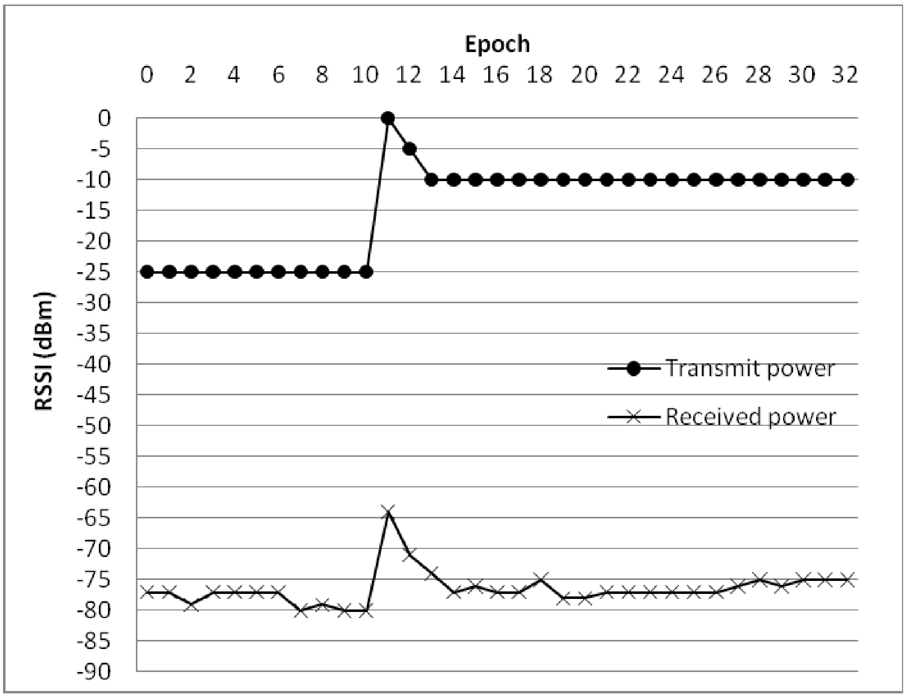

Figure 4.11. RSSI as a function of the number of cycles for Rule-based CogWSN in Experiment 2.

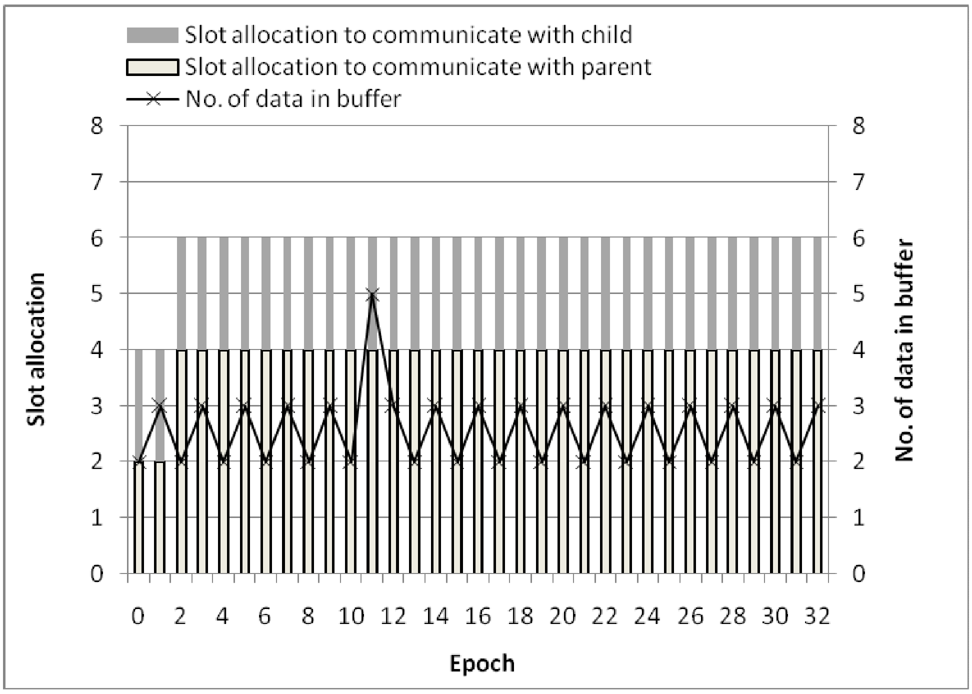

Figure 4.12. Slot allocation and buffer condition as a function of the number of cycles for Rule-based CogWSN in Experiment 2.

For Rule-based CogWSN with Greedy Scoring, at the 21st cycle at which time no more adjustment is performed, the node is moved away from the base station but remains within communication range. The node increases its transmission power towards the maximum. Due to power overuse, a reduction of its transmission power is required. Although the most significant reduction of the transmission power is selected, the goal is not achieved. Thus the adjustment is reversed to increase the transmission power, incremented until a power transmission between -85dBm to -75dBm is successfully maintained after 4 cycles (Figure 4.13). Note that a minor increase of the RSSI in the 32^{nd} cycle, results in the further reduction in transmission power to -15dBm. No significant adjustment is performed for slot allocations (Figure 4.14) as connectivity is maintained and slot utilisation remains almost constant.

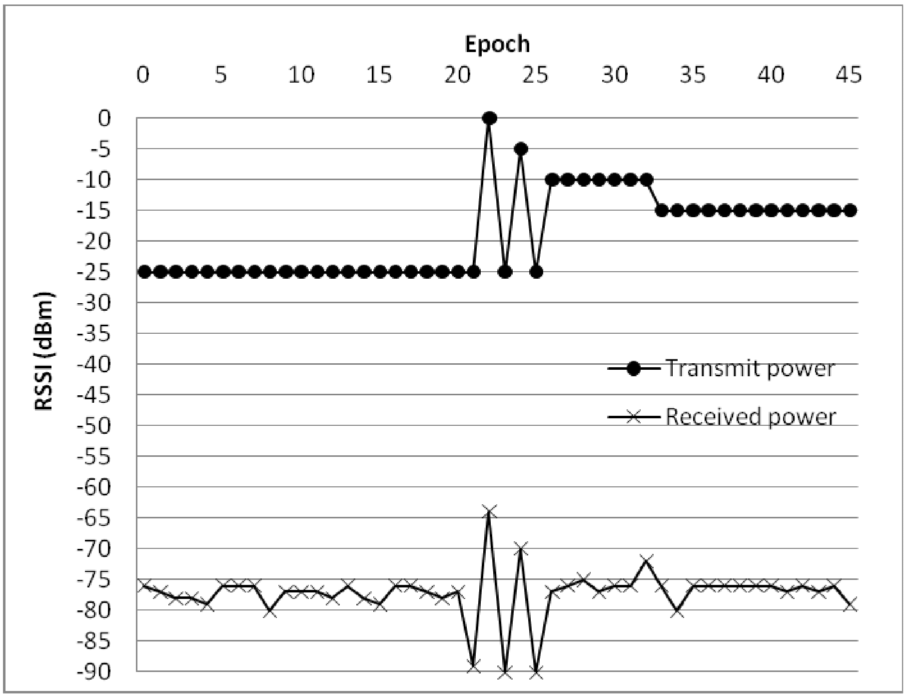

Figure 4.13. RSSI as a function of the number of cycles for Rule-based CogWSN with Greedy Scoring in Experiment 2.

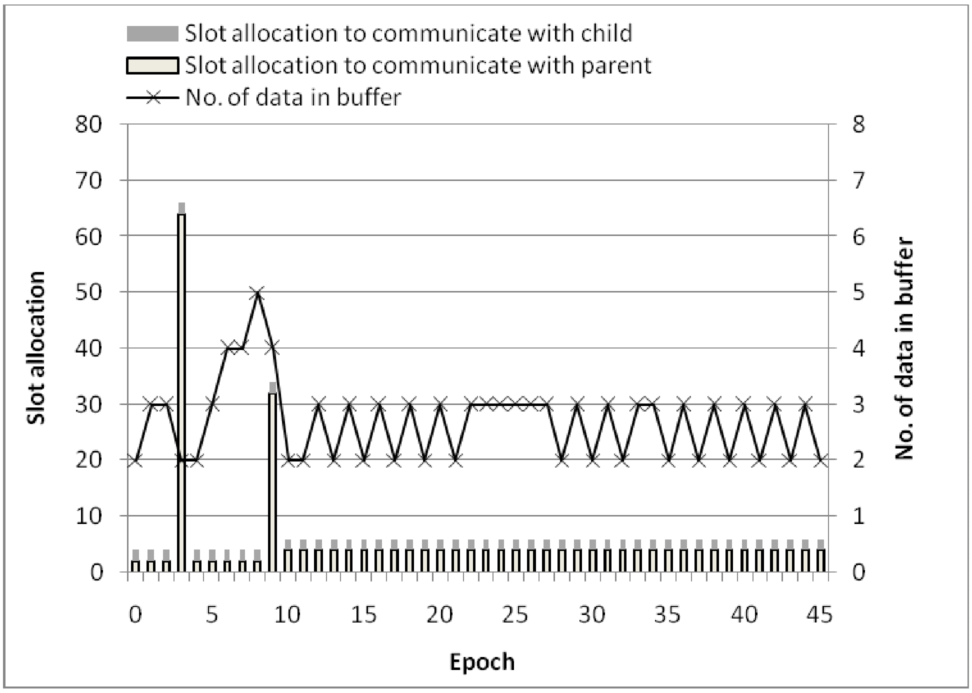

Figure 4.14. Slot allocation and buffer condition as a function of the number of cycles for Rule-based CogWSN with Greedy Scoring in Experiment 2.

In Experiment 3, for Rule-based CogWSN, a node is placed close to a base station; the slot allocation to communicate with both the parent and child nodes is set to 64 and 2, respectively, to emulate a situation where too many slots are allocated to communicate with the parent. The sensor data is sampled at 0.5Hz and the transmission power set to -25dBm. No adjustment is carried out for the transmission power since the RSSI is maintained between -85dBm to -75dBm (Figure 4.15). The allocation of 64 slots to communicate with the parent node is excessive, leaving too many unused allocated slots; therefore, the number of slots must be reduced. After 4 cycles, the slot allocation to communicate with the parent node is reduced to s_1 equals 4, satisfying the slot utilisation criterion of between $\frac{1}{2}s_1$ to s_1 as shown in Figure 4.16.

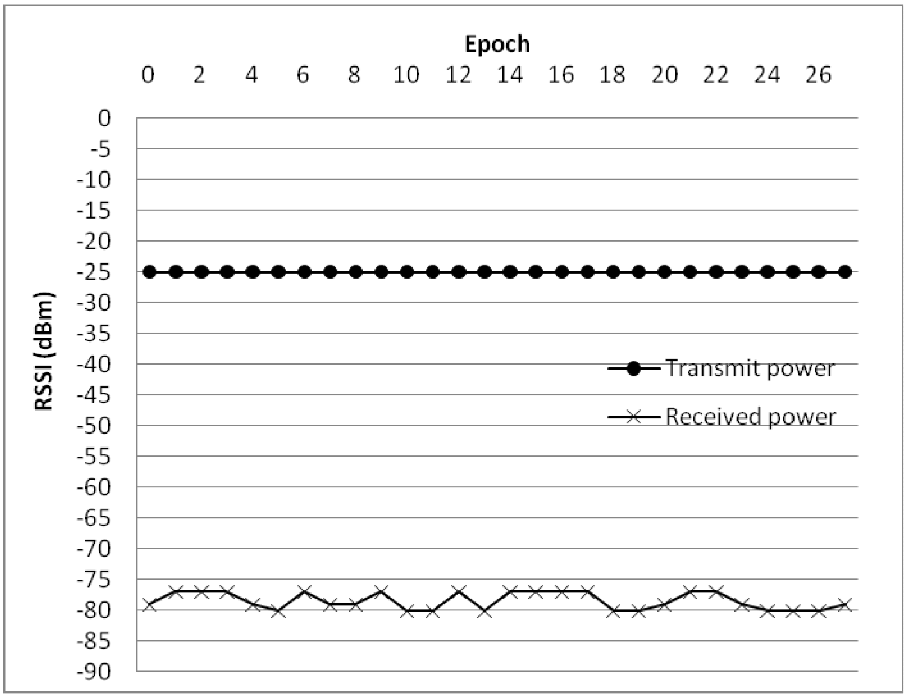

Figure 4.15. RSSI as a function of the number of cycles for Rule-based CogWSN in Experiment 3.

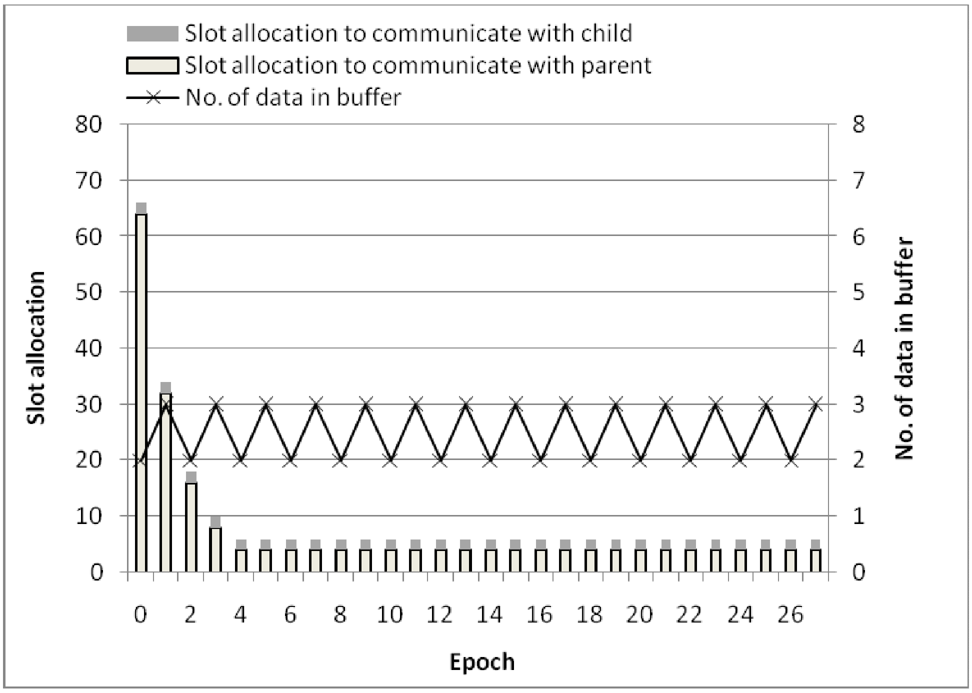

Figure 4.16. Slot allocation and buffer condition as a function of the number of cycles for Rule-based CogWSN in Experiment 3.

For Rule-based CogWSN with Greedy Scoring, no adjustment is carried out for the transmission power since the RSSI remains between -85dBm to -75dBm (Figure 4.17). Once again, the allocation of 64 slots to communicate with the parent node is excessive, leaving too many unused allocated slots; therefore, the number of slots must be reduced. After 4 cycles, the slot allocation to communicate with the parent node reduces to s_1 equals 4, satisfying the slot utilisation criterion of between $\frac{1}{2}s_1$ to s_1 as shown in Figure 4.18. The adjustment is similar to the scenario in Figure 4.10. Rule-based CogWSN with Greedy Scoring allows tuning directly from the maximum to the optimum slot allocation.

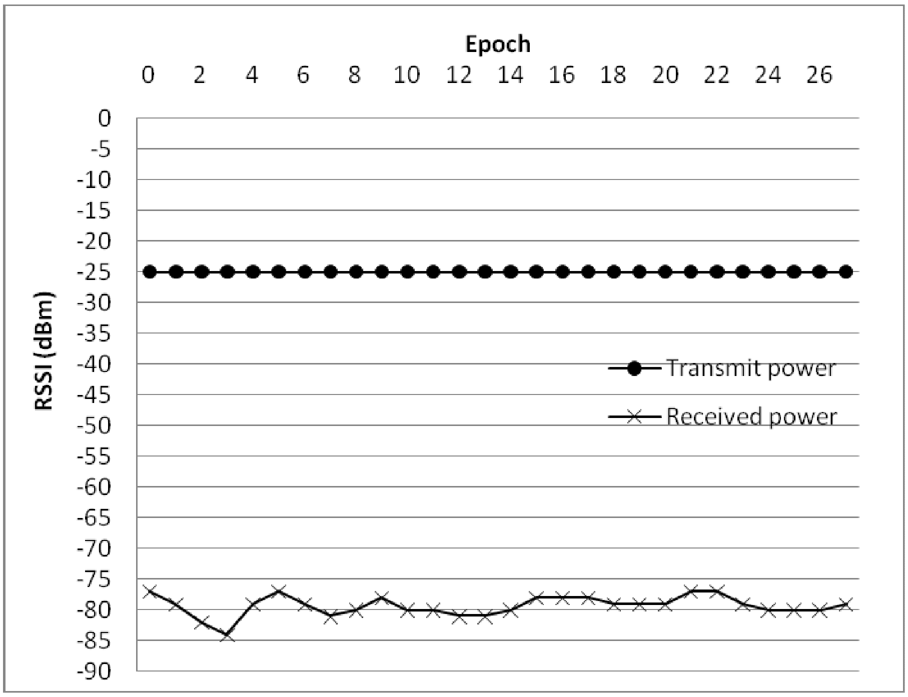

Figure 4.17. RSSI as a function of the number of cycles for Rule-based CogWSN with Greedy Scoring in Experiment 3.

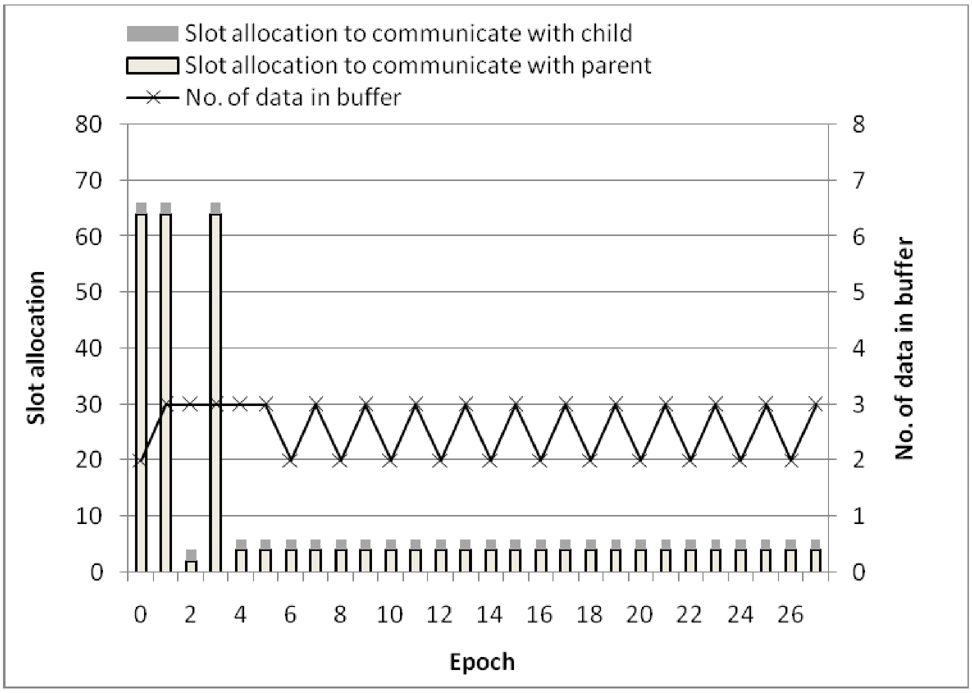

Figure 4.18. Slot allocation and buffer condition as a function of the number of cycles for Rule-based CogWSN with Greedy Scoring in Experiment 3.

In Experiment 4, a node is placed close to a base station, the slot allocation to communicate with the parent node by default is set to 2, the slot allocation to communicate with the child node is set to 2 also, and the sensor data is sampled at 0.5Hz. The transmission power is set to -25dBm. For Rule-based CogWSN, no adjustment is required for the transmission power since the RSSI remains between -85dBm to -75dBm (Figure 4.19). As there are too many packets to be transmitted to the parent node and too few slots allocated, the slot allocation has to be increased. After 9 cycles, the slot allocation to communicate with the parent node is successfully increased to s_1 equals 32, satisfying the slot utilisation criterion of between $\frac{1}{2}s_1$ to s_1, as shown in Figure 4.20.

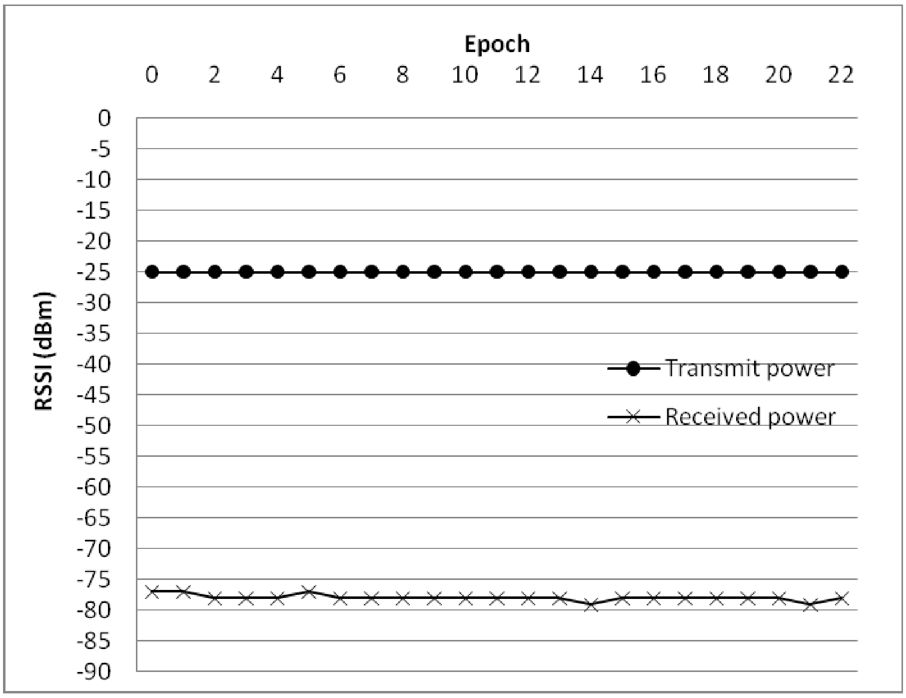

Figure 4.19. RSSI as a function of the number of cycles for Rule-based CogWSN in Experiment 4.

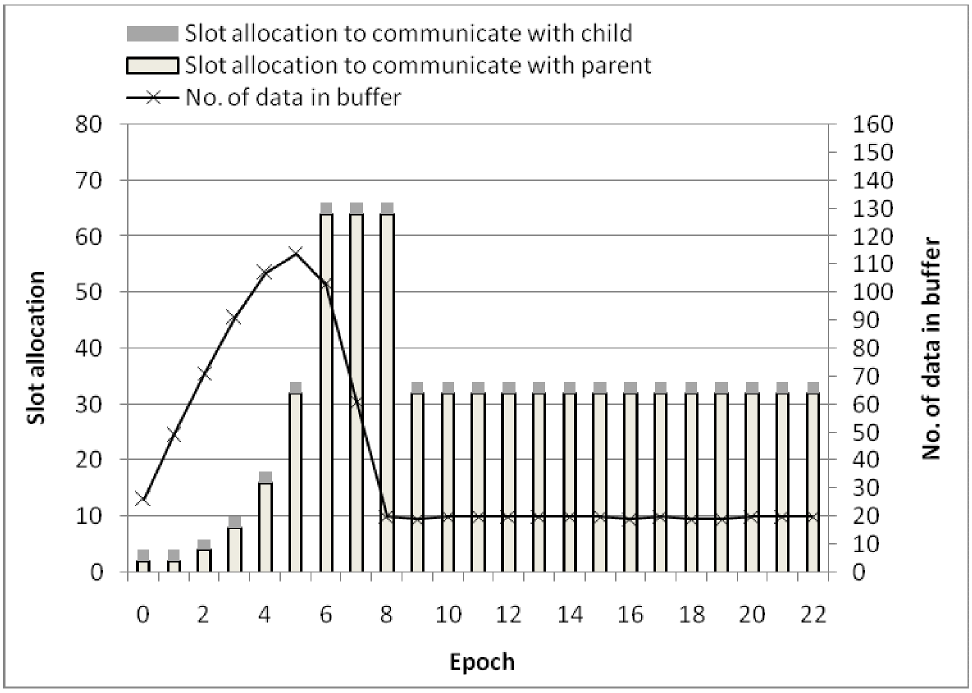

Figure 4.20. Slot allocation and buffer condition as a function of the number of cycles for Rule-based CogWSN in Experiment 4.

For Rule-based CogWSN with Greedy Scoring, there is also no adjustment being carried out for the transmission power since the RSSI remains between -85dBm to -75dBm (Figure 4.21). The slot allocation required to successfully download the data to the parent node is achieved in 8 cycles (Figure 4.22). As a consequence of the buffer filling quickly, adjustment is not able to maintain the goal, and the solution is not able to converge after the goal is initially reached. As a consequence of the buffer being highly utilised and the insufficient slot allocation to communicate with the parent, packets suffer long delays. Conversely when there are fewer packets in the buffer and too generous a slot allocation to communicate with the parent, unnecessary power consumption results owing to idle communication.

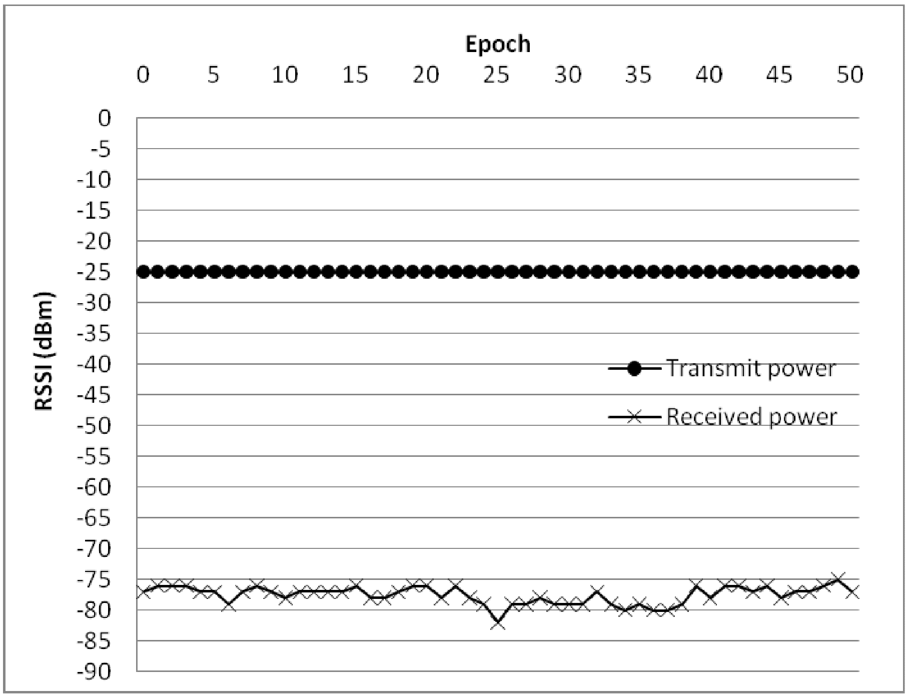

Figure 4.21. RSSI as a function of the number of cycles for Rule-based CogWSN with Greedy Scoring in Experiment 4.

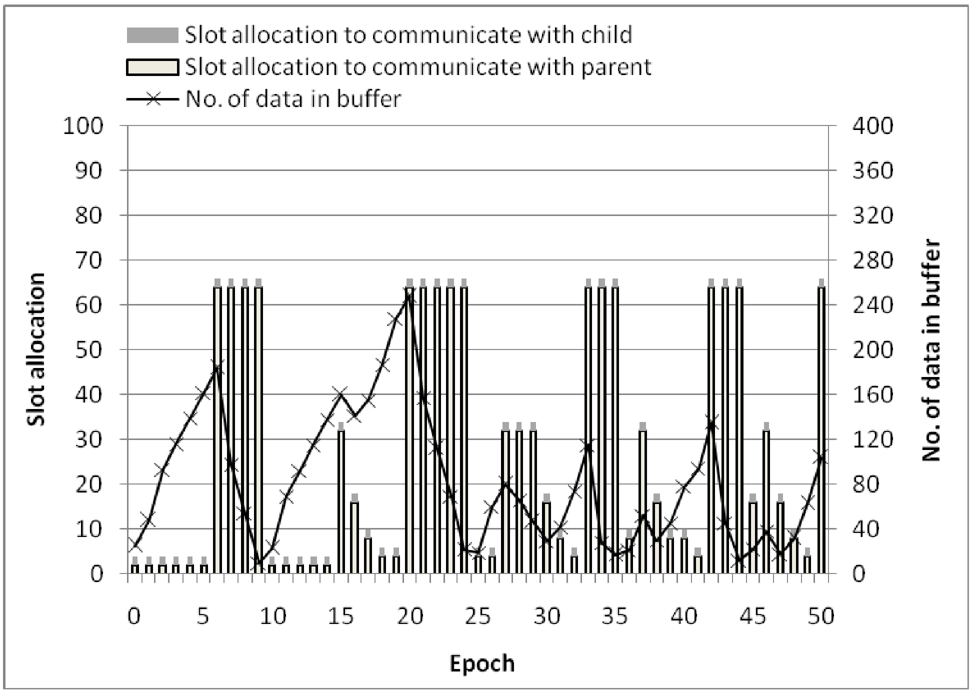

Figure 4.22. Slot allocation and buffer condition as a function of the number of cycles for Rule-based CogWSN with Greedy Scoring in Experiment 4.

In Experiment 5, a node is placed close to a base station and the slot allocation to communicate with the parent and child nodes is set to 2 and 64 respectively.

Another child node is placed near to this node, the latter becoming the parent node as illustrated in Figure 4.23. The mentioned results in the following figures (Figures 4.24 to 4.31) are based on Node 1 in Figure 4.23. Both nodes are sampled at a rate of 0.5Hz. For Rule-based CogWSN, no adjustment is made to the transmission power (Figure 4.24) since the task is to manage the slot allocation. As shown in Figure 4.25, the goal is achieved in 8 cycles. In this case the data is buffered inconsistently and consequently the slot allocations to communicate with parent and child nodes have to be adjusted accordingly.

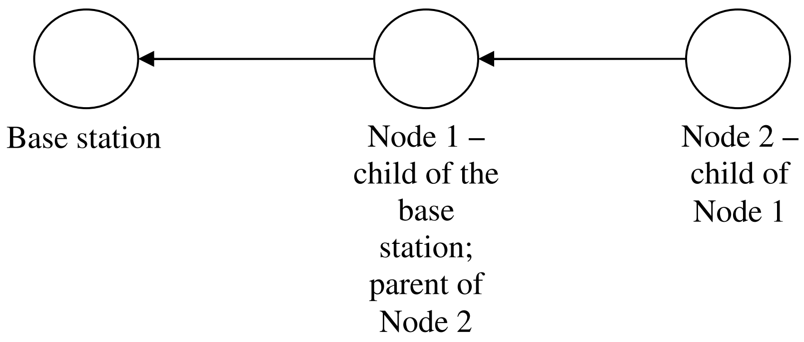

Figure 4.23. A scenario where a node is a child of the base station and is a parent of another child node.

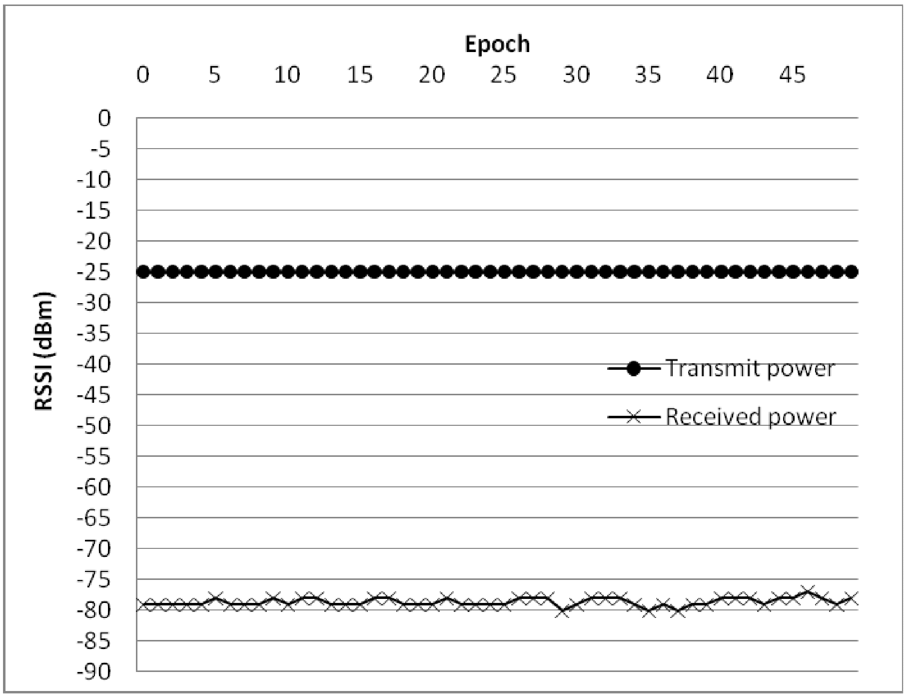

Figure 4.24. RSSI as a function of the number of cycles for Rule-based CogWSN in Experiment 5.

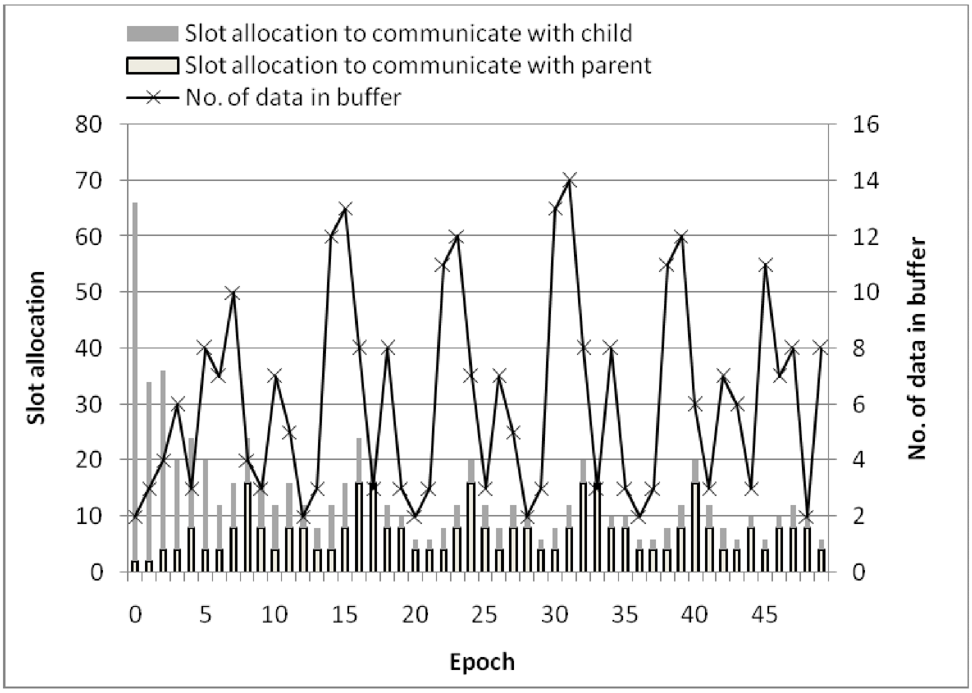

Figure 4.25. Slot allocation and buffer condition as a function of the number of cycles for Rule-based CogWSN in Experiment 5.

For Rule-based CogWSN with Greedy Scoring, no adjustment of the transmission power is required since the RSSI remains between -85dBm to -75dBm (Figure 4.26). The slot allocation to communicate with the parent node the goal is achieved in 13 cycles as shown in Figure 4.27; subsequently, this slot allocation is maintained to s_1 equals 8. Although the buffer size is 3, it should be noted that the node is able to send more than 3 packets during the transmission with the parent node, thus satisfying the slot utilisation criterion of between $\frac{1}{2}s_1$ to s_1.

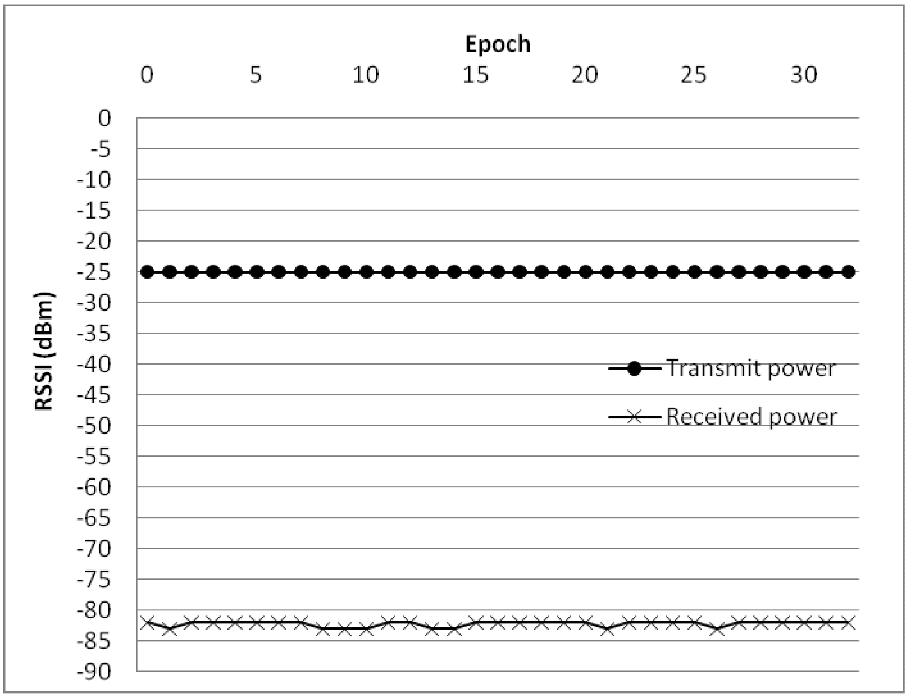

Figure 4.26. RSSI as a function of the number of cycles for Rule-based CogWSN with Greedy Scoring in Experiment 5.

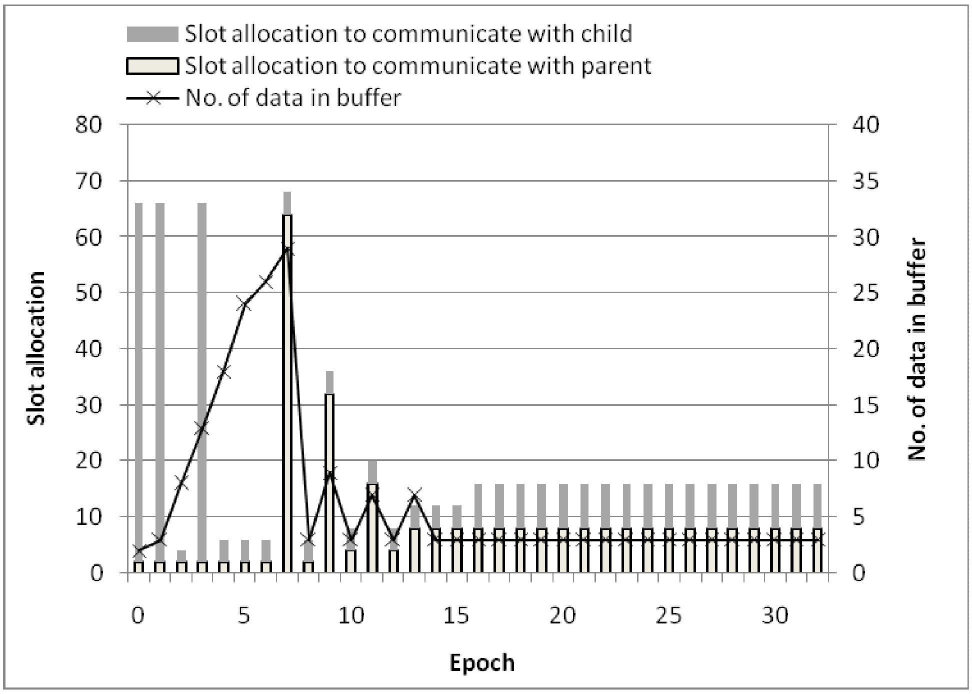

Figure 4.27. Slot allocation and buffer condition as a function of the number of cycles for Rule-based CogWSN with Greedy Scoring in Experiment 5.

In Experiment 6, a node is placed close to a base station and the slot allocation to communicate with both parent and child nodes are both set to 2. Another child node is then placed near to this node, the latter becoming the parent node as illustrated in Figure 4.23. Both nodes are sampled at rate of 0.5Hz. For Rule-based CogWSN, no adjustment of the transmission power is instigated (Figure 4.28) since the main task is that of slot allocation. As shown in Figure 4.29, both slot allocations to communicate with parent and child nodes increment until the 6th cycle and subsequently fluctuate due to the inconsistent data occupancy in the buffer.

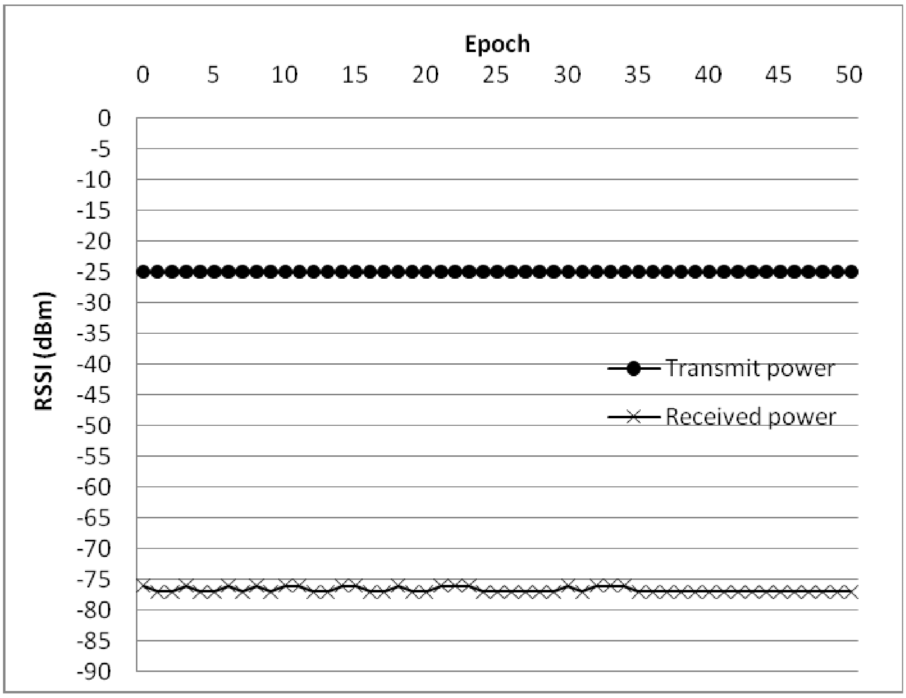

Figure 4.28. RSSI as a function of the number of cycles for Rule-based CogWSN in Experiment 6.

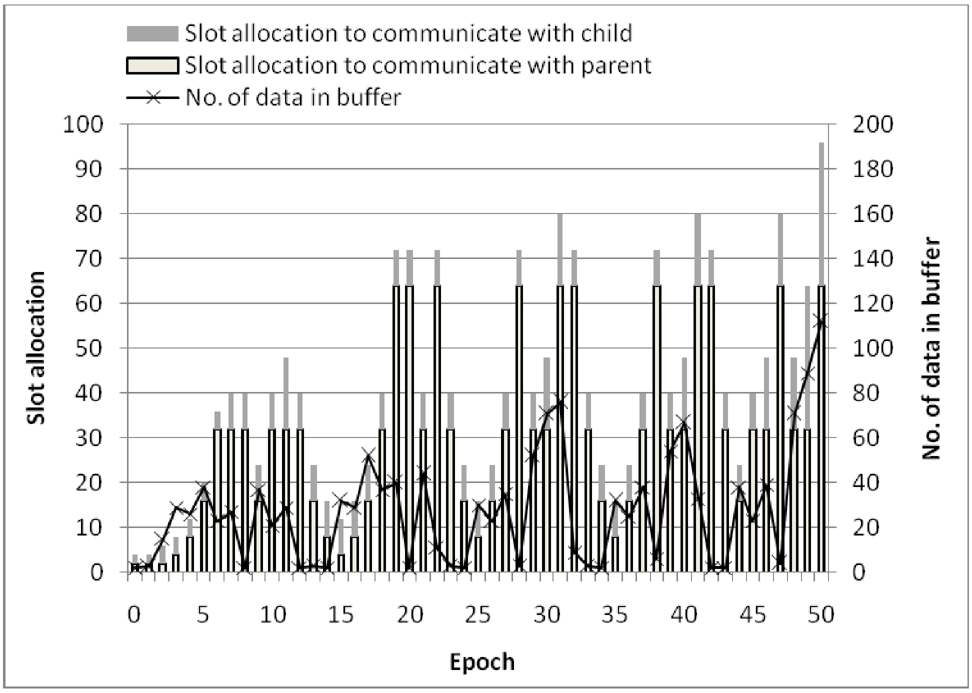

Figure 4.29. Slot allocation and buffer condition as a function of the number of cycles for Rule-based CogWSN in Experiment 6.

For Rule-based CogWSN with Greedy Scoring, no adjustment is being carried out for the transmission power since the RSSI remains between -85dBm to -75dBm (Figure 4.30). The slot allocation to communicate with the parent node, the goal is achieved in 9 cycles as shown in Figure 4.31. At the 20th cycle and onwards, the slot allocation is maintained at s_1 equals 8, satisfying the slot utilisation rule of between $\frac{1}{2}s_1$ to s_1.

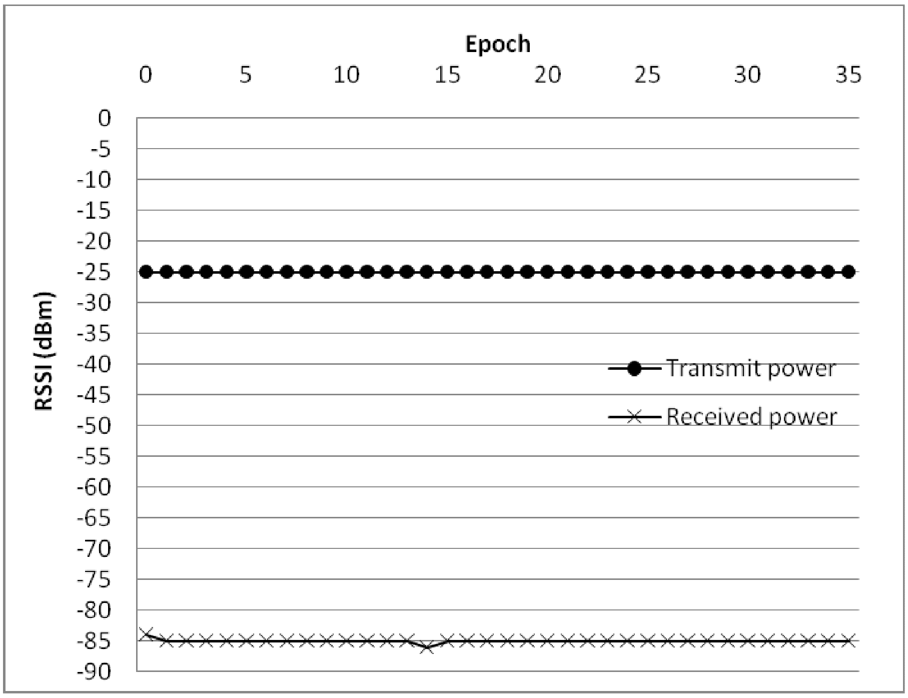

Figure 4.30. RSSI as a function of the number of cycles for Rule-based CogWSN with Greedy Scoring in Experiment 6.

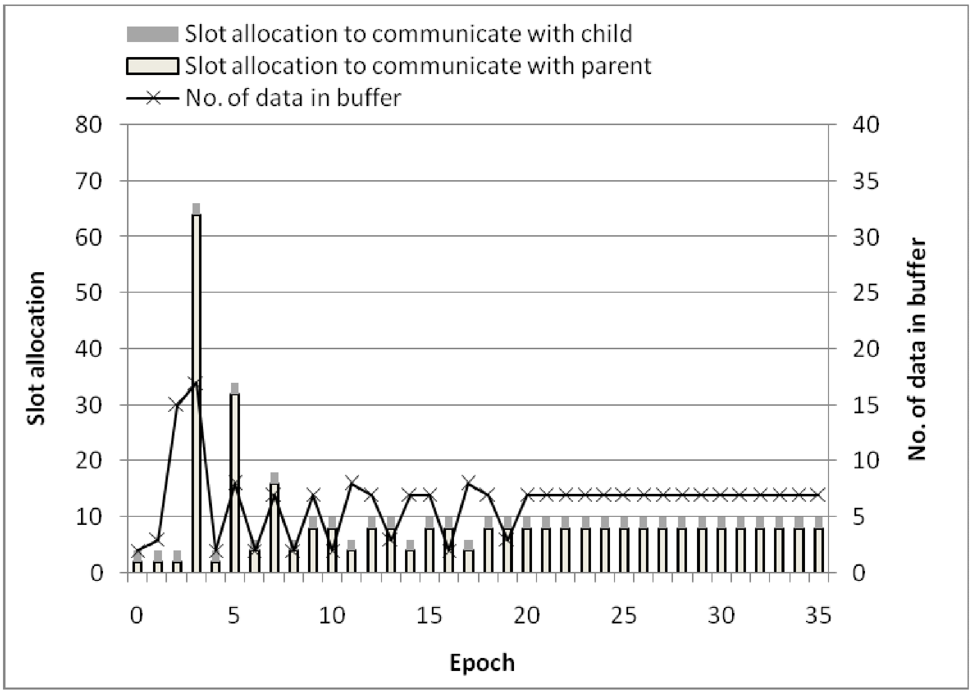

Figure 4.31. Slot allocation and buffer condition as a function of the number of cycles for Rule-based CogWSN with Greedy Scoring in Experiment 6.

The goal of each task is achieved when all targets are met. In some cases, the adjustment continues in a repeated pattern even once the target is initially reached.

Figure 4.32 summarises the number of cycles required to achieve the goal for Rule-based CogWSN (RBA) and Rule-based CogWSN with Greedy Scoring (RBL) from Experiments 1 to 6 (Exp1 to Exp6), where the goal are the combined targets of;

- received power between -85dBm to -75dBm,
- slot utilisation to parent node, x is $\frac{1}{2}s_1 \leq x \leq s_1$, where slot allocation to communicate with parent node, s_1 = 2, 4, 8, 16, 32, or 64, and
- slot utilisation from children nodes, y is $\frac{1}{2}s_2 \leq y \leq s_2$, where slot allocation to communicate with child node, s_2 = 2, 4, 8, 16, 32, or 64.

RBA outperforms RBL for Experiment 5 and Experiment 6 since additional cycles are needed for RBL to obtain the correct score for the rules to achieve the goal; for the remaining experiments, RBL and RBA exhibit similar performance. It is noted that the RBA performs dynamic adjustment in order to maintain its goals; thresholding with hysteresis can be used to control the fluctuations [225].

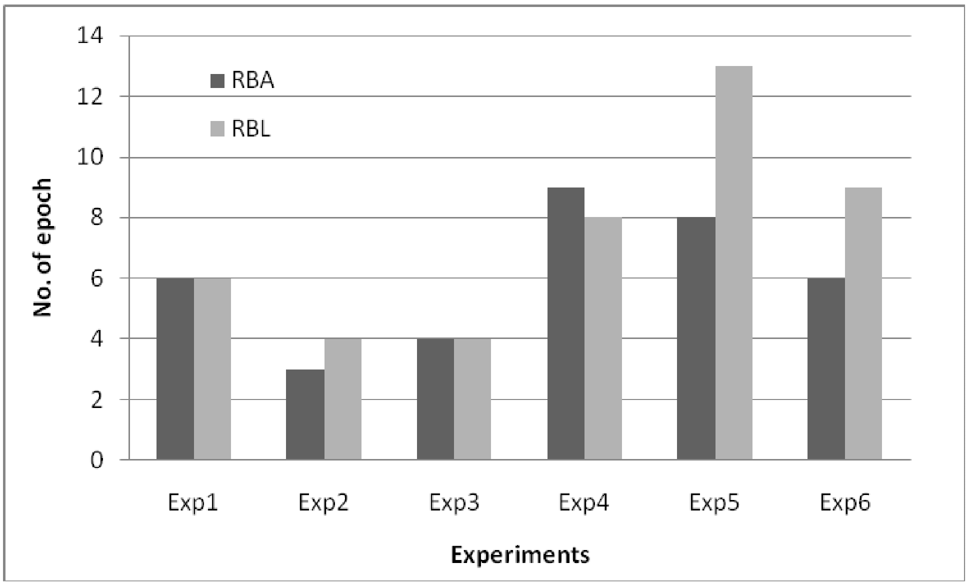

Figure 4.32. Number of cycles required to achieve the goal for Rule-based CogWSN and Rule-based CogWSN with Greedy Scoring.

In order to compare the performance of the CogWSN for different approaches, the average of the result for additional 10 cycles (average number of cycles required to achieve the goal) after the goal is achieved are taken into the consideration, to investigate how well the CogWSN performs in maintaining the goal after it is achieved. Two performance metrics are defined;

- the power usage, i.e. the average of power transmission used to transmit data packets as defined in Equation 4.1;
- and the slot utilisation, i.e. the average of buffer utilisation over slot allocation as defined in Equation 4.2.

Power usage = Sum of the additional cycles of transmission power / The number of additional cycles (4.1)

Slot utilisation = Sum of (current buffer / slot allocation to parent) / The number of additional cycles (4.2)

Figure 4.33 shows the transmission power comparison between RBA and RBL. For Experiment 2, when the power transmission for RBA and RBL are at -10dBm, the received power lies around the limit of receiver sensitivity of -75dBm. In cycles where the received power is slightly above -75dBm, RBL attempts to lower its power transmission to -15dBm, resulting in less power consumption as compared to RBA. For the remaining experiments, little difference in performance between RBL and RBA is evident. Overall, both RBA and RBL are able to minimise the transmission power to 0.1mW and below for this set of experiments.

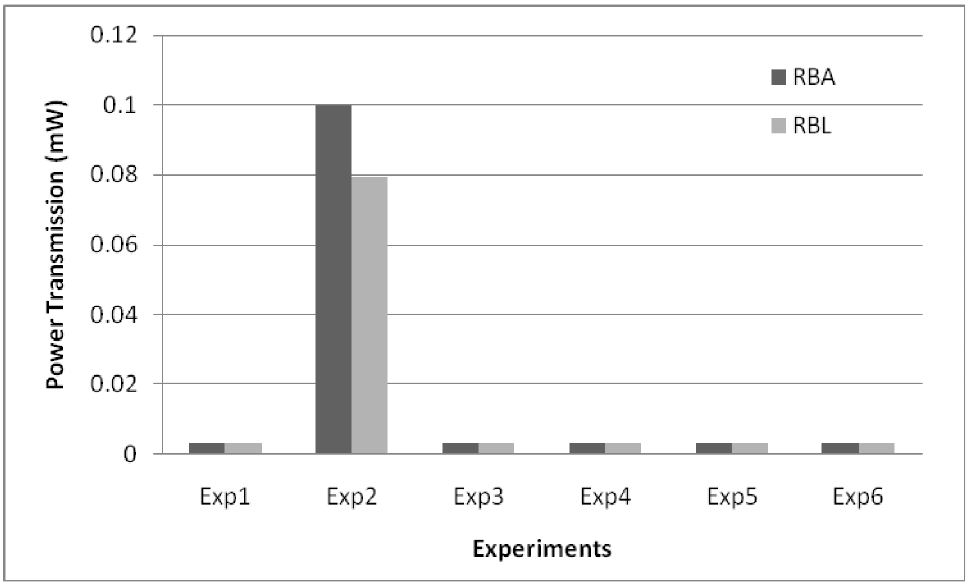

Figure 4.33. Power transmission comparison between Rule-based CogWSN and Rule-based CogWSN with Greedy Scoring.

Figure 4.34 shows the slot utilisation comparison between Rule-based CogWSN and Rule-based CogWSN with Greedy Scoring; ideally slot utilisation should lie between 0.6 to 0.8 [226]. If the slot utilisation is below 0.6, a significant number of slots allocated are not utilised; if the slot utilisation is above 0.8, there is a risk that slots allocated are insufficient to transport packets on slight increases in the number of buffered packets. For the purposes of the present comparison, a slot utilisation of between 0.5 to 1 is deemed acceptable since at least half of the slots allocated are utilised [96]. Overall, RBL and RBA exhibit a similar performance for Experiments 1 to 3. For Experiment 4, RBL provides very poor performance when the target is not achieved; it needs several cycles in order to tune its score to correctly match the rules. Consequently RBL suffers from delays in allocating additional slots and captures a large number of packets in the buffer. For Experiment 5, RBL produces low utilisation of the allocated slots, with more slots allocated than required. For Experiments 6, RBA and RBL cannot meet the acceptable level of performance due to, on average, the number of slots allocated is smaller than the slots required.

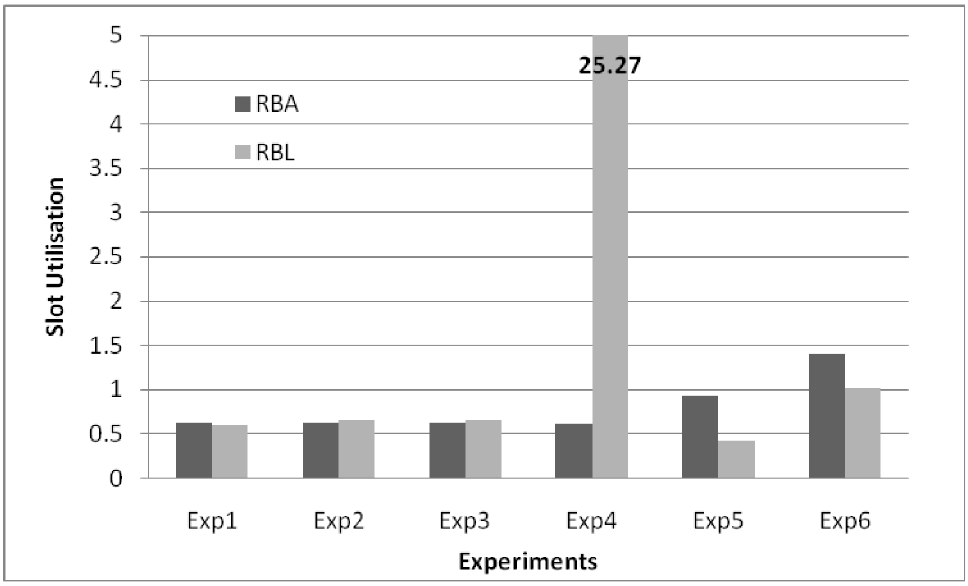

Figure 4.34. Slot utilisation comparison between Rule-based CogWSN and Rule-based CogWSN with Greedy Scoring.

4.7 Conclusions

Rule-based CogWSN without learning capability and rule-based learning are presented. The adaptation of the proposed schemes is embedded into the CogWSN's cognitive cycle. Overall Rule-based CogWSN is expected to present an ideal performance in any situation since the intelligence capturing the task is pre-installed, comprising a complete set of rules on the sensor nodes before deployment. An extension to the cognition to enhance the performance can be implemented by assigning a score to each rule, implemented through tuning a setting with a parameter, an example being the Rule-based CogWSN with Greedy Scoring. Since Rule-based CogWSN with Greedy Scoring requires feedback in order to search for the correct rule (as highlighted in Section 4.5), a time penalty results to achieve the goal.

Six experiments were designed, based on a subset of practical scenarios of WSN deployment as shown in Table 4.5.

Table 4.5. Experiments designed based on a subset of practical scenarios.

Scenario	**Experiments and Set-up**	**Initial Parameters**	**Goals**
A node that is close to a base station where its power transmission is too high.	Experiment 1 Setup: Node is placed close to the base station.	Power transmission, 0dBm; slot allocation for parent node, 2; slot allocation for child, 2; sampling rate, 0.5Hz	1) Received power between -85dBm to -75dBm 2) Slot utilisation to parent node, x is $\frac{1}{2}s_1 \leq x \leq s_1$, where slot allocation to communicate with parent node, s_1 = 2, 4, 8, 16, 32, or 64 3) Slot utilisation from children nodes, y is $\frac{1}{2}s_2 \leq y \leq s_2$, where slot allocation to communicate with child node, s_2 = 2, 4, 8, 16, 32, or 64
A node that is moving far away from a base station where initially its power transmission is low due to it is close to the base station.	Experiment 2 Setup: When the node is placed close to the base station and became stable, the node is moved far from the base station but still within the communication range.	Power transmission, -25dBm; slot allocation for parent node, 2; slot allocation for child, 2; sampling rate, 0.5Hz	
A node that allocates too many slots to communicate with a parent and the traffic is suddenly reduced.	Experiment 3 Setup: Node is placed close to the base station.	Power transmission, -25dBm; slot allocation for parent node, 64; slot allocation for child, 2; sampling rate, 0.5Hz	
A node that allocates too few slots to communicate with a parent and the traffic is suddenly generated at a higher rate.	Experiment 4 Setup: Node is placed close to the base station.	Power transmission, -25dBm; slot allocation for parent node, 2; slot allocation for child, 2; sampling rate, 5Hz	
A node that has a child or children that sending packets at low rate and its slot allocation to the child or children is too much.	Experiment 5 Setup: Node is placed close to the base station. Child node is placed near to the parent node.	Power transmission, -25dBm; slot allocation for parent node, 2; slot allocation for child, 64; sampling rate, 0.5Hz; its' child sampling rate, 0.5Hz	
A node that has a child or	Experiment 6 Setup: Node is placed	Power transmission, -25dBm; slot	

children that sending packets at low rate and its slot allocation to the child or children is too little.	close to the base station. Child node is placed near to the parent node.	allocation for parent node, 2; slot allocation for child, 2; sampling rate, 0.5Hz; its' child sampling rate, 0.5Hz	

In Experiment 1, both Rule-based CogWSN and Rule-based CogWSN with Greedy Scoring have the same performance in terms of number of cycles needed to achieve their goals. From Figure 4.7, Rule-based CogWSN requires 4 cycles to reduce its power transmission while from Figure 4.9, Rule-based CogWSN with Greedy Scoring only needs 1 cycle. Rule-based CogWSN tunes step by step while Rule-based CogWSN with Greedy Scoring can adjust at maximum tuning. Both maintain their power transmissions so that packets arrived at the base station with signal strengths between -85dBm to -75dBm for the next 10 cycles after their goals are achieved. From Figure 4.8 and Figure 4.10, both require 6 cycles in order to adjust their slot utilisations to between 0.6 to 0.8.

In Experiment 2, both Rule-based CogWSN and Rule-based CogWSN with Greedy Scoring exhibit almost the same performance in terms of number of cycles needed to achieve their goals. On significant distances from the base station, nodes increase their power transmissions in order to transmit packets to the base station at signal strength between -85dBm to -75dBm in 3 to 4 cycles as shown in Figure 4.11 and Figure 4.13.

In Experiment 3, both Rule-based CogWSN and Rule-based CogWSN with Greedy Scoring take 4 cycles to achieve their goals. Both do not adjust their power transmission as it does not contribute to the problem to be solved as shown in Figure 4.15 and Figure 4.17. In Figure 4.16, each adjustment for Rule-based CogWSN is an improvement towards solving the problem. In Figure 4.18, Rule-based CogWSN with Greedy Scoring always invokes the maximum adjustment.

In Experiment 4, at the outset, the buffer fills up at a faster rate since the slots allocated to transmit packets to the base station are too few. Both use almost the same number of cycles to achieve the goals. Rule-based CogWSN provides a constant slot allocation to communicate with the base station after the goals are achieved but Rule-based CogWSN with Greedy Scoring has to adjust dynamically as shown in Figure 4.20 and Figure 4.22 respectively. Again, both do not adjust their power transmission as it is not the problem to be solved as shown in Figure 4.19 and Figure 4.21. Since for this scenario, Rule-based CogWSN with Greedy Scoring requires dynamic adjustment in order to maintain its goals, thus rendering the learning mechanism ineffective.

In Experiment 5, Rule-based CogWSN with Greedy Scoring requires more cycles than Rule-based CogWSN in order to achieve its goals. From Figure 4.27, after the goals are achieved, Rule-based CogWSN with Greedy Scoring allocates a constant number of slots to communicate with the base station while from Figure 4.25, Rule-based CogWSN has to adjust dynamically.

In Experiment 6, again, Rule-based CogWSN with Greedy Scoring requires more cycles than Rule-based CogWSN in order to achieve its goals. A similar behaviour as in Experiment 5 is observed for Figure 4.29 and Figure 4.31. For both, after the goals are achieved, on average the number of slots allocated is slightly smaller than the slots required.

In conclusion, Rule-based CogWSN makes adjustment step by step. Each adjustment contributes some level of improvement in solving the problem, while Rule-based CogWSN with Greedy Scoring tends to solve the problem as fast as it can; in some scenarios, the outcome is not as expected. Both have their own strengths and weaknesses in different scenarios as shown in Table 4.6.

Table 4.6. Strengths and weakness for Rule-based CogWSN and Rule-based CogWSN with Greedy Scoring.

	Rule-based CogWSN		**Rule-based CogWSN with Greedy Scoring**	
	Strengths	**Weakness**	**Strengths**	**Weakness**
Experiment 1	Fast in Goal 2 achievement	Slow in Goal 1 achievement	Fast in Goal 1 achievement	Slow in Goal 2 achievement
Experiment 2	Fast in Goal 1 achievement	None	None	Slow in Goal 2 achievement
Experiment 3	None	None	None	None
Experiment 4	Constantly maintain the goals once the goals are achieved	None	None	Dynamically adjust in order to maintain the goals
Experiment 5	None	Dynamically adjust in order to maintain the goals	Constantly maintain the goals once the goals are achieved	None
Experiment 6	None	Dynamically adjust in order to maintain the goals	Constantly maintain the goals once the goals are achieved	None

Chapter 5: Supervised CogWSN

The approach developed in Chapter 4 relies on the ability to learn parameters embedded within pre-determined rules derived from real scenarios. There are a few potential disadvantages to such an approach: (1) in some situations the rules may be difficult to derive, and (2) rules derived by humans may not be able to capture the more subtle features of the decision process. In this Chapter, CogWSN is considered in the framework of supervised learning (based on the review in Section 2.4) i.e. the decision process is assumed to be a function to be learned from given examples, and an Artificial Neural Network (ANN) is used to codify an approximation to the function.

The Chapter starts with an introduction to ANNs and their suitability for the CogWSN's decision process. The detail of each phase of the decision process is discussed and characterised in the following sections.

5.1 Supervised Learning

An ANN is a machine learning technique frequently used for function approximation [151]. It consists of an interconnected group of artificial neurons, a model based on the biological network of neurons in the brain. There are two issues which influence the performance of ANNs in application implementation; the representation of the power of the network and the learning algorithm. The representation of power of an ANN refers to its ability to represent a desired function accurately whilst the learning algorithm is a procedure to find a set of optimal weights in the network.

A basic ANN is formed through a single-layer network (also called as single-layer perceptron), which consists of one or more outputs, o, each of which is connected with a weighting factor, w_{io}, to all of the inputs, i (Figure 5.1).

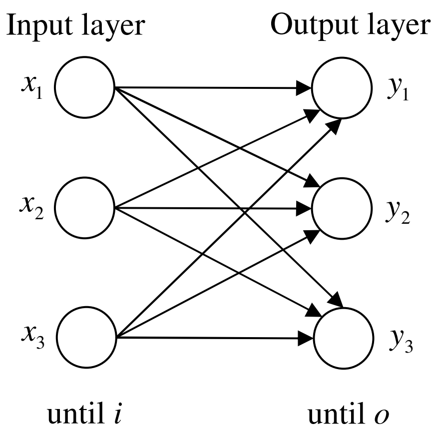

Figure 5.1. A single-layer ANN network.

The network rule uses the output of a threshold function. Take a simple case in which a network has two inputs, x_1 and x_2, a single output, y, and a bias term or offset, θ (Figure 5.2).

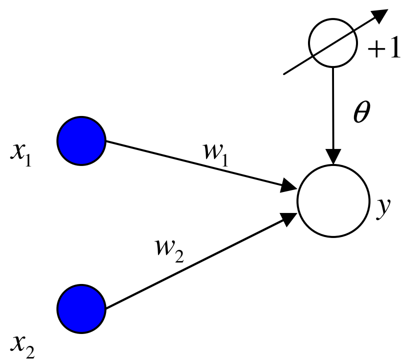

Figure 5.2. An example of a simple ANN single-layer network.

The output of the artificial neuron is the weighted sum of the inputs plus the bias term. The output of the network is formed by the activation of the output artificial neuron, some function of the input as in the Equation 5.1;

$$y = \sum_{i=1}^{2} w_i x_i + \theta \tag{5.1}$$

The activation function F can be linear or non-linear depending on the threshold such as Signum, Heaviside, or Sigmoid [151]. An example of the activation function using the Heaviside step function is (Equation 5.2);

$$F(y) = \begin{cases} +1 & if\ y > 0 \\ -1 & otherwise \end{cases} \tag{5.2}$$

From the above example, the output of the network is either +1 or -1. If the total input is positive, the output is assigned to class +1; if the total input is 0 or negative, then the output is -1.

Consider a set of learning samples consisting of inputs x (x_1, x_2, …, x_i) and a desired output $d(x)$. The **perceptron learning rule** can be stated as:

1. Start with random weights for the connections;
2. Select an input vector x from the set of training samples;
3. If $y \neq d(x)$ (incorrect response), modify all connections w_i according to:
   $\Delta w_i = d(x)x_i$;
4. Go back to 2 until all the training samples are exhausted.

The threshold is modified according to (Equation 5.3);

$$\Delta\theta = \begin{cases} 0 & if\ the\ artificial\ neuron\ responds\ correctly \\ d(x) & otherwise \end{cases} \tag{5.3}$$

For each weight, a new value is computed by adding a correction to the old value as in Equation 5.4. The same computation is also carried out for the offset as in Equation 5.5.

$$w_i(t+1) = w_i(t) + \Delta w_i(t) \tag{5.4}$$

$$\theta(t+1) = \theta(t) + \Delta\theta(t) \tag{5.5}$$

A single-layer network can also use the **delta learning rule** [151]; the delta learning rule uses the network output without further mapping into desired output values. As shown in Figure 5.3, the output is defined as in Equation 5.6;

$$y = \sum_{i=1}^{n} w_i x_i + \theta \tag{5.6}$$

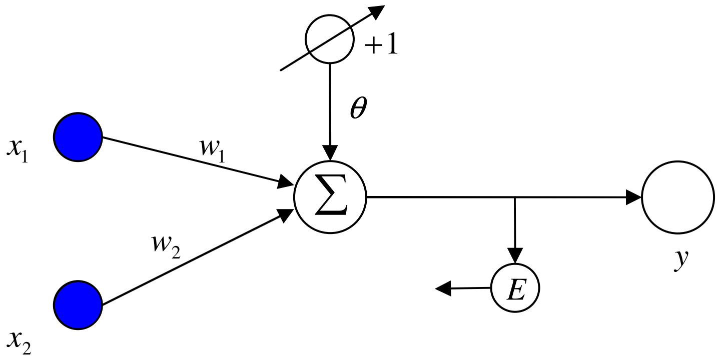

Figure 5.3. A feed-forward network with Delta rule.

The network is trained with input values x^p and desired output values d^p. For every given input sample, the output of the network differs from the target value d^p by $(d^p - y^p)$, where y^p is the actual output for this pattern. The delta-rule now uses a cost- or error-function based on these differences to adjust the weights. The error function, referred to as Least Mean Square (LMS), is the sum of squared error, where E is defined as (Equation 5.7) [227];

$$E = \sum_p E^p = \frac{1}{2}\sum_p (d^p - y^p)^2 \tag{5.7}$$

where the index p ranges over the set of input patterns and E^p represents the error on pattern p. The LMS procedure finds the values of all weights that minimise the error function by the gradient descent method [151] viz. a change in the weight is made proportional to the negative of the derivative of the error as measured on the current pattern with respect to each weight (Equation 5.8) [227];

$$
\begin{aligned}
\Delta_p w_i &= -\gamma \frac{\partial E^p}{\partial w_i} \\
&= -\gamma \frac{\partial E^p}{\partial y^p} \frac{\partial y^p}{\partial w_i} \\
&= -\gamma[-(d^p - y^p)]x_i \, , \; \because \frac{\partial E^p}{\partial y^p} = -(d^p - y^p) \, , \; \frac{\partial y^p}{\partial w_i} = x_i \\
&= \gamma(d^p - y^p)x_i
\end{aligned} \tag{5.8}
$$

where γ is a constant of proportionality. The updated weight can be written as (Equation 5.9);

$$w_i(t+1) = w_i(t) + \Delta w_i(t) \tag{5.9}$$

where $\Delta w_i(t) = \eta(d(x) - y)x_i$ and $\eta = \gamma$, also known as the learning rate.

The delta learning rule can be written as:

1. Start with random weights for the connections;
2. Select an input vector x from the set of training samples;
3. If $E \neq 0$ (incorrect response), modify all connections w_i according to:

   $\Delta w_i(t) = \eta(d(x) - y)x_i$;
4. Go back to 2 until all the training samples are exhausted.

Generally, a single-layer network operates well for AND, OR, any m-to-n function, NOT, NAND, NOR, but has limitations with XOR [228]. Therefore, one of the routes to removing this limitation is to add hidden layers to modify the single-layer into a multi-layer network implementation. In order to adjust the weights of each artificial neuron in such a way that the error between the desired and the actual output is reduced requires that the network compute the error derivative of the weights. The back-propagation algorithm is a widely used method for determining the error derivative of weights, calculating how the error changes as each weight is increased or decreased slightly. For Supervised CogWSN, a multi-layer network with back-propagation learning rule (Figure 5.4 with the activation function as in Figure 5.5) is applied. The reason for using a multi-layer network is its capability to solve any function, not restricted to linear functions. The multi-layer network does not require large processing power or memory. As long as initial inputs and desired outputs are mapped, it provides solutions even when inputs are not mapped in advance at the expense of certain errors that can be corrected later using a lookup table.

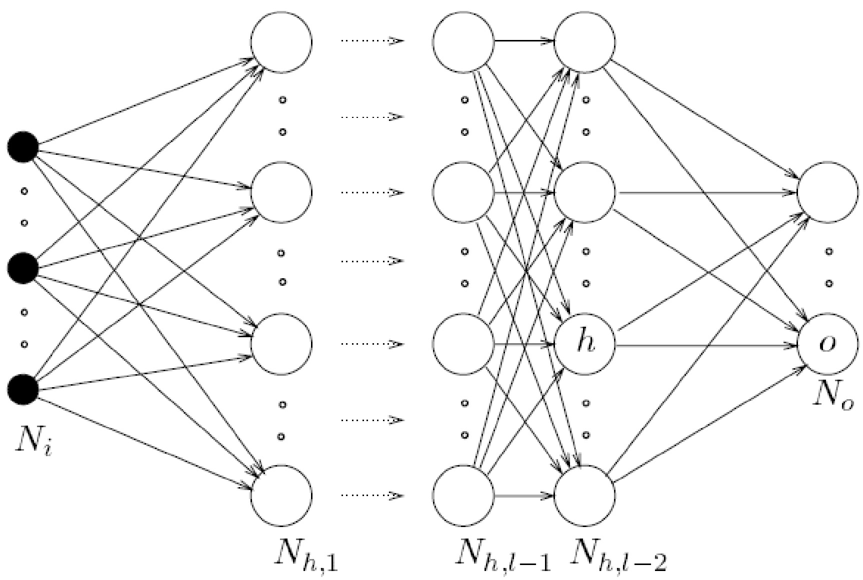

Figure 5.4. A multi-layer ANN network with *l* layers of artificial neurons [227].

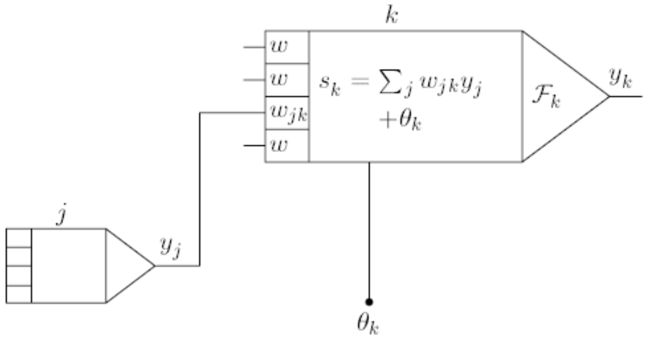

Figure 5.5. A model of the activation function for a multi-layer ANN network [227].

The activation function is as in Equation 5.10;

$$y_k^p = F(s_k^p) \tag{5.10}$$

where $s_k^p = \sum_j w_{jk} y_j^p + \theta_k$. The error measure E^p is defined as the total sum of the squared error for pattern *p* at the output units as in Equation 5.11;

$$E^p = \frac{1}{2}\sum_{o=1}^{N_o}(d_o^p - y_o^p)^2 \tag{5.11}$$

where d_o^p is the desired output for unit *o*. The weight can be written as in Equation 5.12 [227];

$$\begin{aligned}
\Delta_p w_{jk} &= -\gamma \frac{\partial E^p}{\partial w_{jk}} \\
&= -\gamma \frac{\partial E^p}{\partial s_k^p}\frac{\partial s_k^p}{\partial w_{jk}} \\
&= -\gamma(-\delta_k^p)\, y_j^p\,, \because \frac{\partial E^p}{\partial s_k^p} = -\delta_k^p\,, \frac{\partial s_k^p}{\partial w_{jk}} = y_j^p \\
&= \gamma \delta_k^p y_j^p
\end{aligned} \tag{5.12}$$

where γ is a constant of proportionality. δ_k^p for each unit *k* in the network can be found through Equation 5.13 [227];

$$\begin{aligned}\delta_k^p &= -\frac{\partial E^p}{\partial s_k^p} \\ &= -\frac{\partial E^p}{\partial y_k^p}\frac{\partial y_k^p}{\partial s_k^p} \\ &= -\frac{\partial E^p}{\partial y_k^p}F'(s_k^p)\end{aligned} \tag{5.13}$$

Assuming k is an output unit $k=o$ of the network, Equation 5.14 [227] can be obtained for any output unit *o;*

$$\delta_o^p = (d_o^p - y_o^p)F_o'(s_o^p)\,,\ \because \frac{\partial E^p}{\partial y_o^p} = -(d_o^p - y_o^p) \tag{5.14}$$

Assuming k is not an output unit but a hidden unit $k=h$, $E^p = E^p(s_1^p, s_2^p, ..., s_j^p, ...)$, then Equation 5.15 [227] can be obtained;

$$\begin{aligned}\frac{\partial E^p}{\partial y_h^p} &= \sum_{o=1}^{N_o}\frac{\partial E^p}{\partial s_o^p}\frac{\partial s_o^p}{\partial y_h^p} \\ &= \sum_{o=1}^{N_o}\frac{\partial E^p}{\partial s_o^p}\frac{\partial}{\partial y_h^p}\sum_{j=1}^{N_h} w_{ho}y_j^p \\ &= \sum_{o=1}^{N_o}\frac{\partial E^p}{\partial s_o^p}w_{ho} \\ &= -\sum_{o=1}^{N_o}\delta_o^p w_{ho}\end{aligned} \tag{5.15}$$

Substituting Equation 5.15 into Equation 5.13, then Equation 5.14 and Equation 5.16 provide a recursive procedure for computing δ's for all units in the network, which are then used to compute the weight changes according to Equation 5.12;

$$\delta_h^p = F'(s_h^p)\sum_{o=1}^{N_o}\delta_o^p w_{ho} \tag{5.16}$$

For Supervised CogWSN, weight adjustments with the Sigmoid activation function are used [151] since not all combination of inputs can be precisely mapped to the desired outputs. The activation function F is defined as (Equation 5.17);

$$y^{p} = F(s^{p}) = \frac{1}{1+e^{-s^{p}}} \tag{5.17}$$

The change of the weight is updated through (Equation 5.18) [227];

$$\Delta w_{jk}(t) = \gamma\delta_{k}^{p} y_{j}^{p} + \alpha\Delta w_{jk}(t-1) \tag{5.18}$$

where t indexes the presentation number and α is a constant which determines the effect of the previous weight change. Finally, the updated weight can be obtained through Equation 5.19;

$$\begin{aligned} w_{jk}(t+1) &= w_{jk}(t) + \Delta w_{jk}(t) \\ &= w_{jk}(t) + \gamma\delta_{k}^{p} y_{j}^{p} + \alpha\Delta w_{jk}(t-1) \end{aligned} \tag{5.19}$$

5.2 Observe Phase

The Observe phase is implemented using event-based observation as defined in Section 4.2.

5.3 Plan Phase

For this phase, the trigger is best derived based on all detected symptoms since the Supervised CogWSN is able to make decisions using a combination of inputs. Due to space allocation executed when the ANN network is created, a combination of the input will not increase the overhead in terms of space complexity on the decision.

5.4 Implement Phase

With Supervised CogWSN, the option of 'level up' or 'level down' a setting is chosen for the implementation. Referring to Section 4.4, the option of 'tune a setting' with a parameter requires more artificial output neurons to be added from the ANN network, thus increasing the space complexity.

5.5 Evaluate Phase

To increase the accuracy of the decision making, a look up table to filter any incorrect decisions made during the Plan Phase is implemented in this phase. If an incorrect decision is made, a record will be entered into the look up table. The size of the table depends on the memory capacity.

Feedback to retrain the neurons at the nodes is possible but impractical due to limited computation capability of micro-controllers. Another route is to transport feedback to the PC, retrain at the PC, and update the nodes with new trained knowledge. It should be noted that this would generate additional load on the network due to increased communication traffic.

5.6 Verification

A case study is performed to verify the performance of Supervised CogWSN. The objective of the case study is the same as specified in Section 4.6 i.e. to drive the CogWSN to achieve optimum performance in terms of connectivity and slot utilisation. The case study focuses on three metrics; transmission power, slot allocation to parent, and slot allocation to child node. The targets to achieve, monitored conditions, and derived actions for each domain are as shown as in Table 4.2.

The experimental setup as defined in Section 4.6 is configured. A multi-layer network with back-propagation learning using 3-input, 5-hidden [229], and 6-output artificial neurons is implemented. The learning rate γ and momentum α are set to 0.25 and 0.1 respectively. An extremely low learning rate of around 0.001 to 0.005 combination and momentum term between 0.5 to 0.9 do not give satisfactory results for certain specific data set [230]. An average slow learning rate is preferred to avoid oscillation or divergence of the right solution. Momentum is used to prevent the network from converging to local minima. It is noted that a slow learning rate and low momentum results in a longer time to reach convergence, but since the network is trained on a PC, the training time does not impact the overall system performance. The network is trained according to the training inputs and desired outputs as stated in Table 5.1. During link establishment, the knowledge is transferred to a child when the child connects to the parent.

Table 5.1. Training inputs and desired outputs.

Training Inputs	**Desired Outputs**
RSSI < -85dBm	Increase transmit power by 1 level up
RSSI > -75dBm	Decrease transmit power by 1 level down
Slot utilisation to parent node, $x > s_1$, where slot allocation to communicate with parent node, s_1 = 2, 4, 8, 16, 32, or 64	Increase slot allocation by $2s_1$
Slot utilisation to parent node, $x < \frac{1}{2}s_1$, where slot allocation to communicate with parent node, s_1 = 2, 4, 8, 16, 32, or 64	Reduce slot allocation by $s_1/2$
Slot utilisation from children nodes, $y > s_2$, where slot allocation to communicate with child node, s_2 = 2, 4, 8, 16, 32, or 64	Increase slot allocation by $2s_2$
Slot utilisation from children nodes, $y < \frac{1}{2}s_2$, where slot allocation to communicate with child node, s_2 = 2, 4, 8, 16, 32, or 64	Reduce slot allocation by $s_2/2$

Experiments are established as defined in Table 4.4.

In Experiment 1, a node is placed in close proximity to the base station and at the outset, the transmission power is set to the maximum, 0dBm; as a consequence the received power is above -75dBm. After 4 cycles, the received power is adjusted step by step and remains between -85dBm to -75dBm (Figure 5.6). No significant adjustment is carried out for slot allocations as shown in Figure 5.7 since connectivity is maintained and slot utilisation is almost constant.

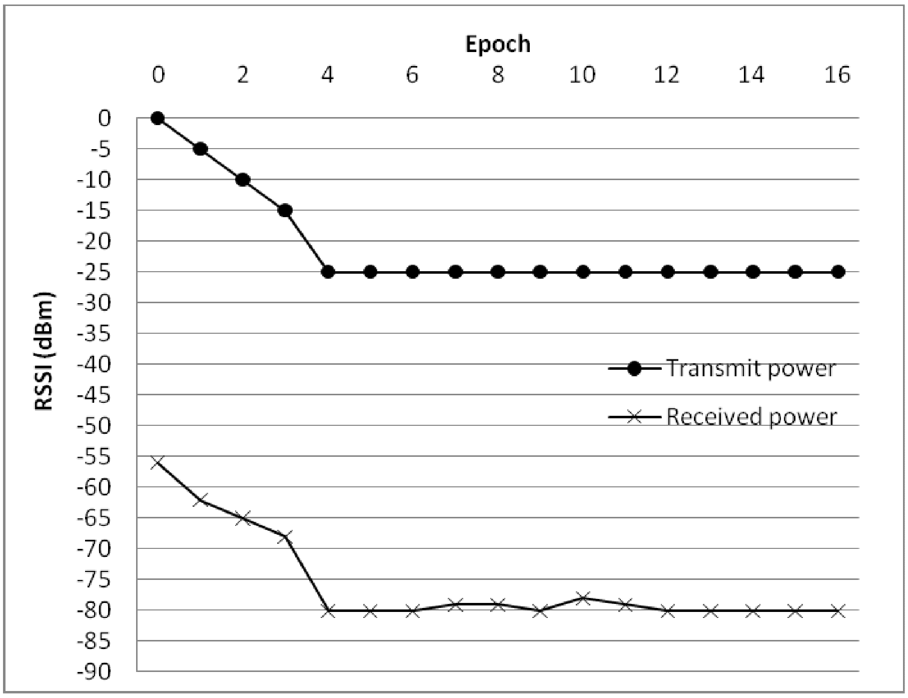

Figure 5.6. RSSI as a function of the number of cycles for Supervised CogWSN in Experiment 1.

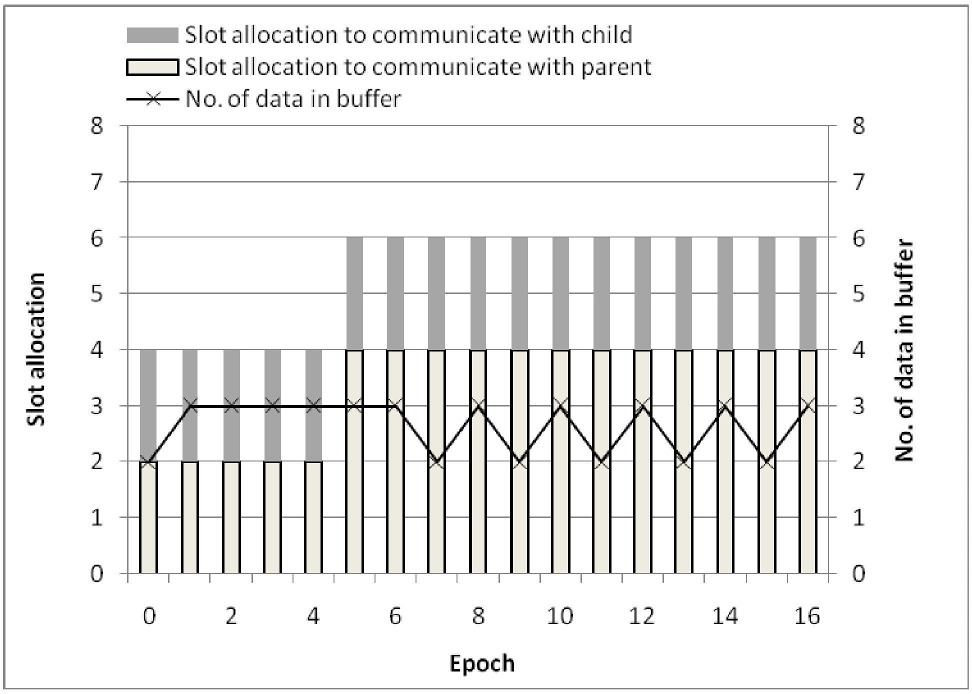

Figure 5.7. Slot allocation and buffer condition as a function of the number of cycles for Supervised CogWSN in Experiment 1.

In Experiment 2, the transmission power is set to the minimum -25dBm and sensor data is sampled at 0.5Hz. At the outset, a node is placed in close proximity to a base station until the received power stabilises. Thereafter, the node is moved further away from the base station but still remains within its communication range. Whenever the node stops receiving an acknowledgement packet, it increases its transmission power towards the maximum. Once the node receives feedback on 'too excessive transmission power', the increments stop. After 2 cycles, the received power is reduced and successfully maintained between -85dBm to -75dBm (Figure 5.8). No adjustment is being carried out for slot allocations as shown in Figure 5.9 since connectivity is maintained and slot utilisation remains almost constant.

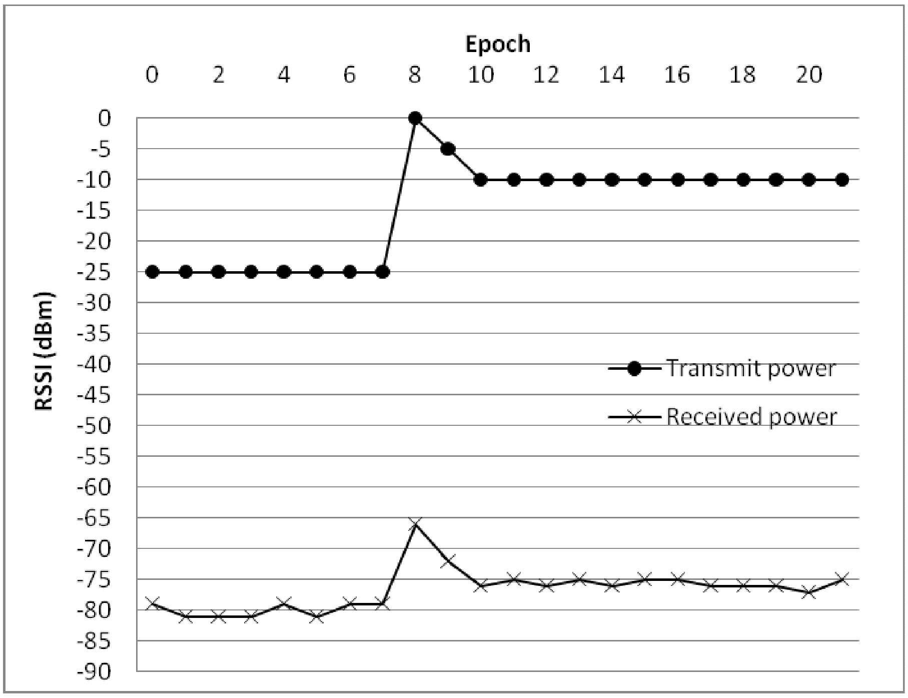

Figure 5.8. RSSI as a function of the number of cycles for Supervised CogWSN in Experiment 2.

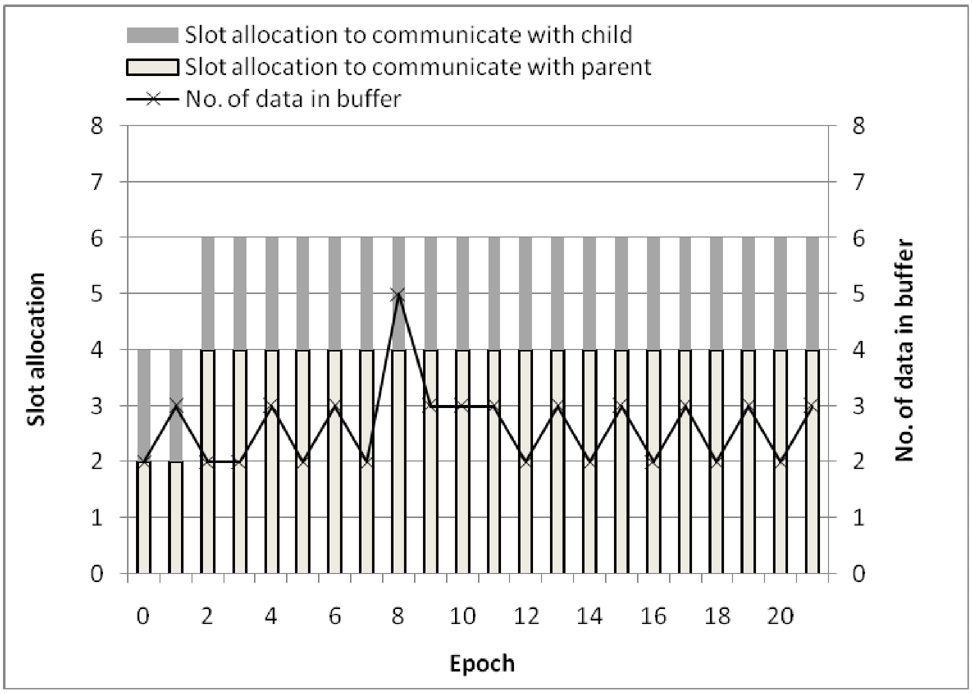

Figure 5.9. Slot allocation and buffer condition as a function of the number of cycles for Supervised CogWSN in Experiment 2.

In Experiment 3, a node is placed in close proximity to a base station, the slot allocation to communicate with the parent node is set to 64 and sensor data is sampled at 0.5Hz. The transmission power is set to -25dBm; so no adjustment to the transmission power is carried out since RSSI remains between -85dBm to -75dBm (Figure 5.10). Since a significant number of allocated slots remain unused, the number of slots is reduced. After 4 cycles, the slot allocation to communicate with the parent node reduces to 4, where s_1 is 4, satisfying the slot utilisation criterion between $\frac{1}{2}s_1$ to s_1 as shown in Figure 5.11.

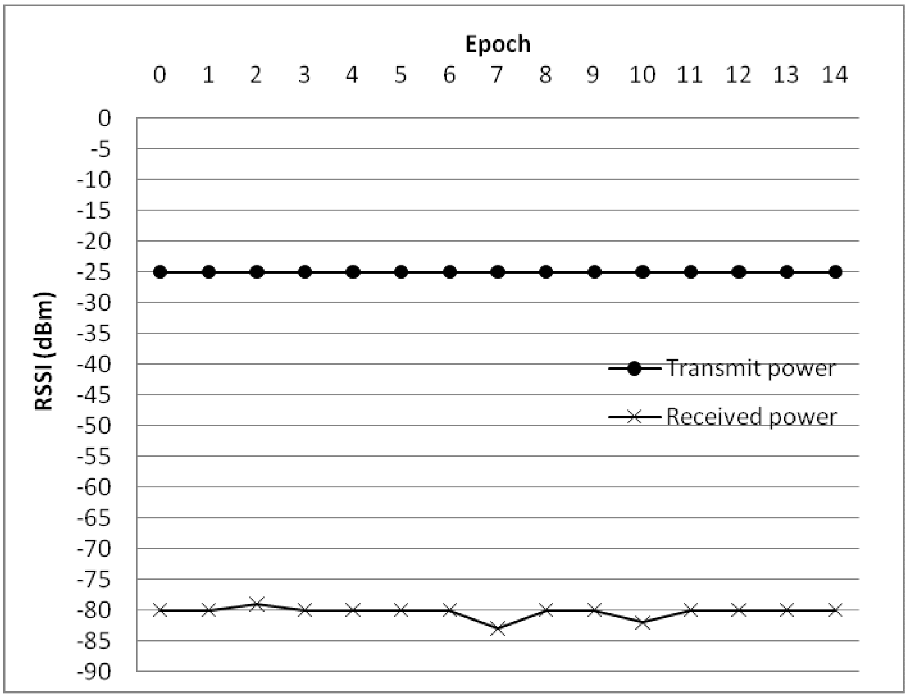

Figure 5.10. RSSI as a function of the number of cycles for Supervised CogWSN in Experiment 3.

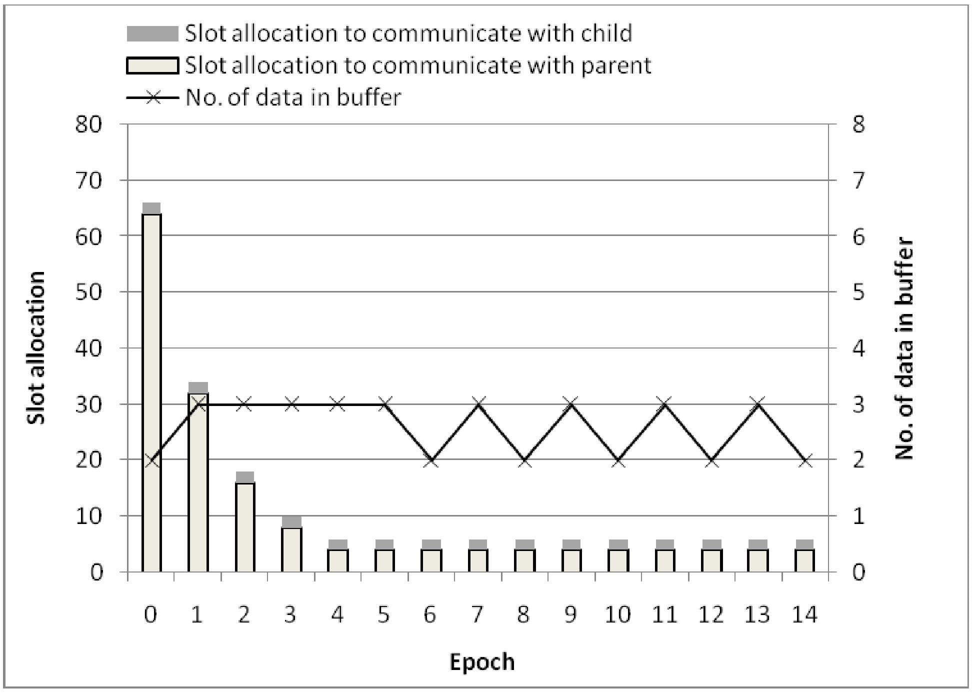

Figure 5.11. Slot allocation and buffer condition as a function of the number of cycles for Supervised CogWSN in Experiment 3.

In Experiment 4, a node is placed in close proximity to a base station; the slot allocation to communicate with the parent node is set to 2 as is the slot allocation to communicate with the child node, with sensor data sampled at 5Hz. The transmission power is set to -25dBm; no adjustment is carried out for the transmission power since the RSSI remains between -85dBm to -75dBm (Figure 5.12). As a consequence of the number of packets needed to be transmitted to the parent node and too few allocated slots, the allocation of slots has to be increased. After 13 cycles, the slot allocation to communicate with the parent node is successfully increased to 32, where s_1 is 32, satisfying the slot utilisation of between $\frac{1}{2}s_1$ to s_1 (Figure 5.13).

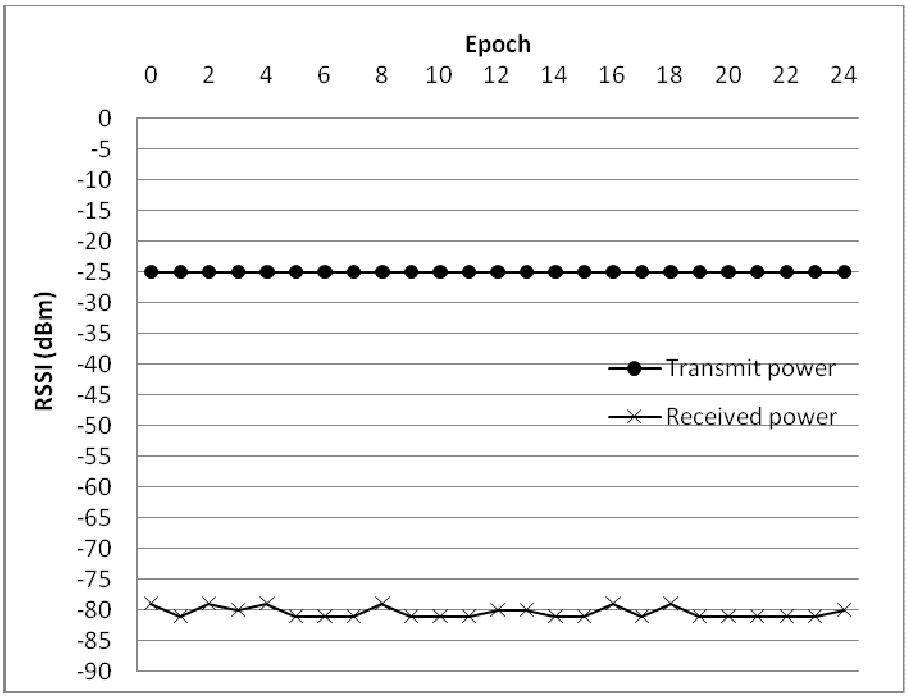

Figure 5.12. RSSI as a function of the number of cycles for Supervised CogWSN in Experiment 4.

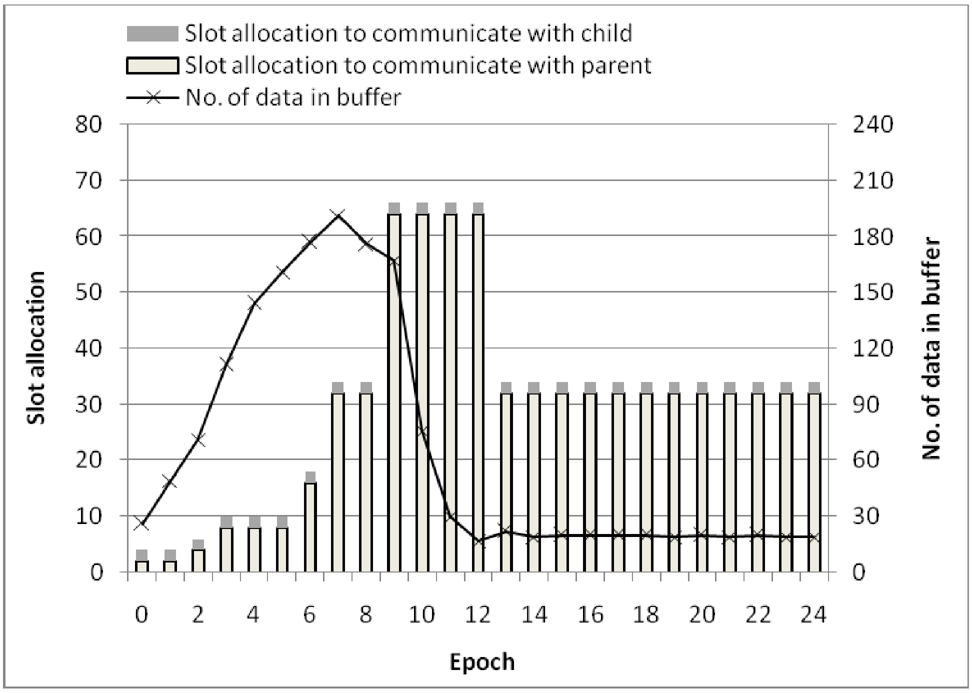

Figure 5.13. Slot allocation and buffer condition as a function of the number of cycles for Supervised CogWSN in Experiment 4.

In Experiment 5, a node is placed in close proximity to a base station and the slot allocation to communicate with the parent and child nodes are set to 2 and 64 respectively. Another child node is placed near to this node, the latter becoming the parent node as illustrated in Figure 4.23. Both nodes run at a sampling rate of 0.5Hz. No active adjustment is carried out for the transmission power as in Figure 5.14 since the focus of the task is to optimise slot allocation. As shown in Figure 5.15, after 23 cycles, both slot allocations to communicate with the parent and child nodes are maintained between 2 to 8.

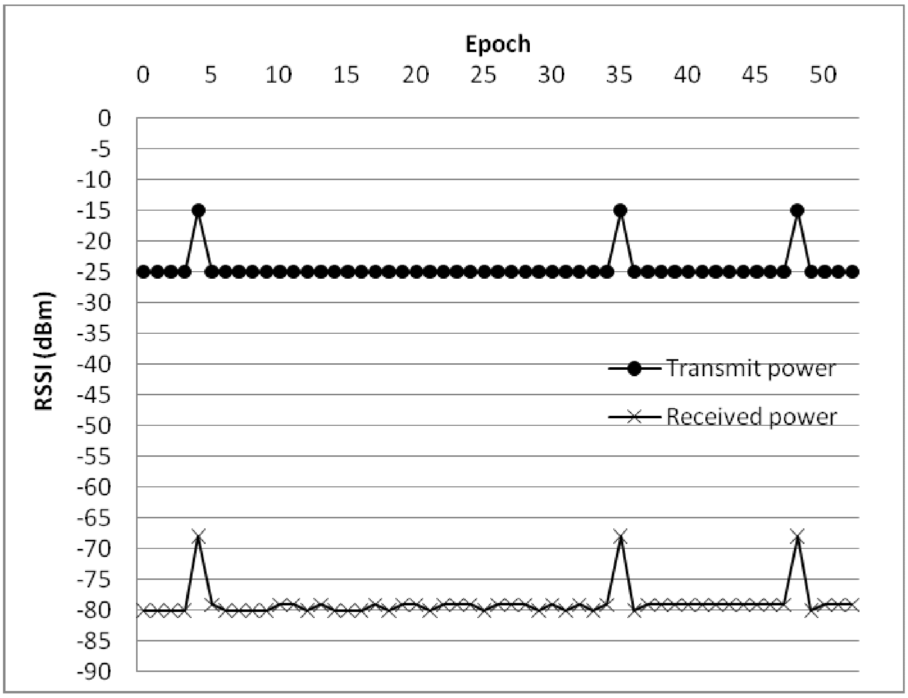

Figure 5.14. RSSI as a function of the number of cycles for Supervised CogWSN in Experiment 5.

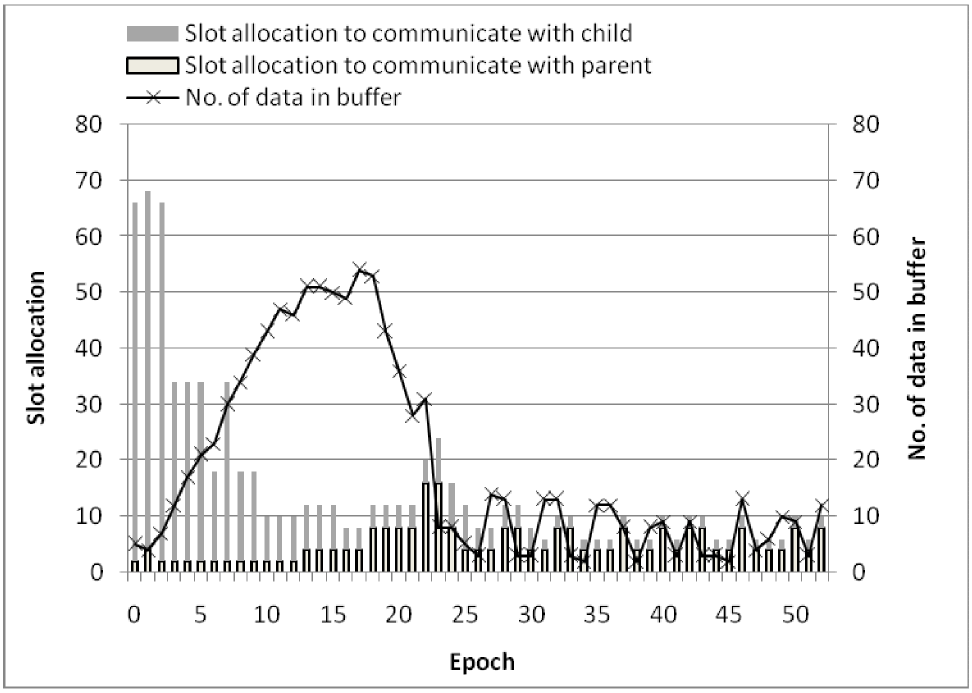

Figure 5.15. Slot allocation and buffer condition as a function of the number of cycles for Supervised CogWSN in Experiment 5.

In Experiment 6, a node is placed in close proximity to a base station and the slot allocation to communicate with both the parent and child nodes is set to 2. Another child node is placed near to this node, effectively becoming its parent node. Both nodes run at a sampling rate of 0.5Hz. Once again, no adjustment to the transmission power is necessary (Figure 5.16) since the focus of the task is to optimise slot allocations. As shown in Figure 5.17, both slot allocations to communicate with parent and child nodes have been increased from the 14th cycle onwards and fluctuates due to the data in the buffer being filled up quickly and inconsistently from the child node.

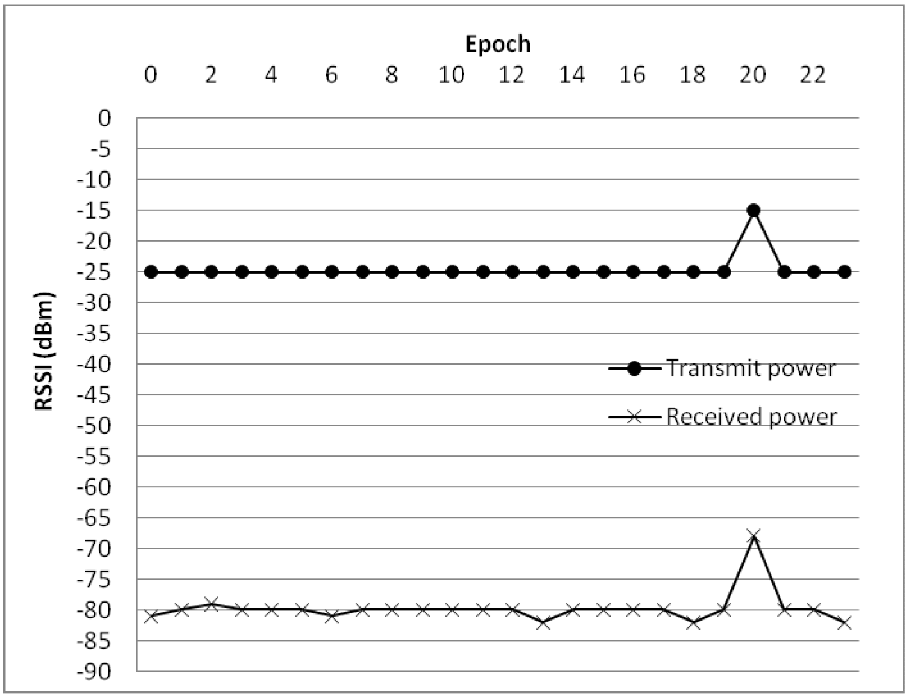

Figure 5.16. RSSI as a function of the number of cycles for Supervised CogWSN in Experiment 6.

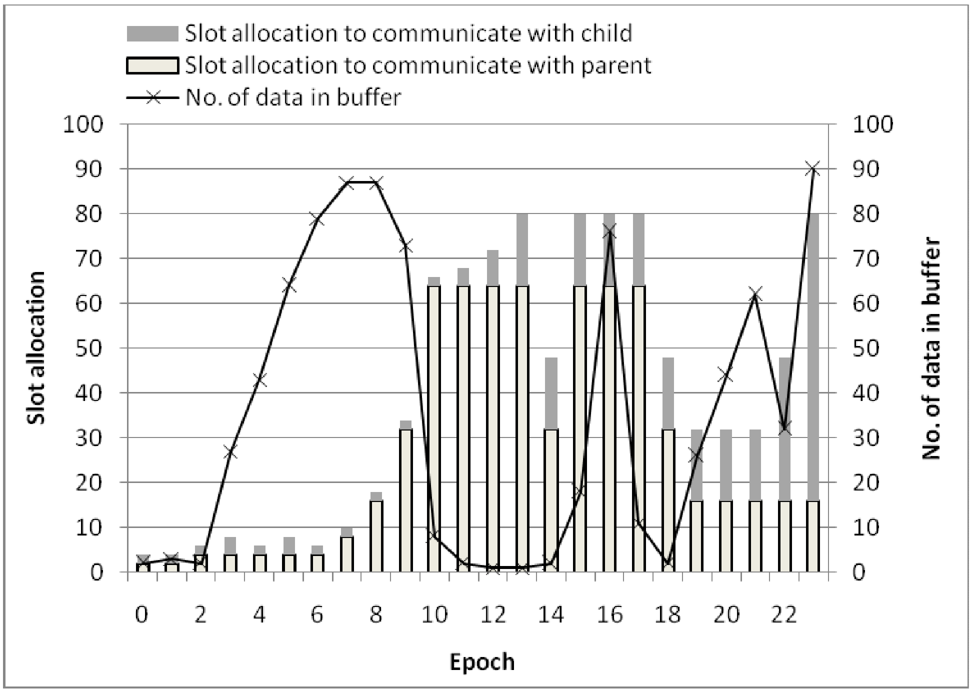

Figure 5.17. Slot allocation and buffer condition as a function of the number of cycles for Supervised CogWSN in Experiment 6.

Table 5.2 and Figure 5.18 show a speed of adjustment comparison between Rule-based CogWSN (RBA), Rule-based CogWSN with Greedy Scoring (RBL), and Supervised CogWSN (SL) derived from Experiments 1 to 6 (Exp1 to Exp6). Supervised CogWSN is able to achieve a performance close to that of a Rule-based CogWSN except for Experiment 5. Referring to Table 5.1, Supervised CogWSN is not trained with a combination of inputs; too few slots are allocated to the parent and too many slots are allocated to the child. Under this scenario, more cycles are required in order to find the correct output. It should be noted that if Supervised CogWSN is trained with a combination of inputs, the performance of Supervised CogWSN from Experiments 1 to 6 is further improved.

Table 5.2. Number of cycles to achieve targets for Rule-based CogWSN, Rule-based CogWSN with Greedy Scoring and Supervised CogWSN.

Experiments and Initial Setup	**No. of cycles to achieve the goal for Rule-based CogWSN**	**No. of cycles to achieve the goal for Rule-based CogWSN with Greedy Scoring**	**No. of cycles to achieve the goal for Supervised CogWSN**
Experiment 1	6	6	5
Experiment 2	3	4	2
Experiment 3	4	4	4
Experiment 4	9	8	13
Experiment 5	8	13	23
Experiment 6	6	9	10

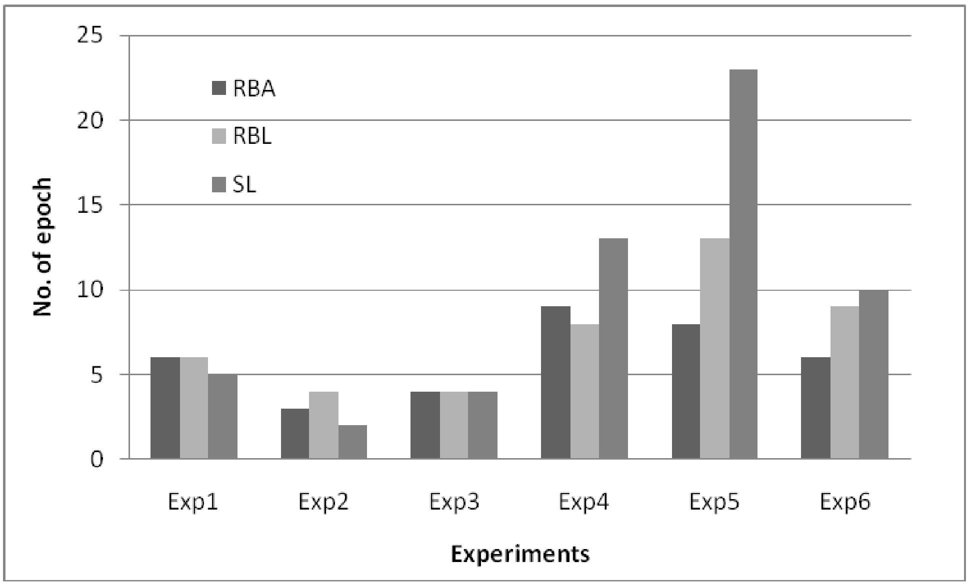

Figure 5.18. Speed of adjustment comparison between Rule-based CogWSN, Rule-based CogWSN with Greedy Scoring, and Supervised CogWSN.

Figure 5.19 shows a transmission power comparison between RBA, RBL, and SL; SL exhibits a similar performance to RBA. In general, all methods maintain the minimum power consumption needed for all communication after the goals are achieved.

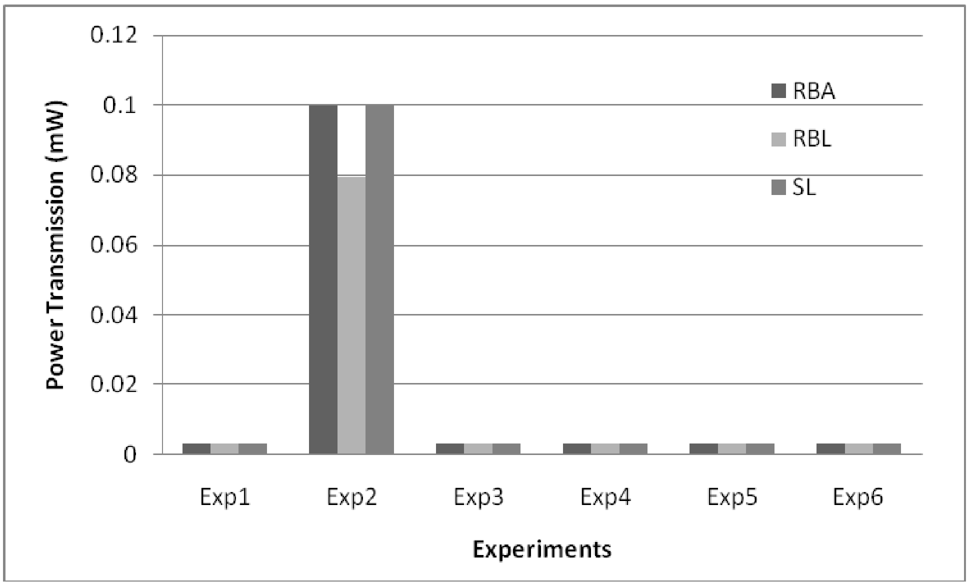

Figure 5.19. Power transmission comparison between Rule-based CogWSN, Rule-based CogWSN with Greedy Scoring, and Supervised CogWSN.

Figure 5.20 shows a slot utilisation comparison between RBA, RBL, and SL. As discussed in Section 4.6, ideally, slot utilisation should lie between 0.6 to 0.8. If the slot utilisation falls below 0.6, a significant number of slots allocated are not utilised; if the slot utilisation is above 0.8, there is a risk that the slots allocated are insufficient to transport all packets with a modest increase in the number of buffered the packets. For the purposes of the present comparison, a slot utilisation of between 0.5 to 1 is deemed acceptable since at least half of the slots allocated are utilised. Overall, RBL offers better performance except for Experiment 4 since RBL allocates fewer slots and captures a significant number of packets in the buffer. SL exhibits similar performance to RBA for Exp 1 to Exp 4. For Exp 5, SL sometimes allocates fewer slots than required; therefore, on average, it provides the highest slot utilisation. For Exp 6, after the goals are achieved, RBL allocates more slots than required; therefore, it suffers from low slot utilisation.

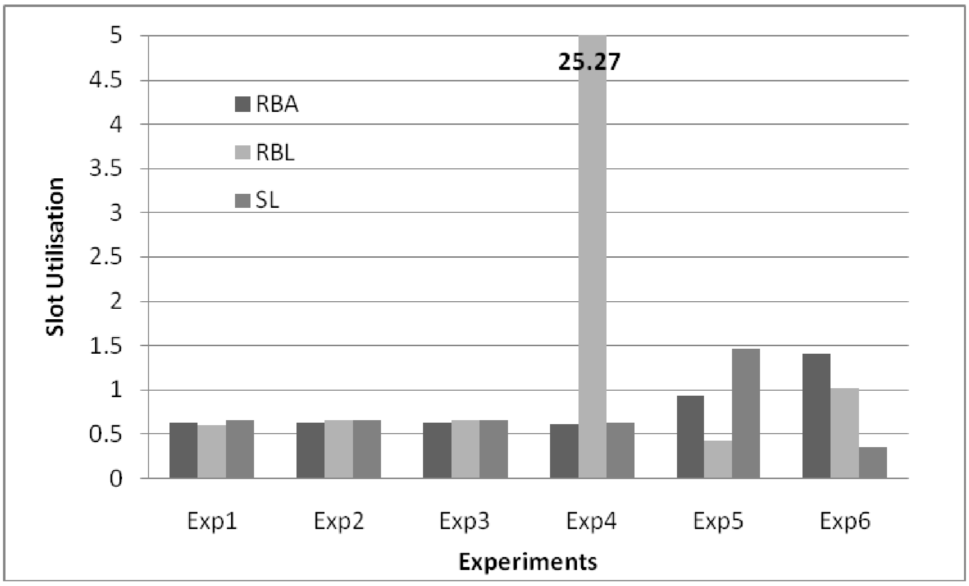

Figure 5.20. Slot utilisation comparison between Rule-based CogWSN, Rule-based CogWSN with Greedy Scoring, and Supervised CogWSN.

5.7 Conclusions

Rule-based approaches have some potential limitations since in some scenarios, the rules may be difficult to derive, and may not be able to capture the more delicate features of the decision process. One route to resolving these limitations is though

ANNs. ANNs operate based on the training of key inputs and desired output; the complete solution set of inputs and desired output is not compulsory. For the inputs that are not defined, the ANN is able to recommend an output (may not be the correct output) based on the existing trained inputs and desired outputs.

Training inputs and the desired output as in Table 5.1 are embedded into a Supervised CogWSN implementation and a range of performance evaluation experiments as defined in Table 4.4 are conducted. From Table 5.2, for Experiments 1 to 4, since the scenarios are exactly matched in terms of the trained inputs and desired output, the implementation adjusts accordingly using a smaller number of cycles to achieve the goals. For Experiment 4, the expected number of packets is not transmitted to the parent node. Therefore, more cycles are needed to transmit all packets to the parent node. For Experiments 5 and 6, the Supervised CogWSN is not trained with combination inputs. Therefore, some errors in the output are expected and a filter table to refine the decision is required (as mentioned in Section 5.5). In this case, it is noted that more cycles are required as compared to Rule-based CogWSN and Rule-based CogWSN with Greedy Scoring.

In conclusion, utilising Supervised CogWSN as a core decision process element provides a performance close to that of a Rule-based CogWSN and Rule-based CogWSN with Greedy Scoring. At the outset, it is necessary to train the multi-layer ANN network with correct pairing of inputs and desired outputs. More extensive training will result in improved performance, but excessive training pushes the solution towards that obtained through the rule-based approach at the expense of memory and computational resources. With more computational and memory resources, a greater number of neurons can be added in the hidden layer to enhance performance.

Chapter 6: Reinforcement CogWSN

Supervised Learning requires prior training before it can be applied to non-training data. Training the network with all possible training inputs and desired outputs, however, still does not guarantee perfect solution. Therefore, additional fine tuning has to be integrated at the feedback phase. Unsupervised learning is also applicable to CogWSN. Fulfilment of correct goals can be defined as rewards, viewed as important feedback to existing knowledge. Reinforcement Learning is one of the approaches to perform unguided learning for CogWSNs [231].

The Chapter starts with an overview of reinforcement learning and the manner in which it can be deployed within the CogWSN's decision process. The detail of each phase of the decision process is discussed in the following Sections. Benchmarking algorithms are introduced in Section 6.7 and compared with the four proposed CogWSN implementations. Several experiments are conducted to evaluate the performance of the approach.

6.1 Reinforcement Learning

Reinforcement Learning [231, 232] differs from standard supervised learning in that correct input and output pairs are never presented, nor sub-optimal actions explicitly corrected. As illustrated in Figure 6.1, a reinforcement learning cycle involves the interaction of an agent and its operating environment. There are three important representations in the reinforcement learning model for the agent:

1. State: represents the factors from the operating environment being observed by an agent. The state affects the reward (or network performance) as well as action selection in the next iteration.
2. Action: represents an agent's action, which may change or affect the state (or operating environment) and reward (or network performance); so the agent learns to take optimal actions at most times.
3. Reward: Reward represents the gains (losses) achieved (incurred) by an agent for taking an action on its operating environment in the previous

time instant. In other words, it is the consequence of the previous action on the operating environment, here in the form of network performance.

The representations are modelled with states *S*, a set of actions per state *A*, and rewards *R*. Decision cycles are denoted by $t \in T = \{1,2,3,...\}$. By performing an action, an agent, $a \in A$, can move from a state, $s \in S$, to another state. Each state provides the agent with a reward, $r \in R$. The goal of the agent is to maximise its total reward over time learning the optimal action for each state.

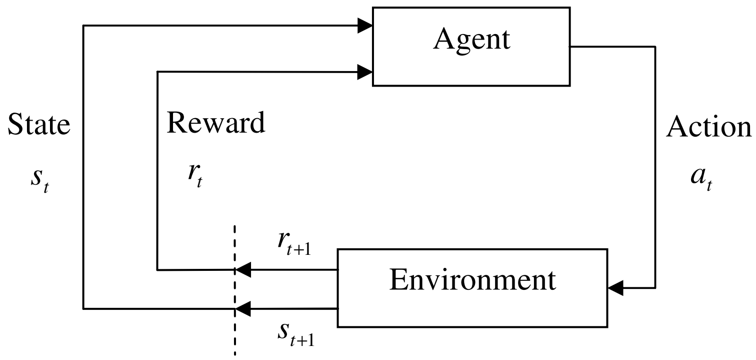

Figure 6.1. Reinforcement Learning cycle.

Q-Learning is a popular technique in Reinforcement Learning [233]. The algorithm calculates the Quality of a state-action combination, defined as in Equation 6.1;

$$Q : S \times A \rightarrow \Re \qquad (6.1)$$

The core of the algorithm is the value iteration update, in which the Q-value for a state-action pair is updated as in Equation 6.2;

$$Q_{t+1}(s_t, a_t) = (1-\alpha)Q_t(s_t, a_t) + \alpha[r_{t+1}(s_{t+1}) + \gamma \max_{a \in A} Q(s_{t+1}, a_{t+1})] \qquad (6.2)$$

where α is the learning rate, $0 \leq \alpha \leq 1$, γ is the discount factor, $0 \leq \gamma \leq 1$.

At decision time t, the agent observes states s_t from its operating environment. Based on s_t, the agent chooses an action a_t. Next, at decision time t+1, the state s_t changes to s_{t+1} as a consequence of the action a_t, and the agent receives delayed reward $r_{t+1}(s_{t+1})$. Subsequently, the agent updates Q-value $Q_{t+1}(s_t, a_t)$ (Equation 6.2). Since the agent is expected to take optimal actions with regard to any state in the remaining decision cycles at time t, t+1, t+2, …, the agent updates the Q-value using the maximum discounted future reward $\gamma \max_{a \in A} Q(s_{t+1}, a)$. As this procedure evolves through time, the agent receives a sequence of rewards and the Q-values converge to optimal actions. Q-learning searches for an optimal policy through maximizing the value function $V^{\pi}(s_t)$:

$$V^{\pi}(s_t) = \max_{a \in A}(Q_t(s_t, a_t)) \tag{6.3}$$

The policy (or action selection) for the agent is described as;

$$\pi(s_t) = \arg\max_{a \in A}(Q_t(s_t, a_t)) \tag{6.4}$$

6.2 Observe Phase

The Observe phase is implemented using event-based observation, introduced in Section 4.2.

6.3 Plan Phase

For this phase, triggering is recommended based on a plan derived based on all detected inputs. The inputs are mapped into the defined states. The state S can represent conditions of the sensor node and action A can represent a tuning function that can be harnessed. The action is selected based on the policy as in Equation 6.3 and Equation 6.4.

6.4 Implement Phase

For the Reinforcement CogWSN, the option of 'level up' or 'level down' a setting is chosen for the implementation since tuning with parameters requires more actions to be implemented which in turn causes an increase in the time to convergence.

6.5 Evaluate Phase

For the Evaluation Phase, the Quality of a state-action pair is updated as in Equation 6.5. $Q_{t+1}(s_t, a_t) = Q(s_t, a_t) + 1$ is added as a criterion in the evaluation to reduce the time to achievement of the goal:

$$Q_{t+1}(s_t, a_t) = \begin{cases} 100 & \text{, if goal} \\ Q(s_t, a_t) + 1 & \text{, if improvement} \\ (1-\alpha)Q_t(s_t, a_t) + \alpha[r_{t+1}(s_{t+1}) + \\ \gamma \max_{a \in A} Q(s_{t+1}, a_{t+1})] & \text{, otherwise} \end{cases} \tag{6.5}$$

where $0 \le Q_{t+1}(s_t, a_t) \le 100$. Without this equation, the learning process takes a relatively long time to achieve the goal and to formulate the necessary knowledge for the next search towards the goal.

6.6 Verification

A case study is performed to evaluate the performance of the Reinforcement CogWSN. The objective of the case study is the same as specified in Section 4.6, driving the CogWSN to achieve optimum performance in terms of connectivity and slot utilisation. The case study focuses on three metrics; power transmission, slot allocation to parent, and slot allocation to child node. The targets to achieve, monitored conditions, and derived actions for each domain are shown in Table 4.2.

The experimental setup is configured as described in Section 4.6. A reinforcement learning model with 27 states (Table 6.1) with 6 actions (Table 6.2) in each state is embedded into all sensor nodes. These states are formed from 6 states with 1 non-goal status, 12 states with 2 non-goal status, 8 states with 3 non-goal status and 1 goal. For the actions, there are 3 functions to adjust and in each function, the adjustment will be increased or decreased; therefore, in total there are 6 actions. It should be noted that State 0 is the goal.

To determine the learning rate, discount factor, and reward value for a particular scenario, a simulation was conducted for a reinforcement learning model with 36 states and 6 actions for each state. The simulation was run for 100000 times, changing each parameter. For each time, the starting state and ending state were randomly selected. As shown in Figure 6.2, the learning rate should be set between 0.1 to 0.5 in order to achieve the minimum number of searches to finding the goal. In Figure 6.3, the best discount factor should be set to 0.5 to reach the goal using the minimum number of searches. From Figure 6.4, the reward value can be selected from -1 to 10 to maintain a low number of searches to finding the goal. Therefore, for the case studies, the learning rate α and discount factor γ were set to 0.5 and 0.7 respectively as an average. The average learning rate is the preferred metric since a low learning rate will result in more time to determine the correct action while a high learning rate will result in oscillation or divergence of the right solution. The discount factor is able to determine the current state against the goal state; an average higher value is preferable as it indicates a closer representation of the state towards the goal.

Table 6.1. List of states of the reinforcement learning model used in CogWSN.

States	Description of the State
0	$-85dBm \leq RSSI \leq -75dBm$; $\frac{1}{2}s_1 \leq x \leq s_1$; $\frac{1}{2}s_2 \leq y \leq s_2$
1	$RSSI < -85dBm$; $\frac{1}{2}s_1 \leq x \leq s_1$; $\frac{1}{2}s_2 \leq y \leq s_2$
2	$RSSI > -75dBm$; $\frac{1}{2}s_1 \leq x \leq s_1$; $\frac{1}{2}s_2 \leq y \leq s_2$
3	$x > s_1$; $-85dBm \leq RSSI \leq -75dBm$; $\frac{1}{2}s_2 \leq y \leq s_2$
4	$x > s_1$; $RSSI < -85dBm$; $\frac{1}{2}s_2 \leq y \leq s_2$
5	$x > s_1$; $RSSI > -75dBm$; $\frac{1}{2}s_2 \leq y \leq s_2$
6	$x < \frac{1}{2}s_1$; $-85dBm \leq RSSI \leq -75dBm$; $\frac{1}{2}s_2 \leq y \leq s_2$
7	$x < \frac{1}{2}s_1$; $RSSI < -85dBm$; $\frac{1}{2}s_2 \leq y \leq s_2$
8	$x < \frac{1}{2}s_1$; $RSSI > -75dBm$; $\frac{1}{2}s_2 \leq y \leq s_2$
9	$y > s_2$; $-85dBm \leq RSSI \leq -75dBm$; $\frac{1}{2}s_1 \leq x \leq s_1$
10	$y > s_2$; $RSSI < -85dBm$; $\frac{1}{2}s_1 \leq x \leq s_1$
11	$y > s_2$; $RSSI > -75dBm$; $\frac{1}{2}s_1 \leq x \leq s_1$
12	$y > s_2$; $x > s_1$; $-85dBm \leq RSSI \leq -75dBm$
13	$y > s_2$; $x > s_1$; $RSSI < -85dBm$
14	$y > s_2$; $x > s_1$; $RSSI > -75dBm$
15	$y > s_2$; $x < \frac{1}{2}s_1$; $-85dBm \leq RSSI \leq -75dBm$
16	$y > s_2$; $x < \frac{1}{2}s_1$; $RSSI < -85dBm$
17	$y > s_2$; $x < \frac{1}{2}s_1$; $RSSI > -75dBm$
18	$y < \frac{1}{2}s_2$; $-85dBm \leq RSSI \leq -75dBm$; $\frac{1}{2}s_1 \leq x \leq s_1$;
19	$y < \frac{1}{2}s_2$; $RSSI < -85dBm$; $\frac{1}{2}s_1 \leq x \leq s_1$
20	$y < \frac{1}{2}s_2$; $RSSI > -75dBm$; $\frac{1}{2}s_1 \leq x \leq s_1$
21	$y < \frac{1}{2}s_2$; $x > s_1$; $-85dBm \leq RSSI \leq -75dBm$
22	$y < \frac{1}{2}s_2$; $x > s_1$; $RSSI < -85dBm$
23	$y < \frac{1}{2}s_2$; $x > s_1$; $RSSI > -75dBm$
24	$y < \frac{1}{2}s_2$; $x < \frac{1}{2}s_1$; $-85dBm \leq RSSI \leq -75dBm$
25	$y < \frac{1}{2}s_2$; $x < \frac{1}{2}s_1$; $RSSI < -85dBm$
26	$y < \frac{1}{2}s_2$; $x < \frac{1}{2}s_1$; $RSSI > -75dBm$

Table 6.2. List of actions of reinforcement learning model used in CogWSN.

Actions	Description of the Action
1	Increase transmit power by 1 level up
2	Decrease transmit power by 1 level down
3	Increase slot allocation by $2s_1$
4	Reduce slot allocation by $s_1/2$
5	Increase slot allocation by $2s_2$
6	Reduce slot allocation by $s_2/2$

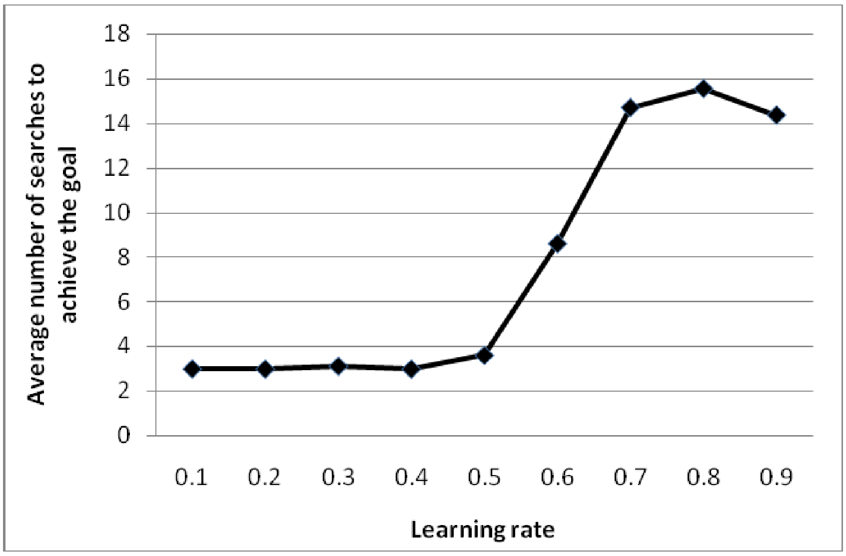

Figure 6.2. Average number of searches needed to achieve the goal with different learning rates, discount factor equals 0.5, and reward equals -1 for 36 states.

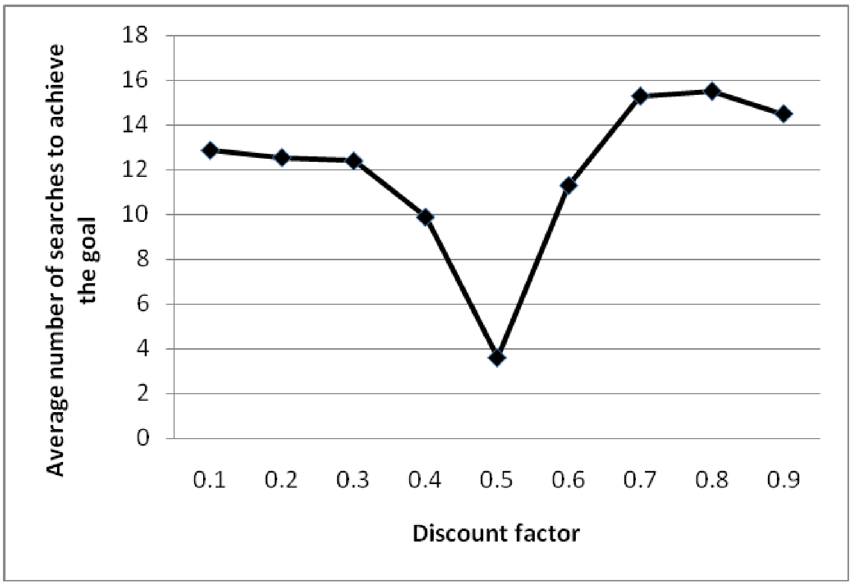

Figure 6.3. Average number of searches needed to achieve the goal with different discount factors, learning rate equals 0.5, and reward equals -1 for 36 states.

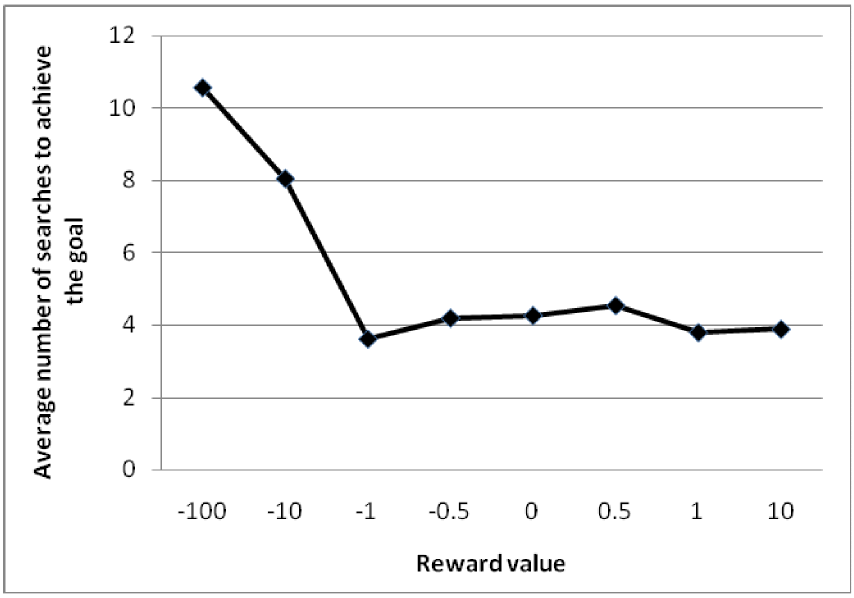

Figure 6.4. Average number of searches needed to achieve the goal with different reward values, learning rate equals 0.5, and discount factor equals 0.5 for 36 states.

Experiments are setup as summarised in Table 4.4.

In Experiment 1, a node is placed in close proximity to the base station. At the outset, the transmission power is set to the maximum, 0dBm resulting in a received power >-75dBm. After 14 cycles, the received power is adjusted and successfully maintained between -85dBm to -75dBm (Figure 6.5). Before that goal is achieved, there are several attempts to adjust the slot allocation (Figure 6.6), a consequence of the time taken to reach the goal. After the goal is achieved, no additional adjustment is carried out for slot allocation since connectivity is maintained and slot utilisation remains almost constant.

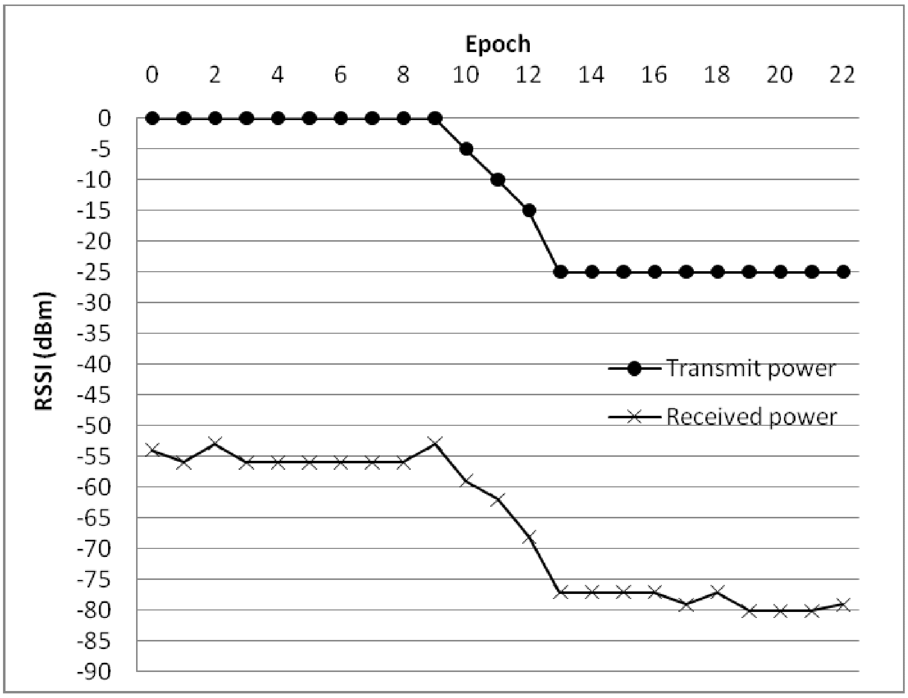

Figure 6.5. RSSI as a function of the number of cycles for Reinforcement CogWSN in Experiment 1.

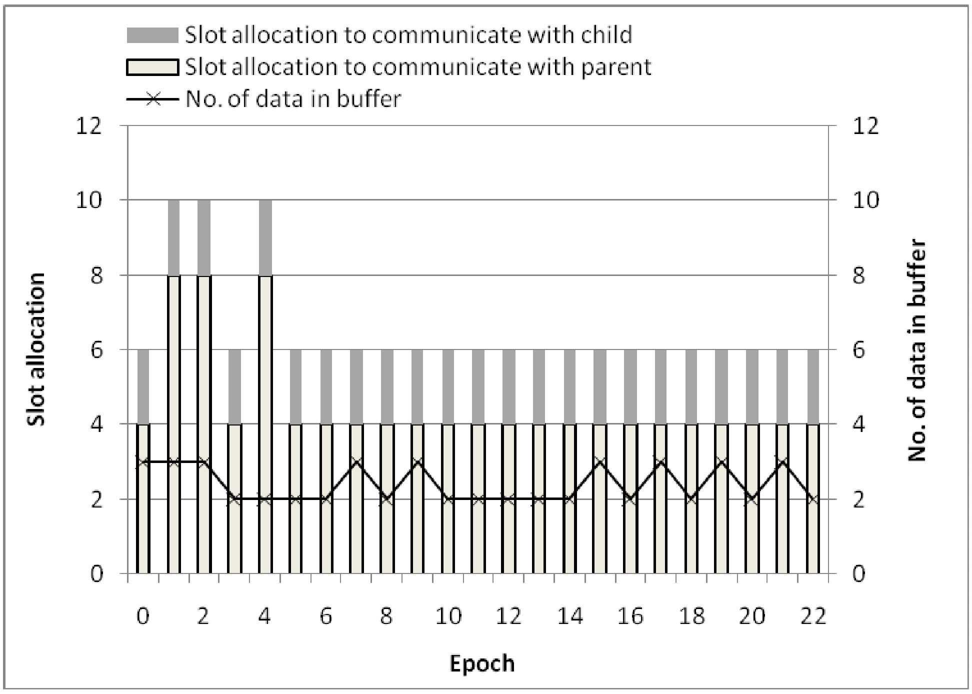

Figure 6.6. Slot allocation and buffer condition as a function of the number of cycles for Reinforcement CogWSN in Experiment 1.

In Experiment 2, the transmission power is set to the minimum -25dBm and sensor data is sampled at 0.5Hz. At the outset, a node is placed in close proximity to a base station until the received power stabilises. Thereafter, the node is moved away from the base station but still within communication range. When the node no longer receives an acknowledgement packet, it increases its power towards the maximum until the node receives feedback that it is using too excessive a transmission power. After 5 cycles, the received power is successfully managed to be between -85dBm to -75dBm (Figure 6.7). Before the goal is achieved, several attempts are made to adjust the slot allocation (Figure 6.8) as a consequence of the time taken to reach the goal. On achieving the goal, no additional adjustment is carried out for slot allocation since connectivity is maintained and slot utilisation remains constant.

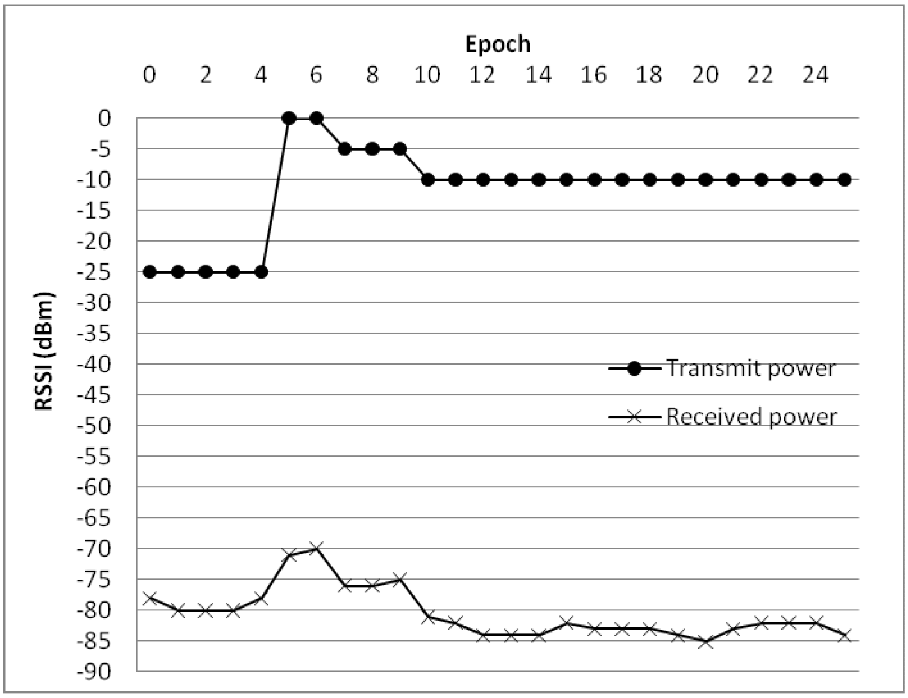

Figure 6.7. RSSI as a function of the number of cycles for Reinforcement CogWSN in Experiment 2.

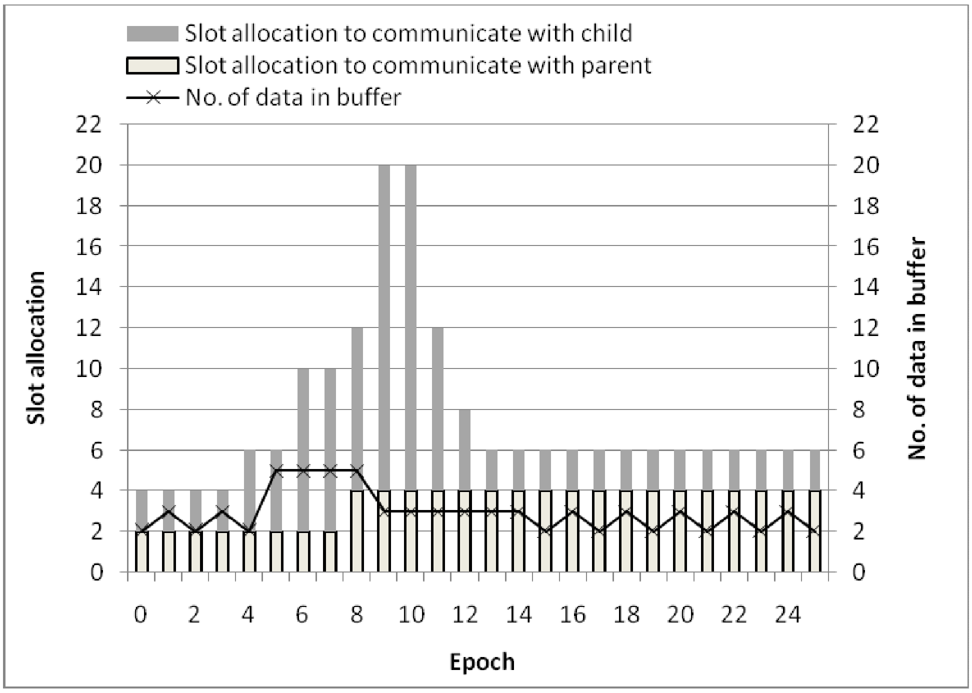

Figure 6.8. Slot allocation and buffer condition as a function of the number of cycles for Reinforcement CogWSN in Experiment 2.

In Experiment 3, a node is placed in close proximity to a base station, the slot allocation to communicate with parent node is set to 64 and sensor data is sampled at 0.5Hz. The transmission power is set to -25dBm. No adjustment is carried out to the transmission power since the RSSI remains between -85dBm to -75dBm (Figure 6.9). The allocation of 64 slots to communicate with the parent is excessive; allocated slots are unused and thus the number of slots can be reduced. After 6 cycles, the slot allocation to communicate with parent node is reduced to s_1 equals 4, satisfying the slot utilisation of between $\frac{1}{2}s_1$ to s_1 (Figure 6.10).

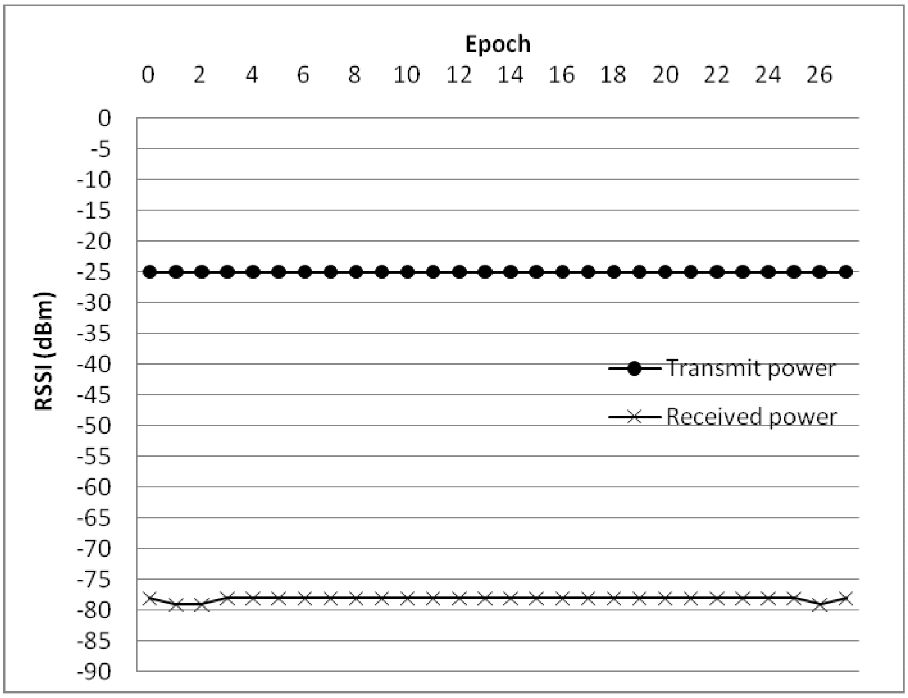

Figure 6.9. RSSI as a function of the number of cycles for Reinforcement CogWSN in Experiment 3.

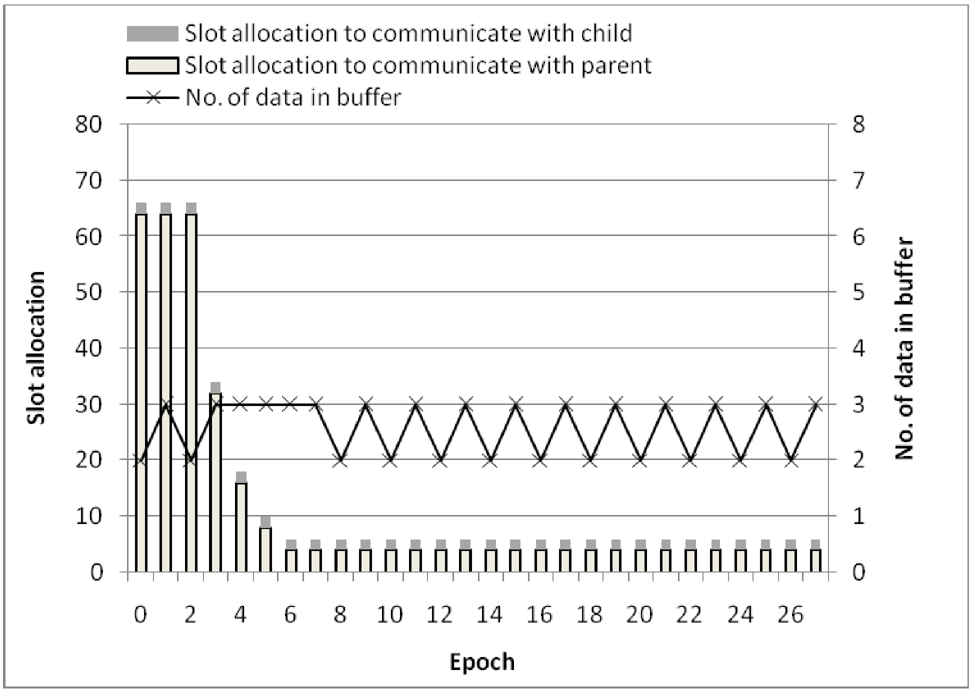

Figure 6.10. Slot allocation and buffer condition as a function of the number of cycles for Reinforcement CogWSN in Experiment 3.

In Experiment 4, a node is placed in close proximity to a base station, the slot allocation to communicate with the parent node is set to 2, as is the slot allocation to communicate with the child node, and sensor data is sampled at 5Hz. The transmission power is set to -25dBm. No adjustment is required for the transmission power since the RSSI remains between -85dBm to -75dBm (Figure 6.11). Too few slots are allocated for the transmission to the parent node and consequently the allocation has to be increased. After 23 cycles, the slot allocation to communicate with parent node is increased to s_1 equals 32, satisfying the slot utilisation of between $\frac{1}{2}s_1$ to s_1 (Figure 6.12).

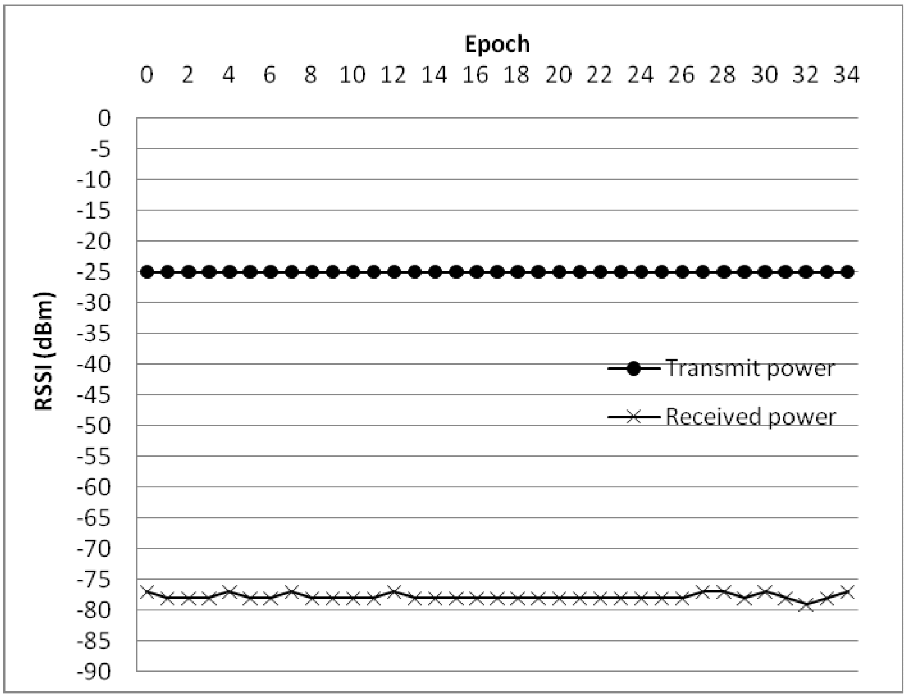

Figure 6.11. RSSI as a function of the number of cycles for Reinforcement CogWSN in Experiment 4.

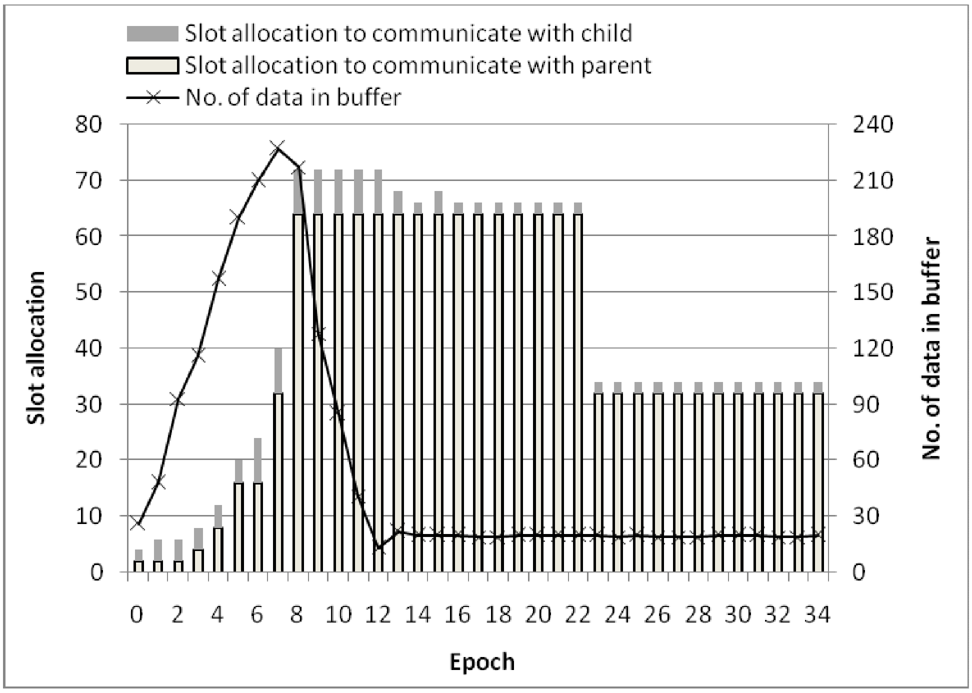

Figure 6.12. Slot allocation and buffer condition as a function of the number of cycles for Reinforcement CogWSN in Experiment 4.

In Experiment 5, a node is placed in close proximity to a base station. The slot allocation to communicate with parent and child nodes is set to 2 and 64 respectively. Another child node is placed near to this node, effectively becoming a parent node. Both nodes are at a sampling rate of 0.5Hz. No active adjustment is required for the transmission power (Figure 6.13) as the task concerns the allocation of slots. As shown in Figure 6.14, after 23 cycles, both slot allocations to communicate with parent and child nodes remain between 4 to 8.

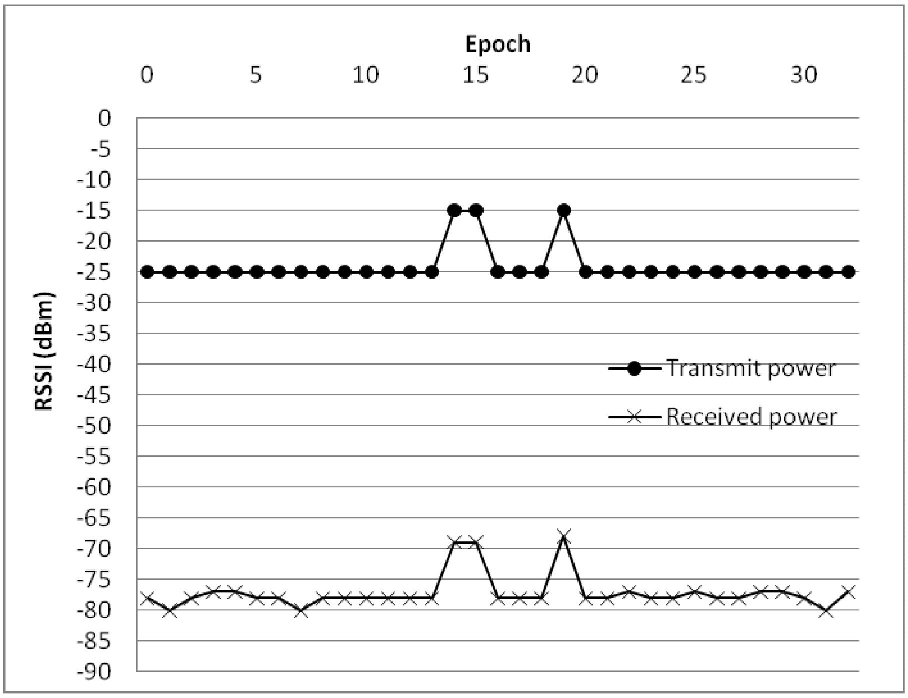

Figure 6.13. RSSI as a function of the number of cycles for Reinforcement CogWSN in Experiment 5.

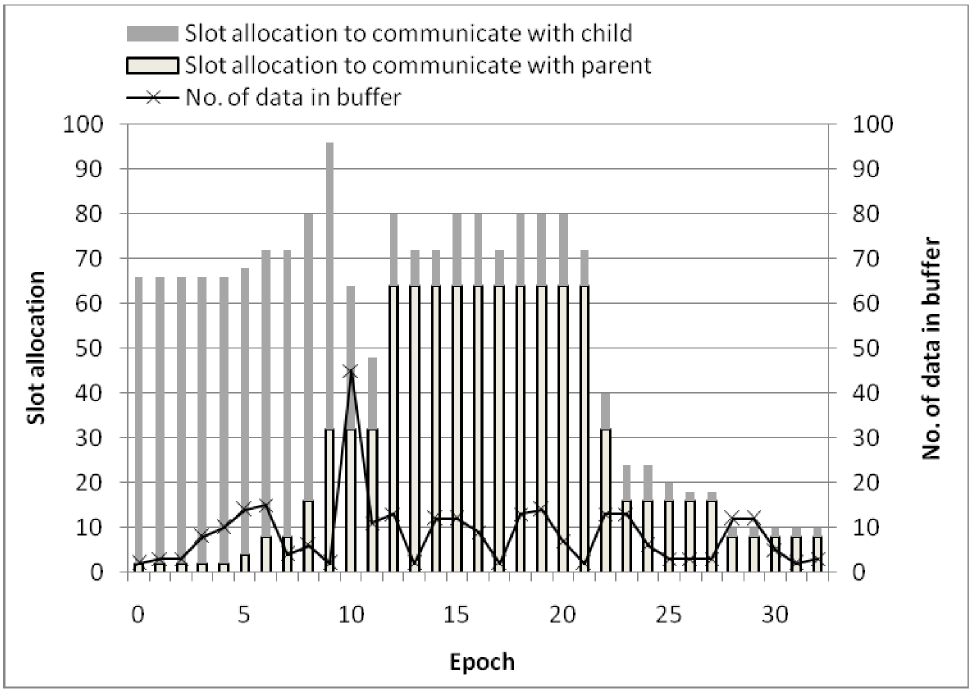

Figure 6.14. Slot allocation and buffer condition as a function of the number of cycles for Reinforcement CogWSN in Experiment 5.

In Experiment 6, a node is placed in close proximity to a base station. The slot allocation to communicate with both parent and child nodes is set to 2. Another child node is placed near to this node, effectively becoming a parent node. Both nodes are at a sampling rate of 0.5Hz. As shown in Figure 6.15, the adjustment occurs from the 17^{th} to the 33^{th} cycle, after which the RSSI remains between -85dBm to -75dBm. As shown in Figure 6.16, the goal in terms of slot allocation is achieved in 25 cycles. From 25 cycles onwards, the buffer occupancy fluctuates since it is populated quickly and randomly by data from the child node.

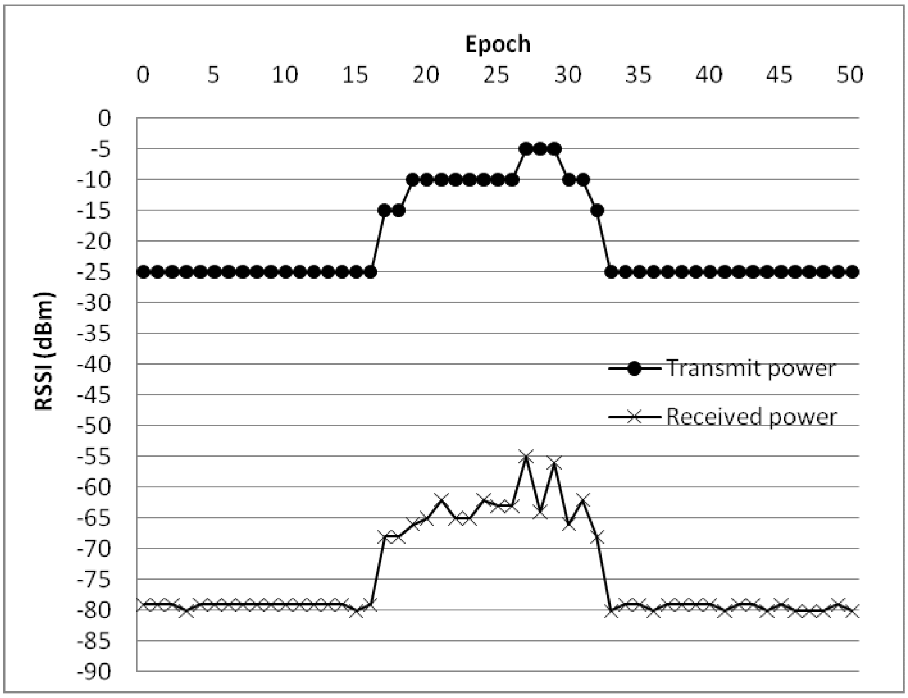

Figure 6.15. RSSI as a function of the number of cycles for Reinforcement CogWSN in Experiment 6.

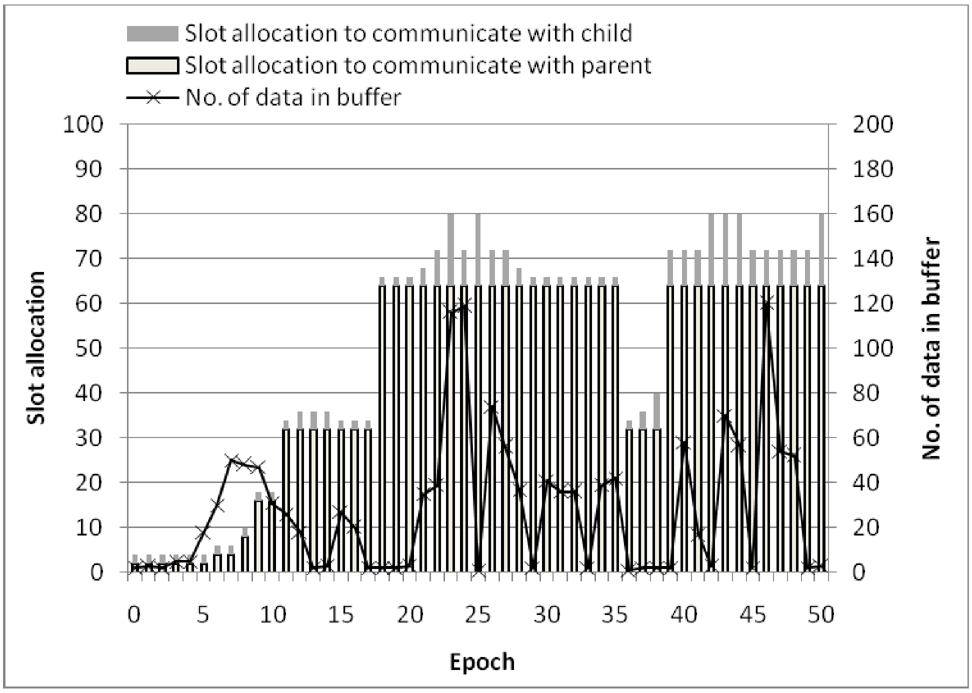

Figure 6.16. Slot allocation and buffer condition as a function of the number of cycles for Reinforcement CogWSN in Experiment 6.

6.7 Comparison and Discussion

In order to benchmark the proposed solutions, a combination of research from [12] (through experiment) which concentrates on transmission power control and [234] (through simulation) which focuses on slot allocations is used. The benchmarking algorithms from the combined work (to solve the transmission power control first and then slot allocation) are shown as in Figure 6.17. α, T_L, and T_H are set to 0.8, -85dBm, and -75dBm respectively. The experimental setup and goals remain the same.

```
Require: R {RSSI from the current sample}
Require: R̄ {Average RSSI}
Require: BUF {Buffer size}
Require: OBS {Occupied buffer space}
Require: PPS {Packet per slot}
Require: TSA {Total slot allocation}
R̄ ← αR + (1 − α)R̄
if R̄ < T_L then
  Increase transmit power by 2 levels
else if R̄ > T_H then
  Decrease transmit power by 1 level
else
  No action is required
end if
TSA ← (BUF − OBS) / PPS
Perform TSA for slot allocation to communicate with
parent node
Perform 1/3 × TSA for slot allocation to communicate
with child node
```

Figure 6.17. The algorithms from a combination of reported research for benchmarking.

In Experiment 1, a node is placed in close proximity to the base station, the transmission power is set to the maximum, 0dBm and sensor data is sampled at 0.5Hz. For the benchmarking algorithms, as shown in Figure 6.18, those parameters result in a received power -75dBm from the outset. After 4 cycles, the received power is adjusted and successfully maintained between -85dBm to -75dBm. As shown in Figure 6.19, the slot allocation to communicate with the parent node is adjusted to lie between 40 to 65. However a significant number of slots allocated (at least 90%) is not being utilised. The algorithms allocate too many of slots if the node is equipped with large buffer size (*BUF)*, low occupied buffer space (*OBS)*, and low packet per slot (*PPS)*.

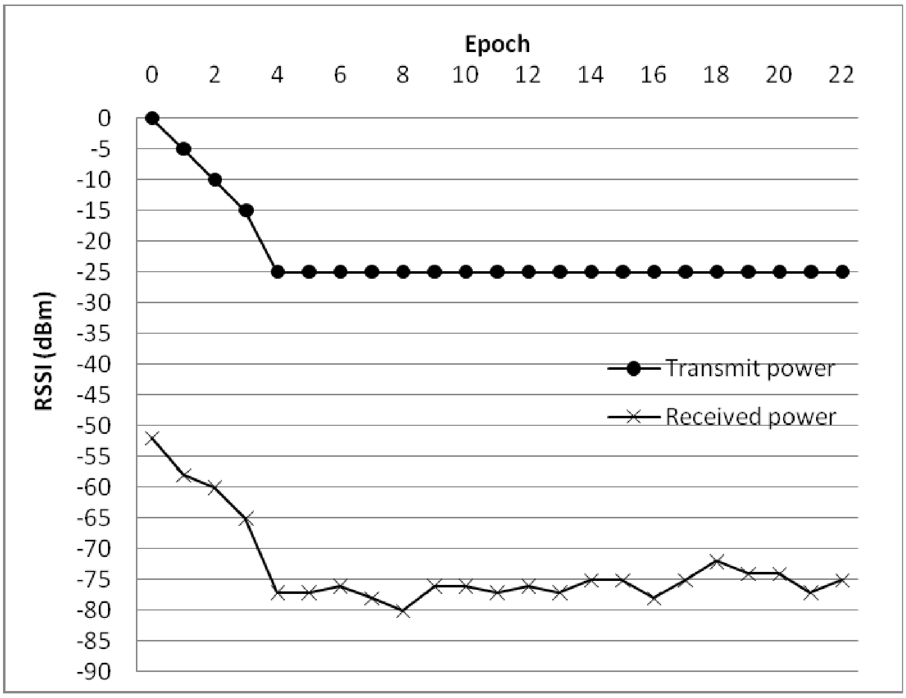

Figure 6.18. RSSI as a function of the number of cycles for benchmarking algorithms in Experiment 1.

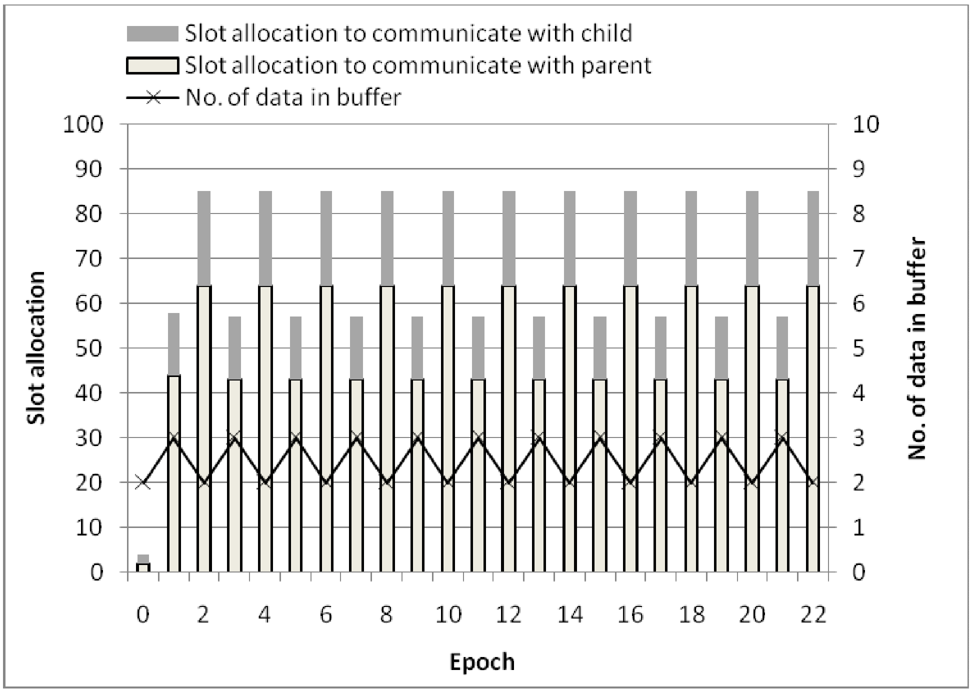

Figure 6.19. Slot allocation and buffer condition as a function of the number of cycles for benchmarking algorithms in Experiment 1.

In Experiment 2, the transmission power is set to the minimum -25dBm and sensor data is sampled at 0.5Hz. At the outset, a node is placed in close proximity to a base station until the received power stabilises. Thereafter, the node is moved gradually away from the base station but still remains within communication range. When the node no longer is in receipt of an acknowledgement, the transmission power is incremented towards the maximum. On receipt of feedback indicating excess use of transmission power, after 3 cycles, the received power is successfully managed (Figure 6.20). As shown in Figure 6.21, the slot allocation to communicate with the parent node is adjusted to lie between 40 to 65; again, a significant number of slots allocated (at least 90%) are not being utilised.

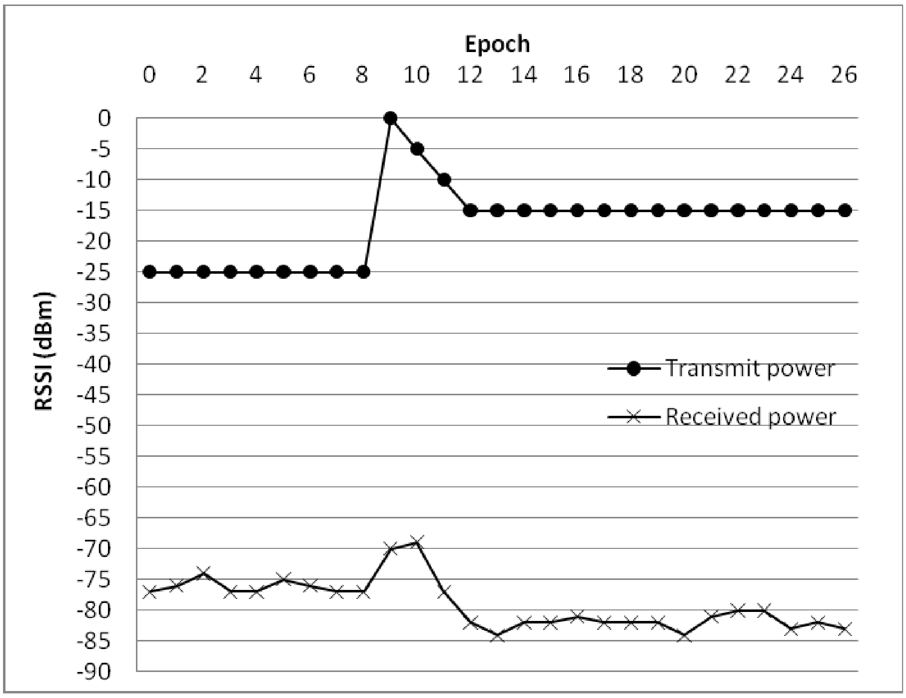

Figure 6.20. RSSI as a function of the number of cycles for benchmarking algorithms in Experiment 2.

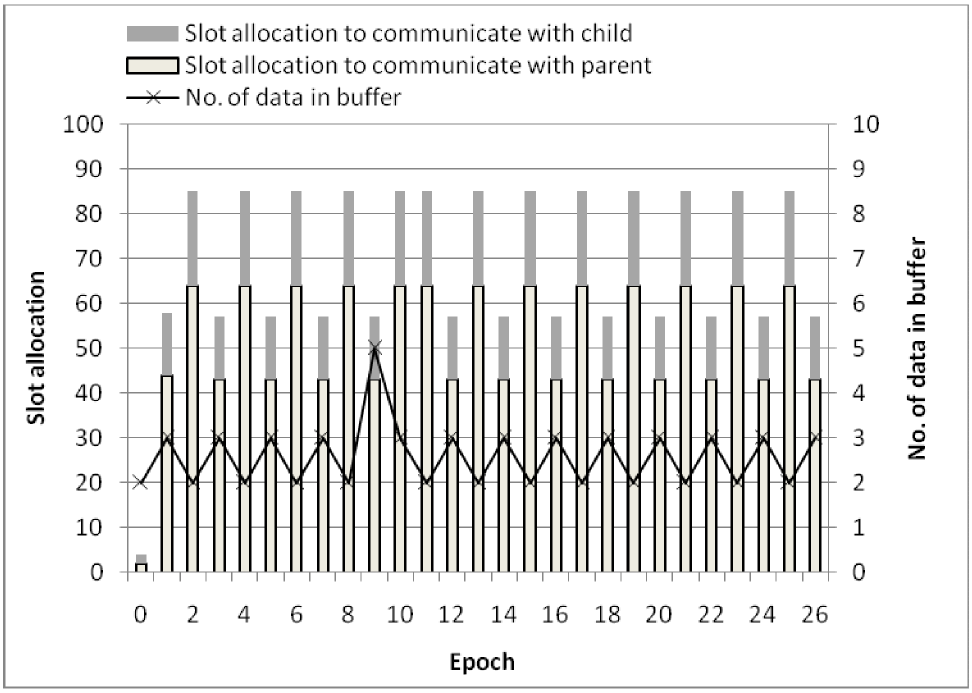

Figure 6.21. Slot allocation and buffer condition as a function of the number of cycles for benchmarking algorithms in Experiment 2.

In Experiment 3, a node is placed in close proximity to a base station, the slot allocation to communicate with parent node and child node is set to 64 and 2 respectively. Sensor data is sampled at 0.5Hz and the transmission power is set to -25dBm. No adjustment is required for the transmission power since the RSSI is maintained between -85dBm to -75dBm (Figure 6.22). As shown in Figure 6.23, the low utilization of the slot allocation remains an issue; the slot allocation to communicate with the parent node is adjusted to lie between 40 to 65.

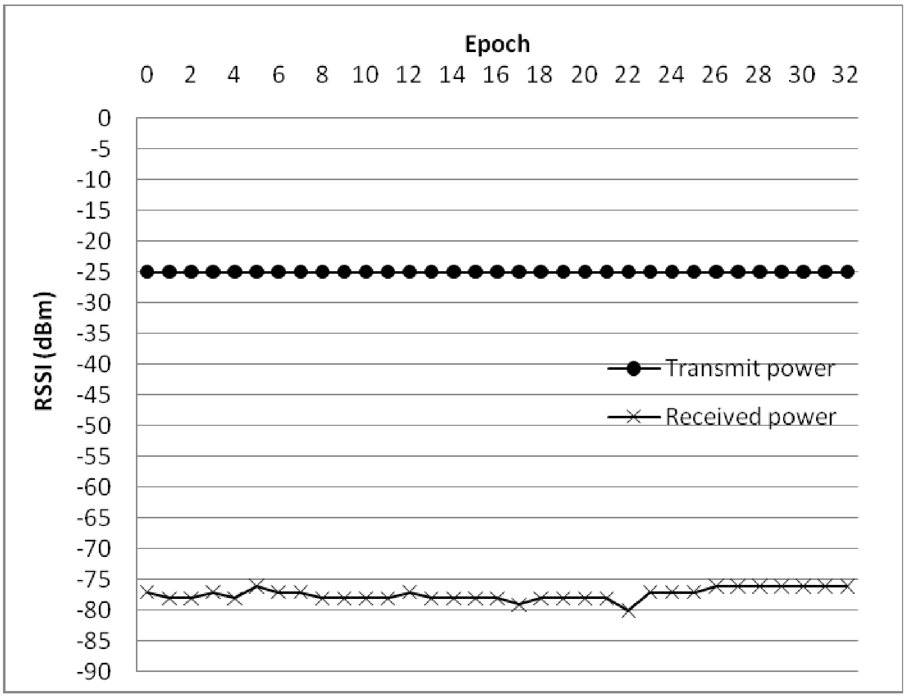

Figure 6.22. RSSI as a function of the number of cycles for benchmarking algorithms in Experiment 3.

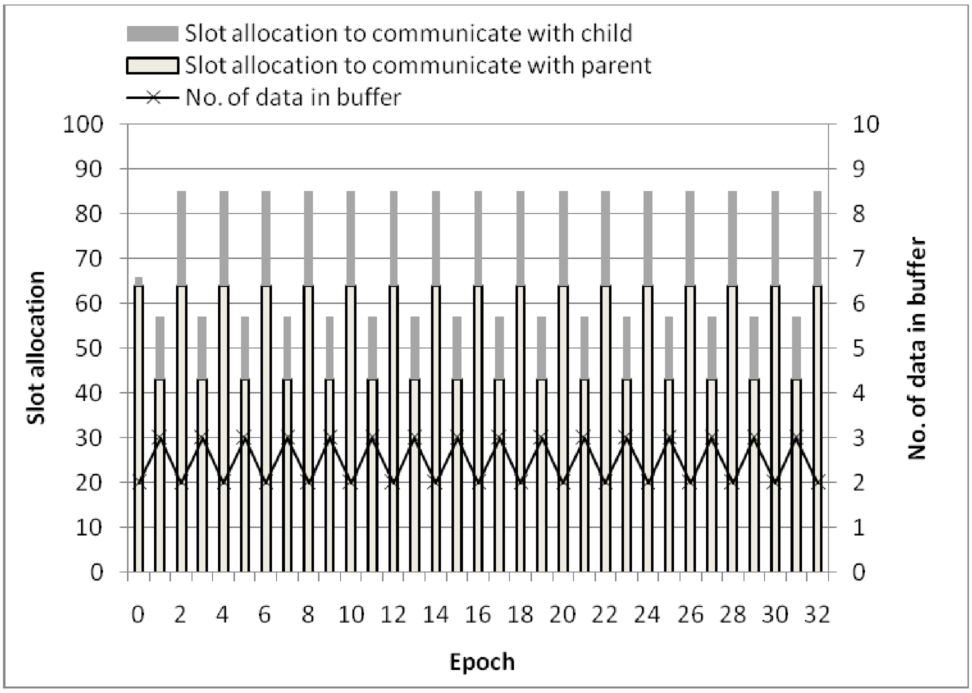

Figure 6.23. Slot allocation and buffer condition as a function of the number of cycles for benchmarking algorithms in Experiment 3.

In Experiment 4, a node is placed in close proximity to a base station, the slot allocation to communicate with the parent node is set to 2 as is the slot allocation to communicate with the child node, and sensor data is sampled at 5Hz. The transmission power is set to -25dBm. For the benchmarking algorithms, no adjustment is required for the transmission power since the RSSI is maintained between -85dBm to -75dBm (Figure 6.24). The slot allocation to communicate with the parent node is successfully increased and maintained between 15 to 30 (Figure 6.25); however the slot allocation to communicate with the child node has been over allocated.

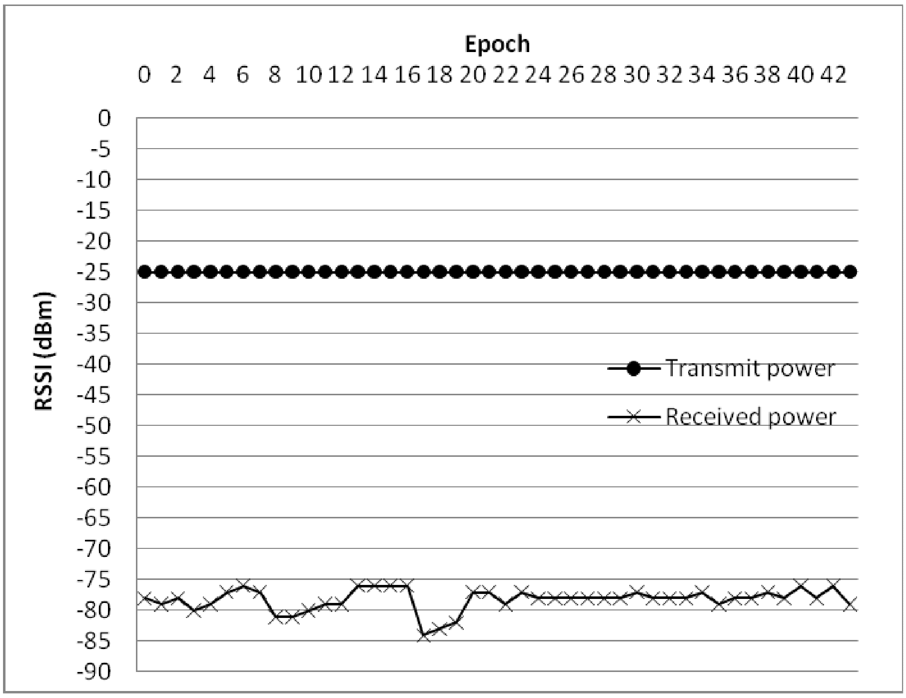

Figure 6.24. RSSI as a function of the number of cycles for benchmarking algorithms in Experiment 4.

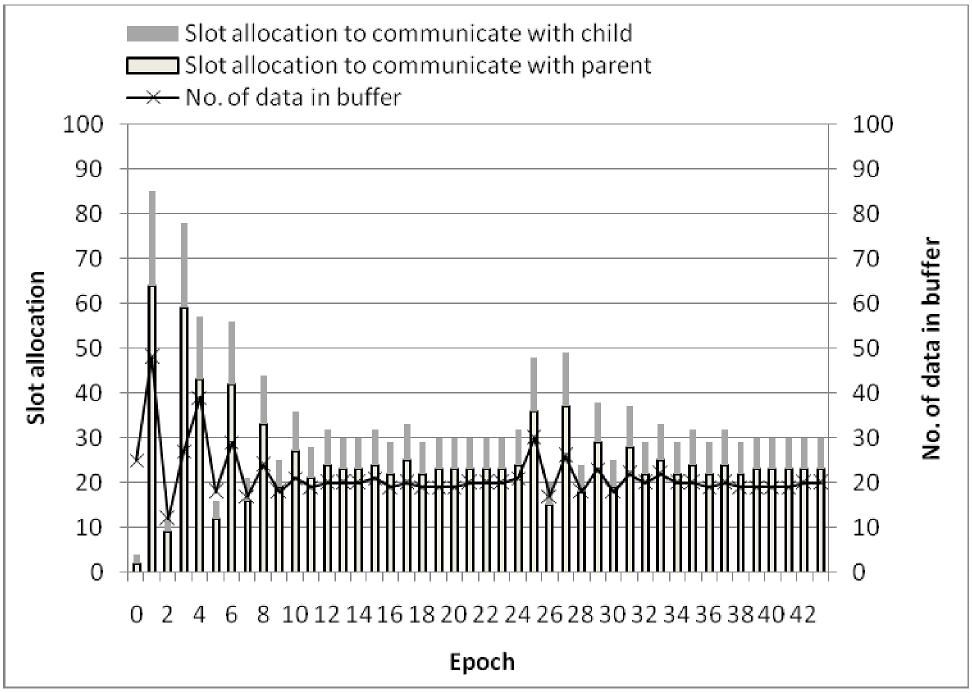

Figure 6.25. Slot allocation and buffer condition as a function of the number of cycles for benchmarking algorithms in Experiment 4.

In Experiment 5, a node is placed in close proximity to a base station. The slot allocation to communicate with the parent and child nodes is set to 2 and 64 respectively. Another child node is placed near to this node, becoming its parent node. Both nodes operate at a sampling rate of 0.5Hz. No major adjustment is required for the transmission power (Figure 6.26). As shown in Figure 6.27, although the buffer is cleared most of the time, the slot allocation to communicate with the parent node is nevertheless over-estimated.

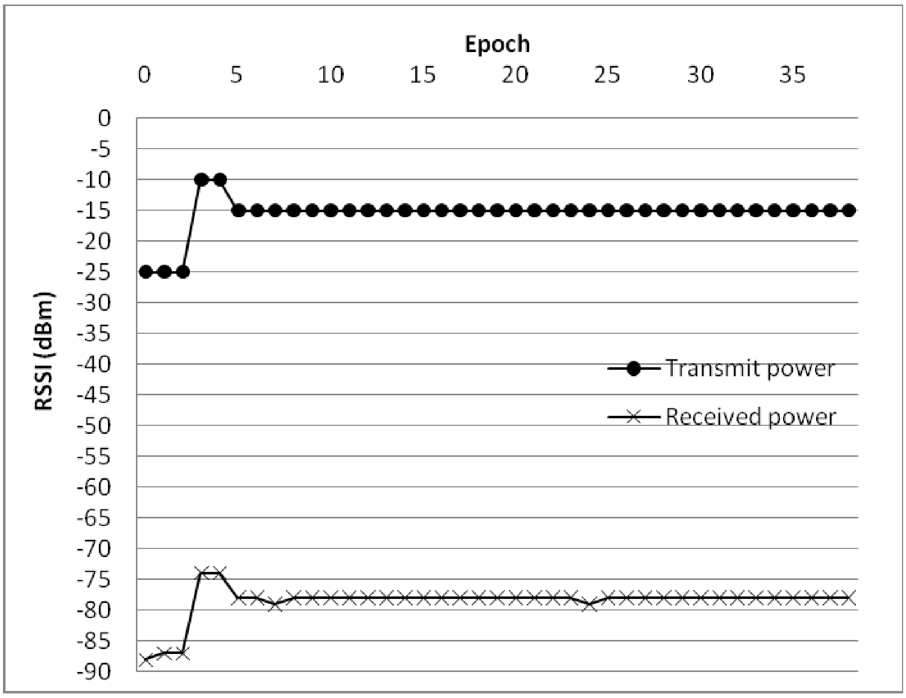

Figure 6.26. RSSI as a function of the number of cycles for benchmarking algorithms in Experiment 5.

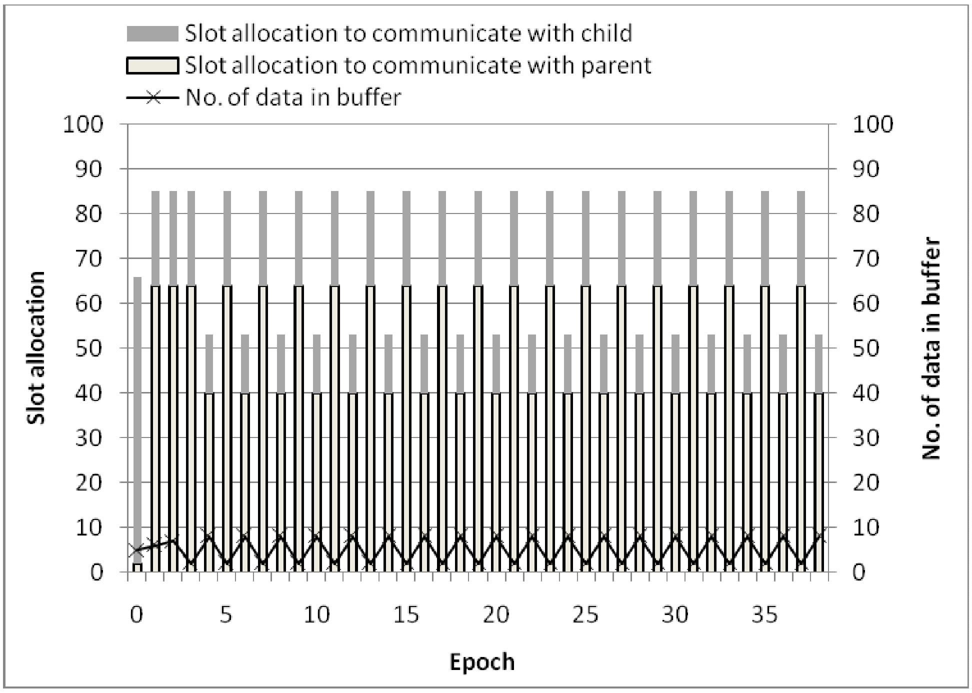

Figure 6.27. Slot allocation and buffer condition as a function of the number of cycles for benchmarking algorithms in Experiment 5.

In Experiment 6, a node is placed in close proximity to a base station, and the slot allocation to communicate with both the parent and child nodes is set to 2. Another child node is placed near to this node, becoming its parent node. Both nodes operate at a sampling rate at 0.5Hz. No major adjustment is required for the transmission power (Figure 6.28). As shown in Figure 6.29, the slot allocation to communicate with parent node is over estimated.

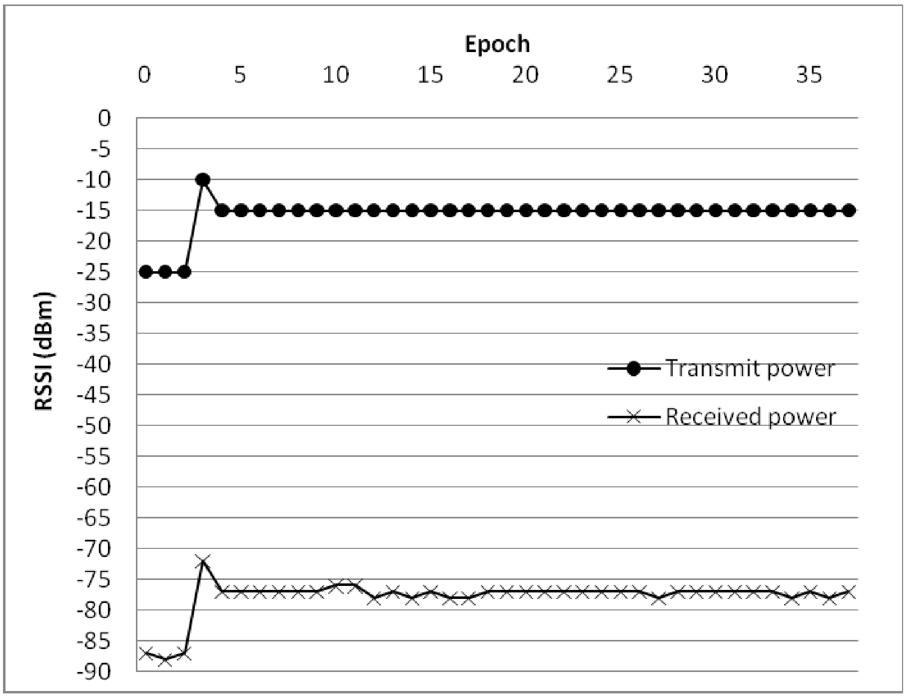

Figure 6.28. RSSI as a function of the number of cycles for benchmarking algorithms in Experiment 6.

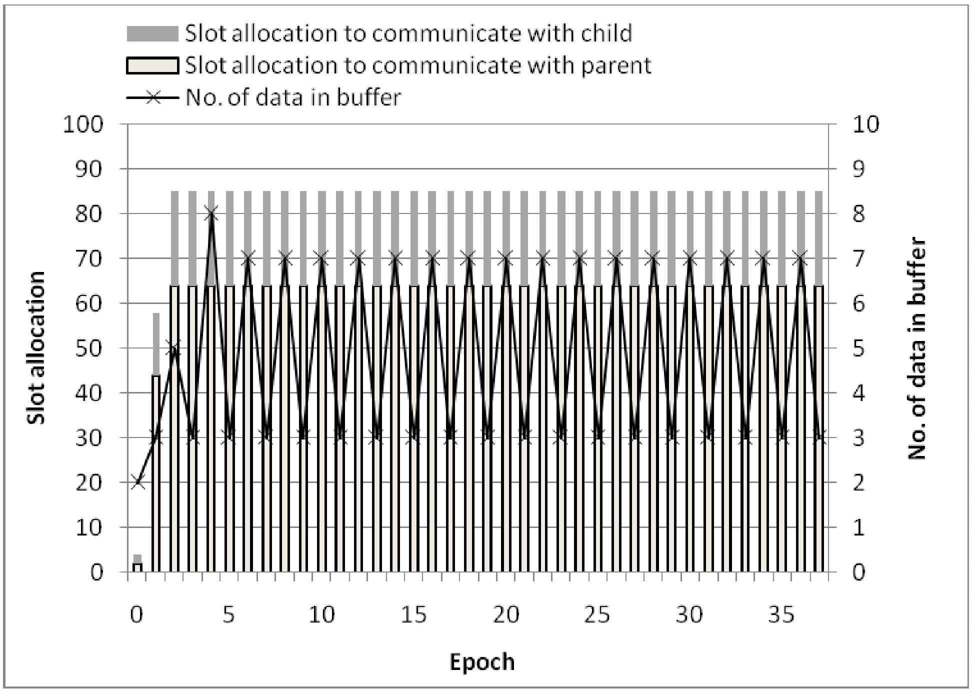

Figure 6.29. Slot allocation and buffer condition as a function of the number of cycles for benchmarking algorithms in Experiment 6.

Table 6.3 and Figure 6.30 show a comparison of the rate of adjustment for Rule-based CogWSN (RBA); Rule-based CogWSN with Greedy Scoring (RBL); Supervised CogWSN (SL); Reinforcement CogWSN (RL); and the benchmarking algorithms (OT) for Experiments 1 to 6 (Exp1 to Exp 6). Supervised CogWSN is able to achieve a performance close to Rule-based CogWSN except for Experiment 5. Supervised CogWSN is not trained with multiple inputs: too few slots to the parent and too many slots to the child are allocated. Under this scenario, more cycles are required in order to provide the correct output. Benchmarking algorithms outperform all others. As expected, Reinforcement CogWSN exhibits the worst performance; the approach requires more cycles in order to achieve the targeted goals since no a priori knowledge is embedded and a number of trials (cycles) are needed to acquire the requisite learning.

Table 6.3. Number of cycles to achieve the targets for Rule-based CogWSN, Rule-based CogWSN with Greedy Scoring, Supervised CogWSN, Reinforcement CogWSN, and benchmarking algorithms.

Experiments and initial setup	**No. of cycles to achieve the goal for Rule-based CogWSN**	**No. of cycles to achieve the goal for Rule-based CogWSN with Greedy Scoring**	**No. of cycles to achieve the goal for Supervised CogWSN**	**No. of cycles to achieve the goal for Reinforcement CogWSN**	**No. of cycles to achieve the goal for bench-marking algorithms**
Experiment 1	6	6	5	13	4
Experiment 2	3	4	2	8	3
Experiment 3	4	4	4	6	1
Experiment 4	9	8	13	23	9
Experiment 5	8	13	23	23	5
Experiment 6	6	9	10	39	4

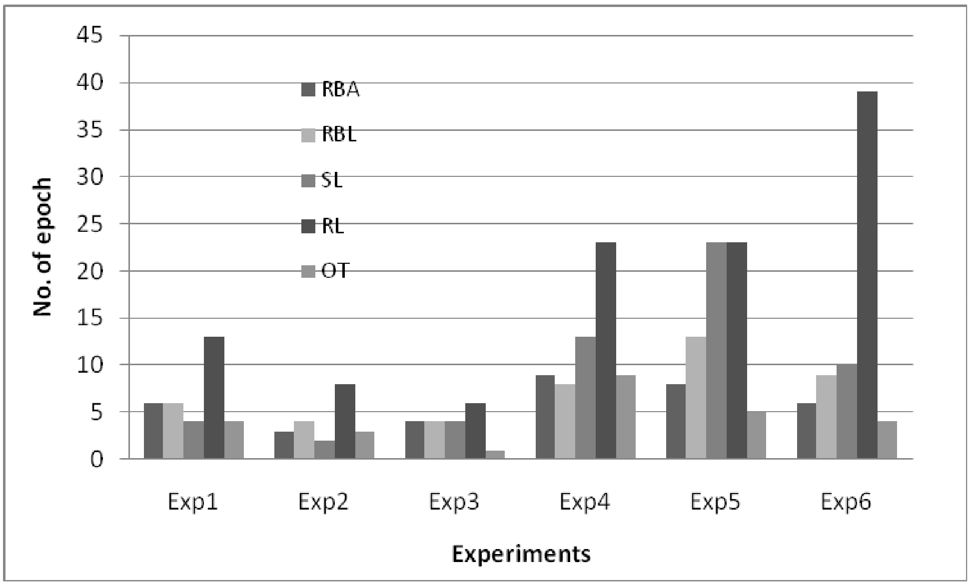

Figure 6.30. A comparison of the speed of adjustment between Rule-based CogWSN, Rule-based CogWSN with Greedy Scoring, Supervised CogWSN, Reinforcement CogWSN, and benchmarking algorithms.

Figure 6.31 shows a comparison of the transmission power optimisation for RBA, RBL, SL, RL, and OT. In Exp 2, OT is able to tune its power transmission even lower; however, in Exp 5 and Exp 6, its power transmission adjustment is slightly over-estimated. Overall, all methods exhibit nominally the same performance.

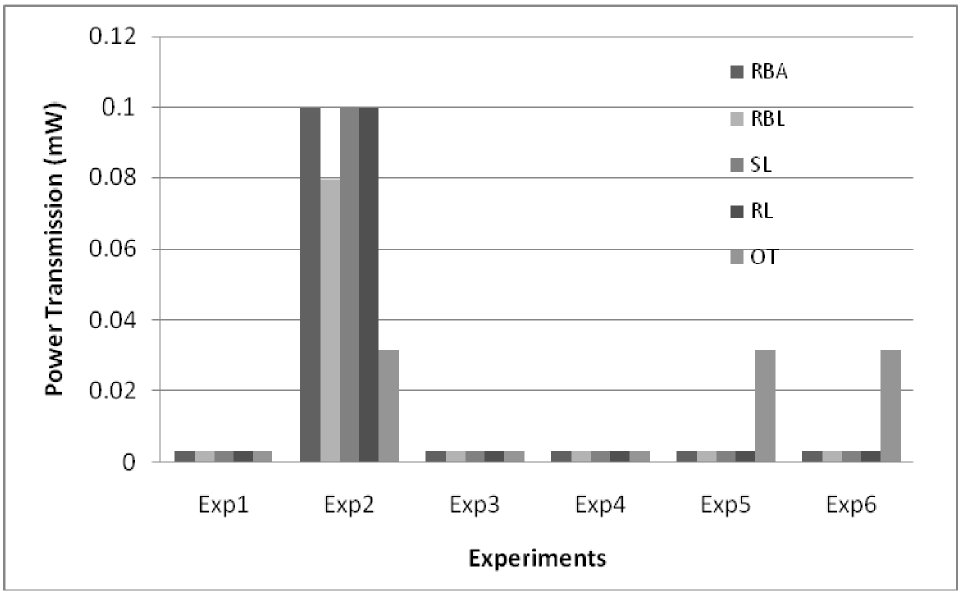

Figure 6.31. Transmission power comparison between Rule-based CogWSN, Rule-based CogWSN with Greedy Scoring, Supervised CogWSN, Reinforcement CogWSN, and benchmarking algorithms.

Figure 6.32 shows a comparison of the slot utilisation optimisation for RBA, RBL, SL, RL, and OT. Again, as discussed in Section 4.6, ideally slot utilisation should lie between 0.6 to 0.8. If the slot utilisation is below 0.6, a significant number of slots allocated are not utilised; if the slot utilisation is above 0.8, there is a risk that the slot allocation is insufficient to transport all packets with concomitant slight increases in the number of packets in the buffer. For the purposes of the present comparison, a slot utilisation between 0.5 to 1 is deemed acceptable since at least half of the slots allocated is utilised. Overall, RBL yields the best performance except for Experiment 4 owing to the fact that although it allocates an optimum number of slots, a lot of packets are stored in the buffer. SL and RBA exhibit similar performance. In most scenarios, OT presents the worst performance due to poor slot utilisation, especially for large buffer size; the exception is in Exp4 due to relatively higher occupied buffer space (*OBS)* and packet per slot (*PPS)*.

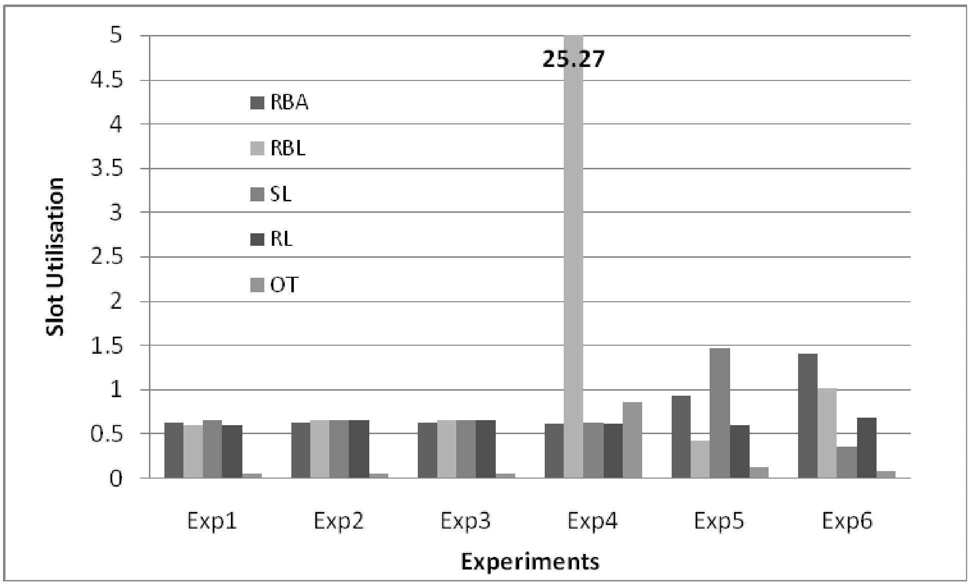

Figure 6.32. A comparison of the slot utilisation between Rule-based CogWSN, Rule-based CogWSN with Greedy Scoring, Supervised CogWSN, Reinforcement CogWSN, and benchmarking algorithms.

An additional three experiments were conducted to evaluate the performance of the three approaches under repeated scenarios. Generally, CogWSN with learning capabilities need to acquire the knowledge and experience from repeated scenarios in order to enhance the existing knowledge. The experiment setup, initial parameters, and targets are as stated in Table 6.4.

Table 6.4. Experiment setup, initial parameters, and targets for Experiments 7 to 9.

Experiments and Initial Setup	**Initial Parameters**	**Target 1**	**Target 2**
Experiment 7 Setup: The node is placed close to the base station and becomes stable; the node is then moved far from the base station but still with the communication range. Target 1 is performed. After Target 1 is achieved, the node is moved back near to the base station. Target 2 is performed. This cycle is repeated three times.	Power transmission, -10dBm; slot allocation for parent node, 16; slot allocation for child, 16; sampling rate, 1Hz	To test how fast the received power will achieve between -85dBm to -75dBm	To test how fast the received power will achieve between -85dBm to -75dBm
Experiment 8 Setup: When the node is placed close to the base station and became stable, the node is moved very far away from the base station until it is out of the communication range. After 5 minutes (data will accumulate in the flash), the node is moved back near to the base station. Target 1 is performed. After the buffer is empty (almost empty), target 2 is performed. This cycle is repeated until the third time.	Power transmission, -10dBm; slot allocation for parent node, 16; slot allocation for child, 16; sampling rate, 1Hz	To check how fast the slot allocation will tune to the maximum and successfully uploaded all the data	To test how fast the slot allocation will back to the normal
Experiment 9 Setup: Parent node is placed close to the base station. Child node is placed near to the parent node. Child node is turned on. Target 1 is performed. Afterthat, child node is turned off. Target 2 is performed. This cycle is repeated until the third time.	Power transmission, -10dBm; slot allocation for parent node, 16; slot allocation for child, 16; sampling rate, 1Hz; its' child sampling rate, 5Hz	To test how fast the slot allocation for parent and child will tune to the optimum	To test how fast the slot allocation for parent and child will tune back to the normal

In Experiment 7, at the outset the node is placed in close proximity to a base station. After the goal is achieved, the node is gradually moved away from the base station but still remains within communication range. Target 1 is performed to test how fast the received power reaches between -85dBm to -75dBm. After Target 1 is achieved, the node is then moved back nearer to the base station. Target 2 tests how fast the received power is maintained between -85dBm to -75dBm. The cycles are repeated two more times.

For Rule-based CogWSN, it takes 2 cycles to achieve Target 1 as in Figure 6.33. The node is moved away from the base station at cycles 9, 23, and 37. After Target 1 is achieved, another 2 cycles are also needed to achieve Target 2. The node is moved nearer to the base station at cycles 17, 32, and 45. Since the slot allocation to communicate with the parent and child nodes are both set to 16, several cycles are needed to meet the goal from the outset (Figure 6.34).

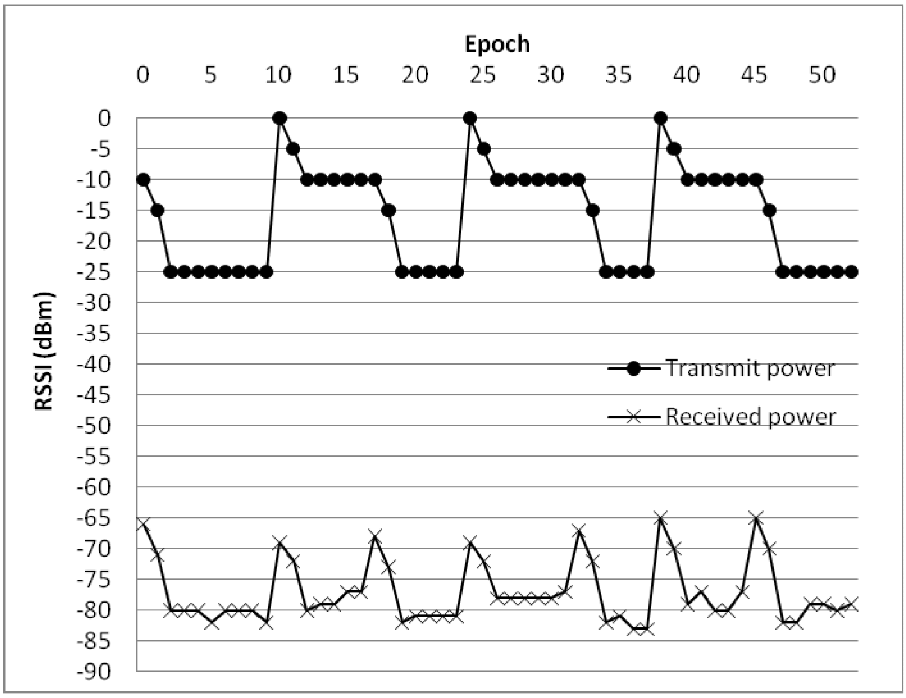

Figure 6.33. RSSI as a function of the number of cycles for Rule-based CogWSN in Experiment 7.

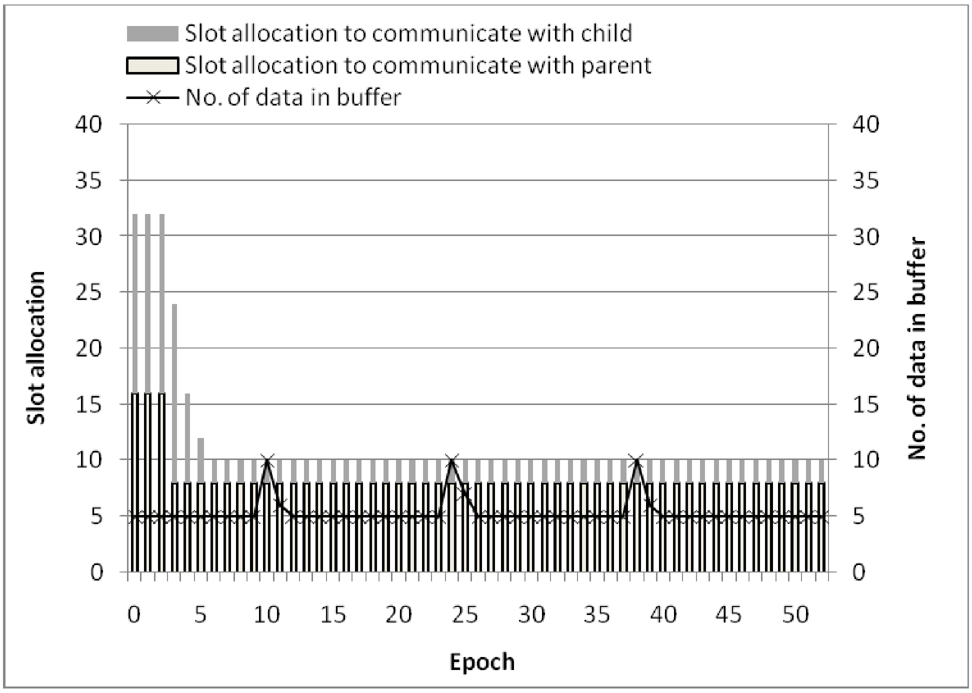

Figure 6.34. Slot allocation and buffer condition as a function of the number of cycles for Rule-based CogWSN in Experiment 7.

For Rule-based CogWSN with Greedy Scoring, the node is moved away from the base station at cycles 14, 101, and 115. A significant adjustment period (64 cycles are needed to achieve the goal) is required to meet Target 1 as shown in Figure 6.35. Any subsequent adjustments to meet Target 1 are much improved, executed in only 2 cycles. The node is moved nearer the base station at cycles 86, 110, and 123; 1 to 2 cycles are required to maintain the RSSI between -85dBm to -75dBm. Since the slot allocation to communicate with the parent and child nodes are both set to 16, 7 cycles are needed to meet the goal from the outset (Figure 6.36).

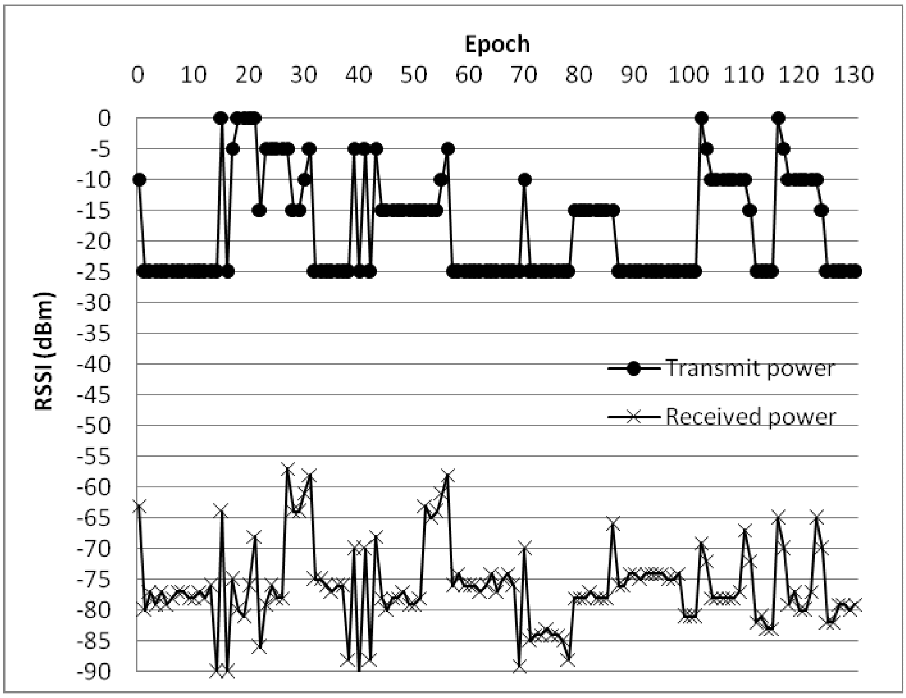

Figure 6.35. RSSI as a function of the number of cycles for Rule-based CogWSN with Greedy Scoring in Experiment 7.

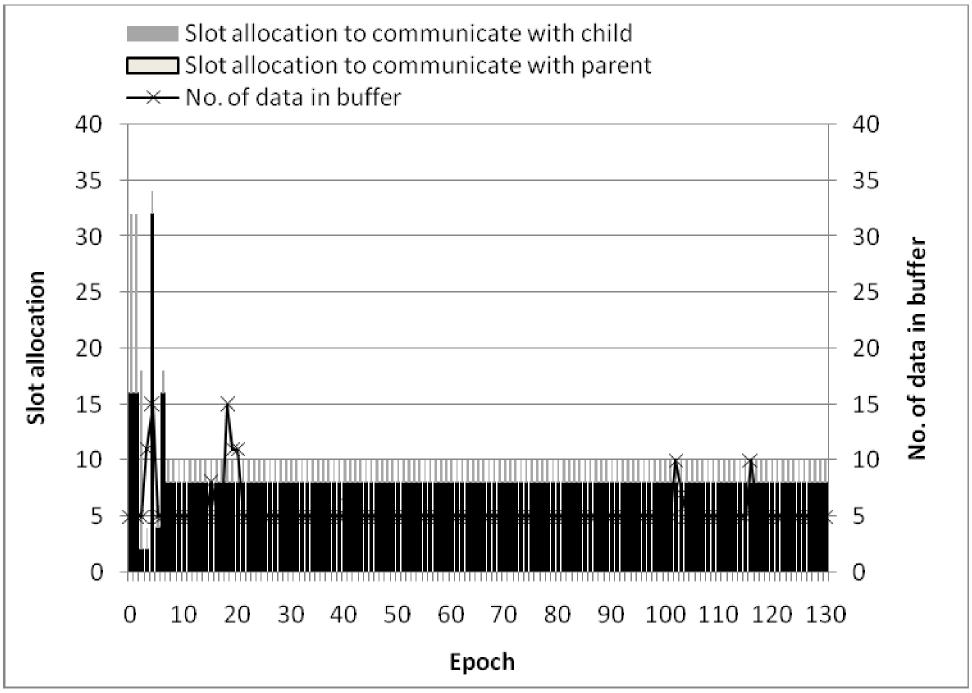

Figure 6.36. Slot allocation and buffer condition as a function of the number of cycles for Rule-based CogWSN with Greedy Scoring in Experiment 7.

The Supervised CogWSN follows the same performance as for the Rule-based CogWSN (Figure 6.37). Since the slot allocation to communicate with the parent and child nodes are both set to 16, several cycles are needed to meet the goal from the outset (Figure 6.38).

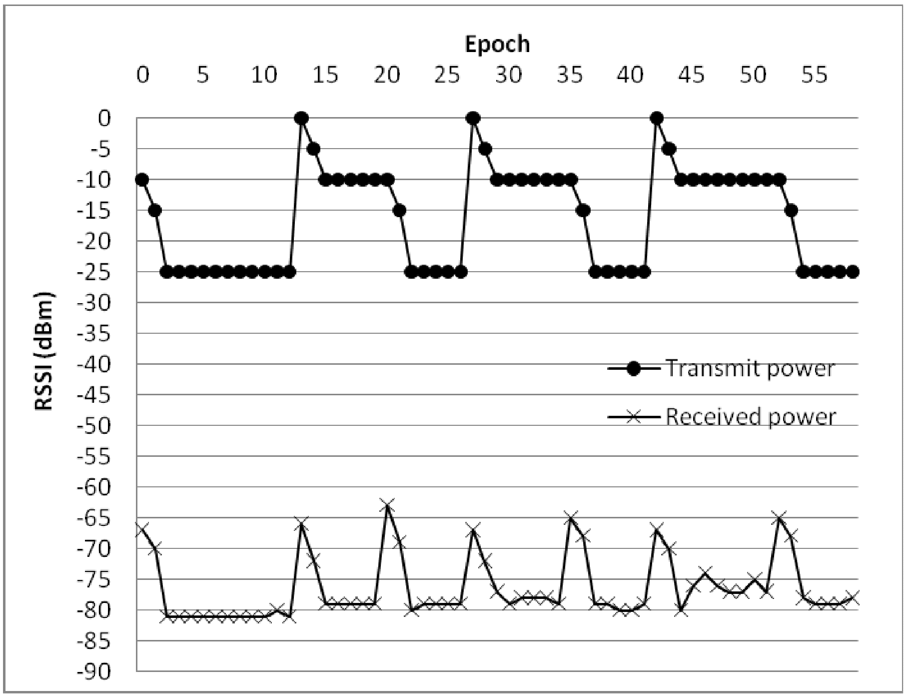

Figure 6.37. RSSI as a function of the number of cycles for Supervised CogWSN in Experiment 7.

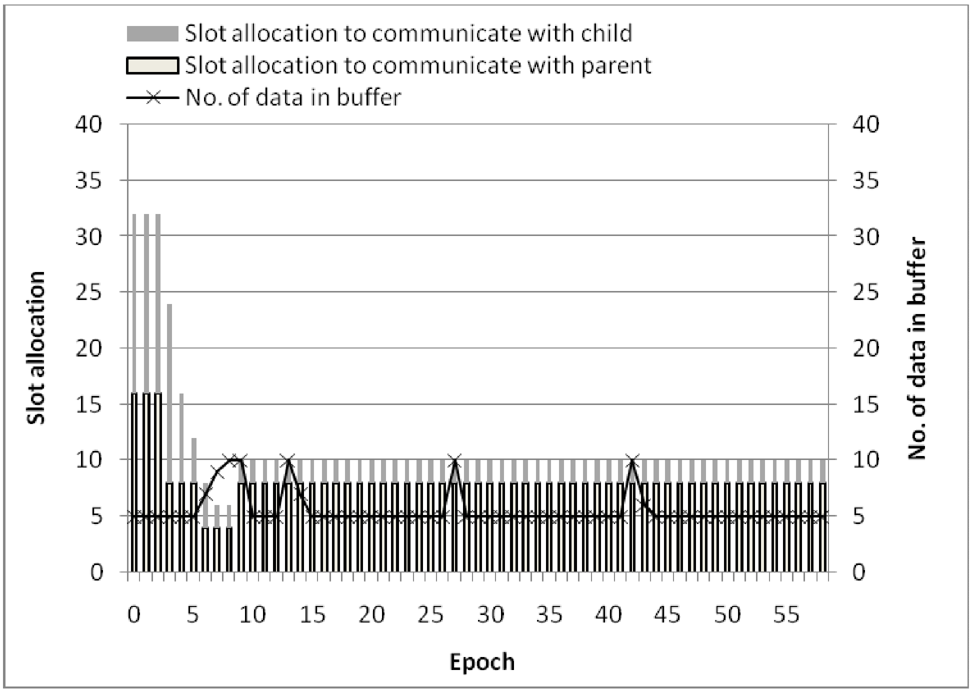

Figure 6.38. Slot allocation and buffer condition as a function of the number of cycles for Supervised CogWSN in Experiment 7.

The Reinforcement CogWSN, as shown in Figure 6.39, requires more cycles to achieve Target 1 and Target 2 as compared to both the Rule-based CogWSN and Supervised CogWSN. The slot allocation adjustment is executed during the 2nd cycle owing to the training (Figure 6.40).

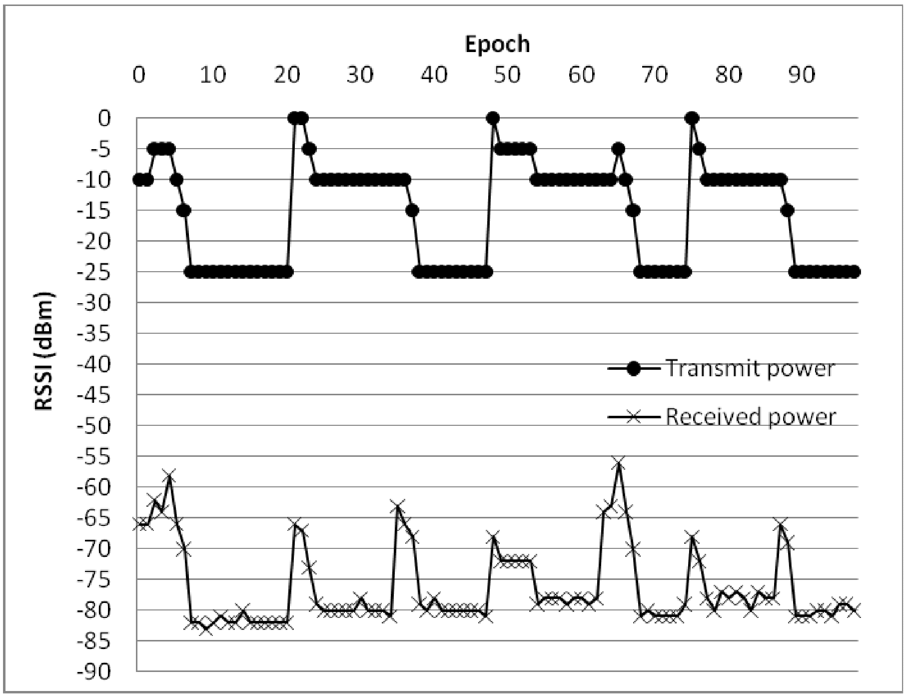

Figure 6.39. RSSI as a function of the number of cycles for Reinforcement CogWSN in Experiment 7.

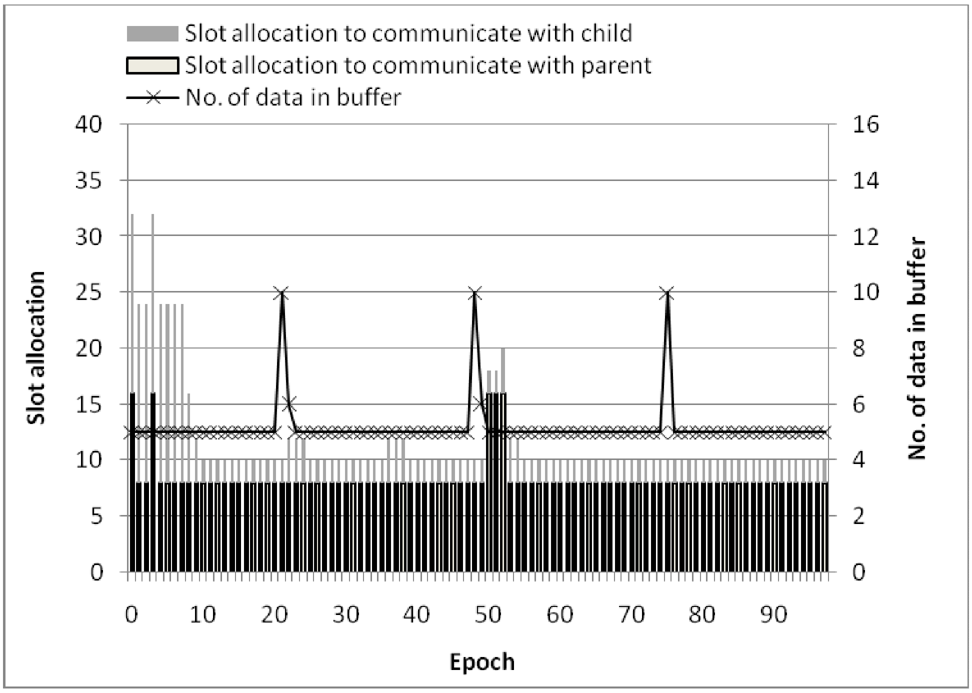

Figure 6.40. Slot allocation and buffer condition as a function of the number of cycles for Reinforcement CogWSN in Experiment 7.

For benchmarking algorithms, which lack any learning capability, the adjustments needed are shown in Figure 6.41 and Figure 6.42.

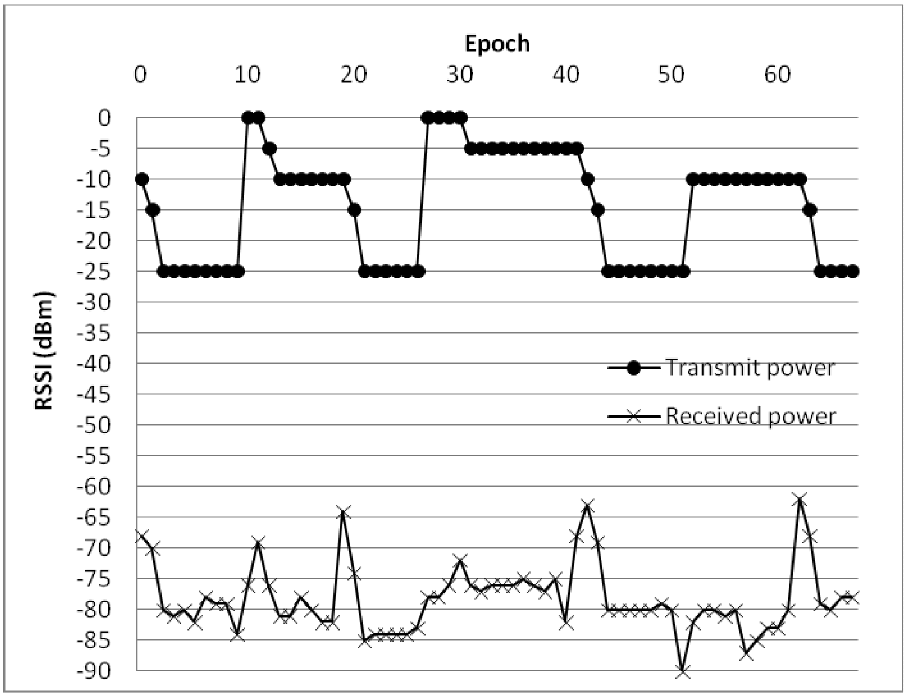

Figure 6.41. RSSI as a function of the number of cycles for Benchmarking Algorithms in Experiment 7.

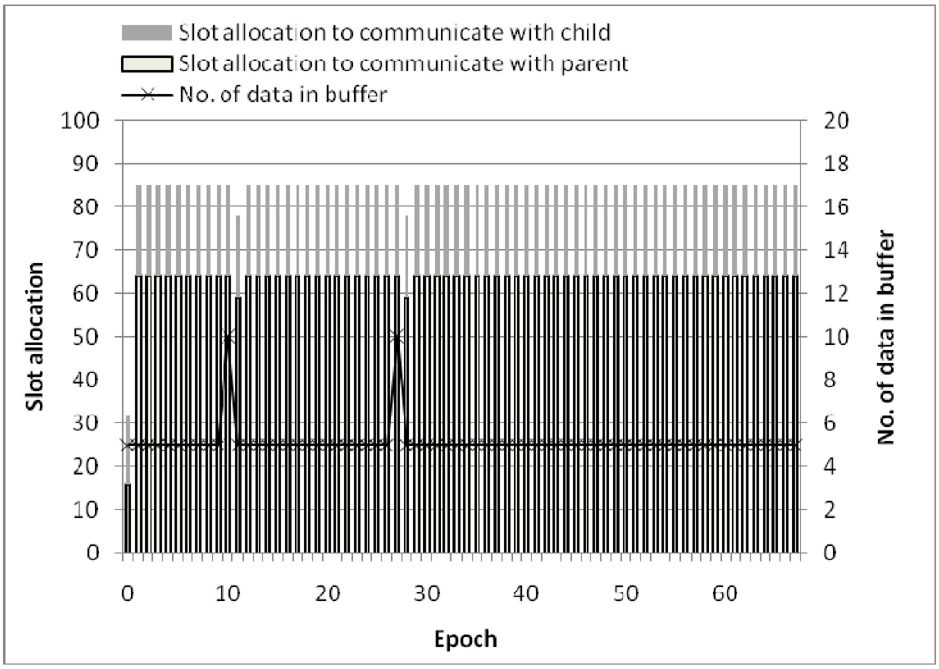

Figure 6.42. Slot allocation and buffer condition as a function of the number of cycles for Benchmarking Algorithms in Experiment 7.

In Experiment 8, at the outset the node is placed in close proximity to the base station. After the goal is achieved, the node is moved away from the base station until it falls out of communication range. After 5 minutes (the accumulated data are stored in the flash memory), the node is moved nearer to the base station. Target 1 is initiated. After the buffer is empty (or almost empty where number of packet in the buffer is less than the slot allocation to communicate with the parent node), Target 2 is initiated. This process is repeated three times.

The Rule-based CogWSN is expected to experience connectivity issue when the node approaches and leaves the coverage of the base station as shown in Figure 6.43. Since the node is continually seeking to establish a connection with the base station, the normal mode is to adjust its power transmission to 0dBm. When the node enters the communications range of the base station, it takes a maximum of 4 cycles to adjust its power transmission to -25dBm so that packets can be received successfully. The time to achieve Target 1 and Target 2 is almost constant at between 4 to 8 cycles (Figure 6.44).

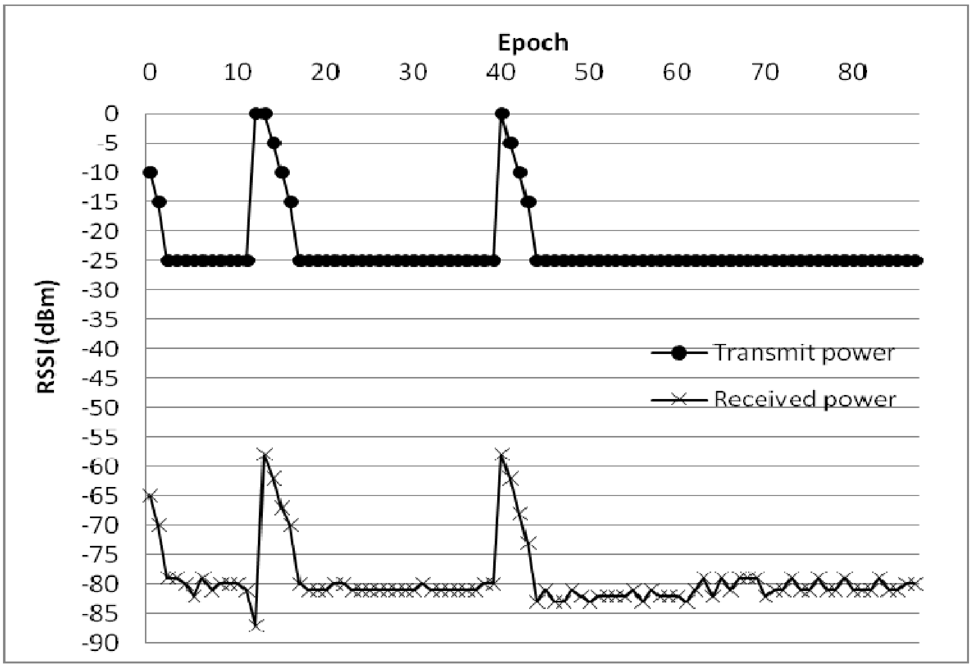

Figure 6.43. RSSI as a function of the number of cycles for Rule-based CogWSN in Experiment 8.

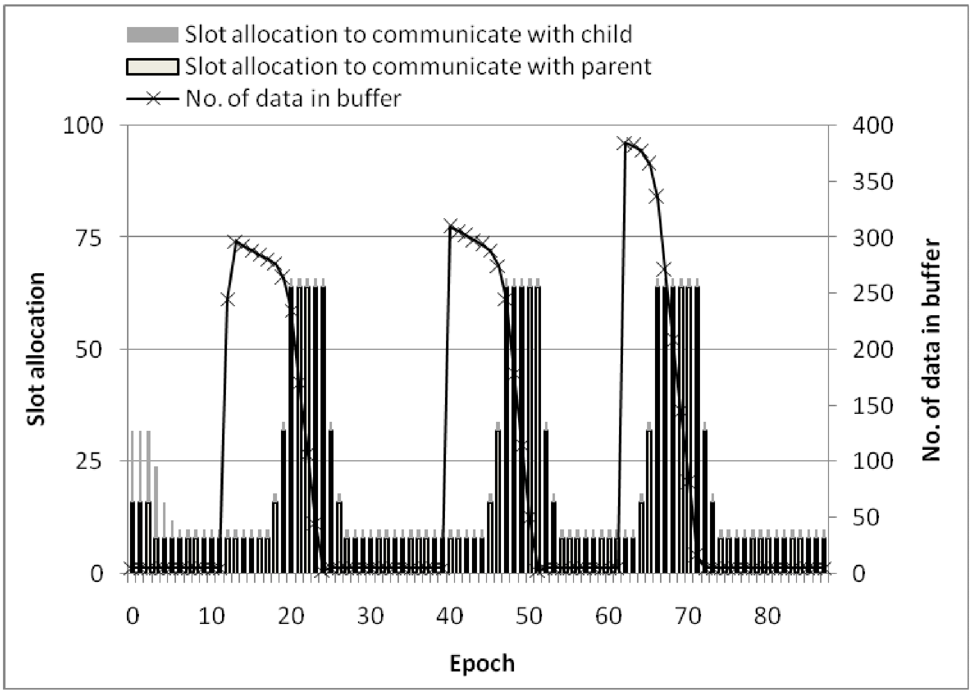

Figure 6.44. Slot allocation and buffer condition as a function of the number of cycles for Rule-based CogWSN in Experiment 8.

The Rule-based CogWSN with Greedy Scoring is also expected to experience connectivity issues when the node approaches and leaves base station coverage (Figure 6.45). The adjustment to achieve Target 1 is difficult to execute as shown in Figure 6.46, since scoring the correct rule complicates the process. Therefore, adjustment is initiated when the emptying of the buffer starts until empty. It is on the 3rd cycle that the correct order of the rules scoring is initiated. At the 4th cycle, 108th cycle, the correct adjustment is implemented and the approach exhibits a performance close to that of the Rule-based CogWSN. It is noted that the comparison result is only for 3 cycles. Rule-based CogWSN with Greedy Scoring is able to achieve acceptable performance like the Rule-based CogWSN but requires more cycles to achieve the goal.

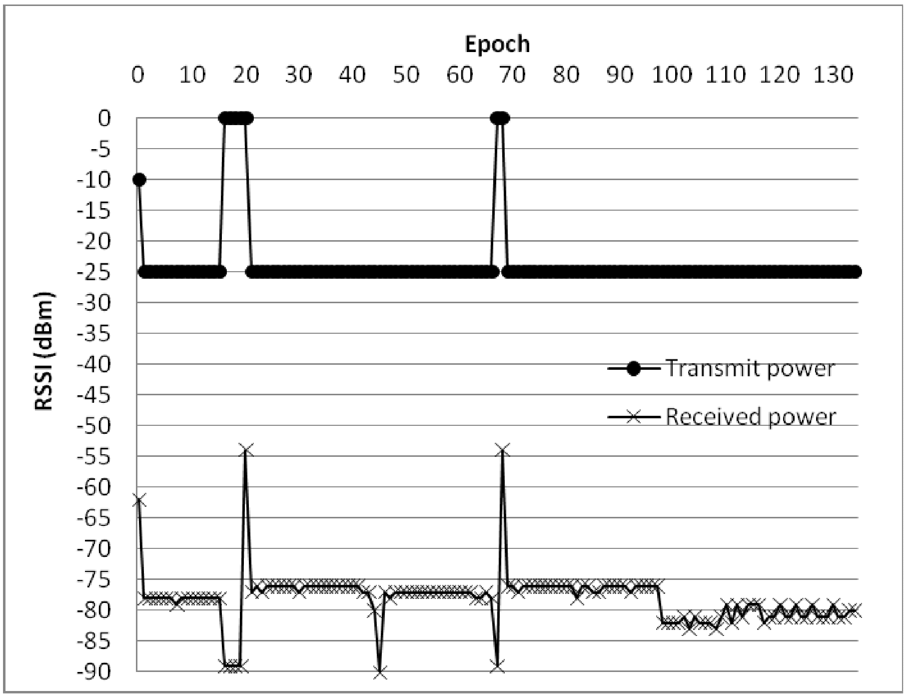

Figure 6.45. RSSI as a function of the number of cycles for Rule-based CogWSN with Greedy Scoring in Experiment 8.

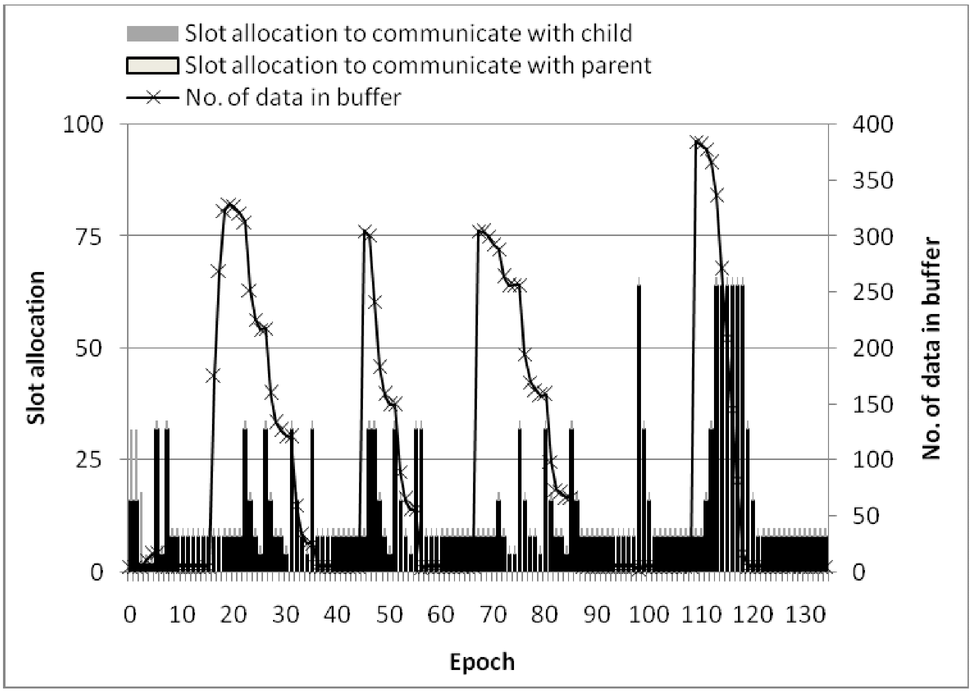

Figure 6.46. Slot allocation and buffer condition as a function of the number of cycles for Rule-based CogWSN with Greedy Scoring in Experiment 8.

The Supervised CogWSN, as shown in Figure 6.47 and Figure 6.48, solves the problem using the same principles as in Rule-based CogWSN. For the 2nd and 3rd cycles, it chooses to adjust the slot allocation to clear the buffer, followed by a reduction in the transmission power. In this case, the number of cycles to achieve Target 1 is reduced by at least 50%.

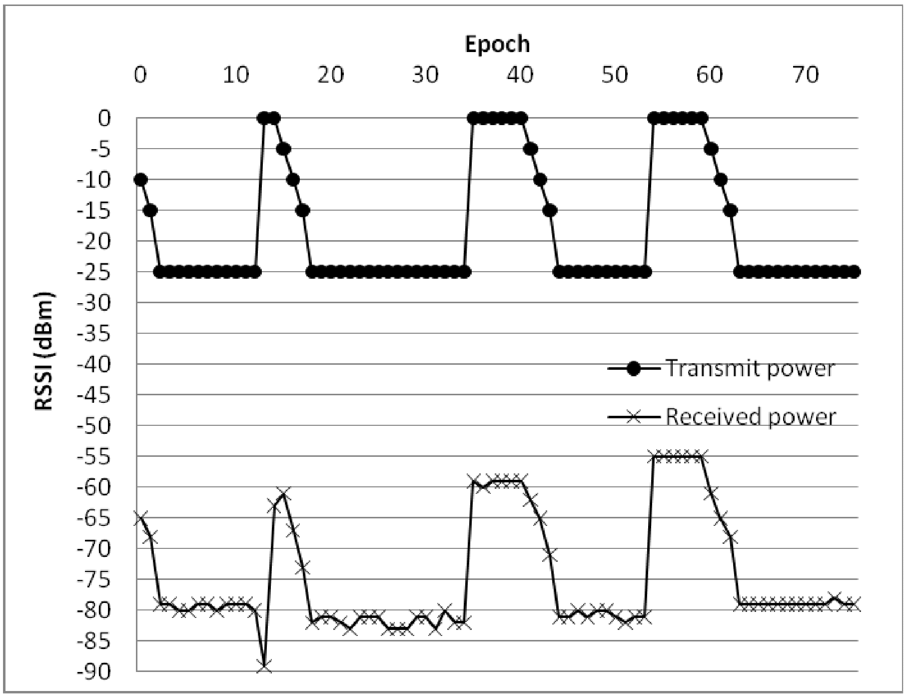

Figure 6.47. RSSI as a function of the number of cycles for Supervised CogWSN in Experiment 8.

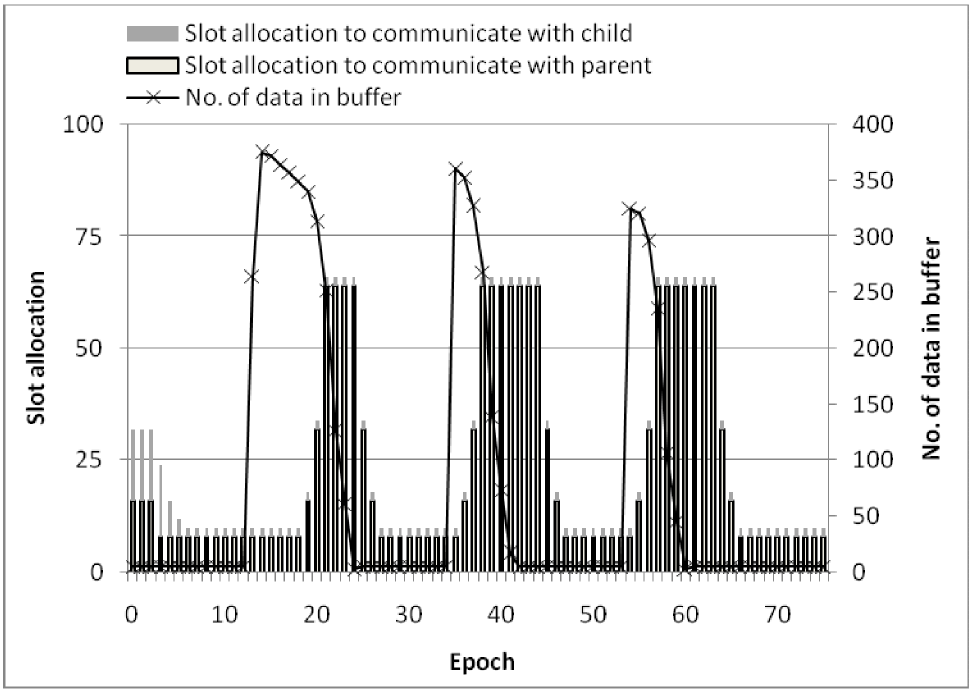

Figure 6.48. Slot allocation and buffer condition as a function of the number of cycles for Supervised CogWSN in Experiment 8.

The Reinforcement CogWSN exhibits a consistent performance for Target 1 and Target 2 as shown in Figure 4.49 and Figure 4.50. Several cycles to adjust are required due to no training or access to any *a priori* knowledge.

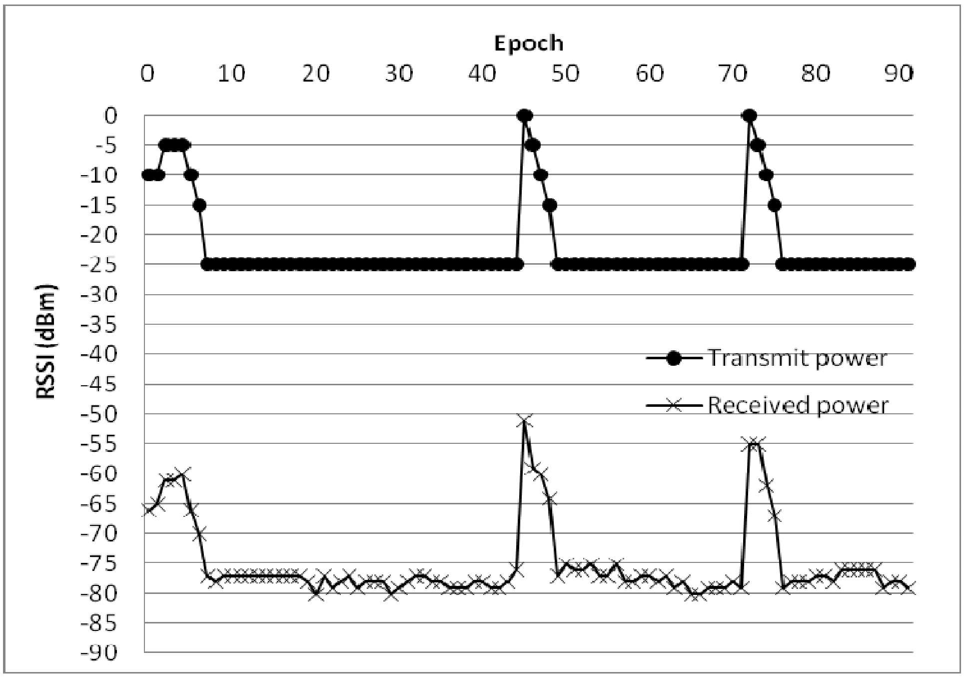

Figure 6.49. RSSI as a function of the number of cycles for Reinforcement CogWSN in Experiment 8.

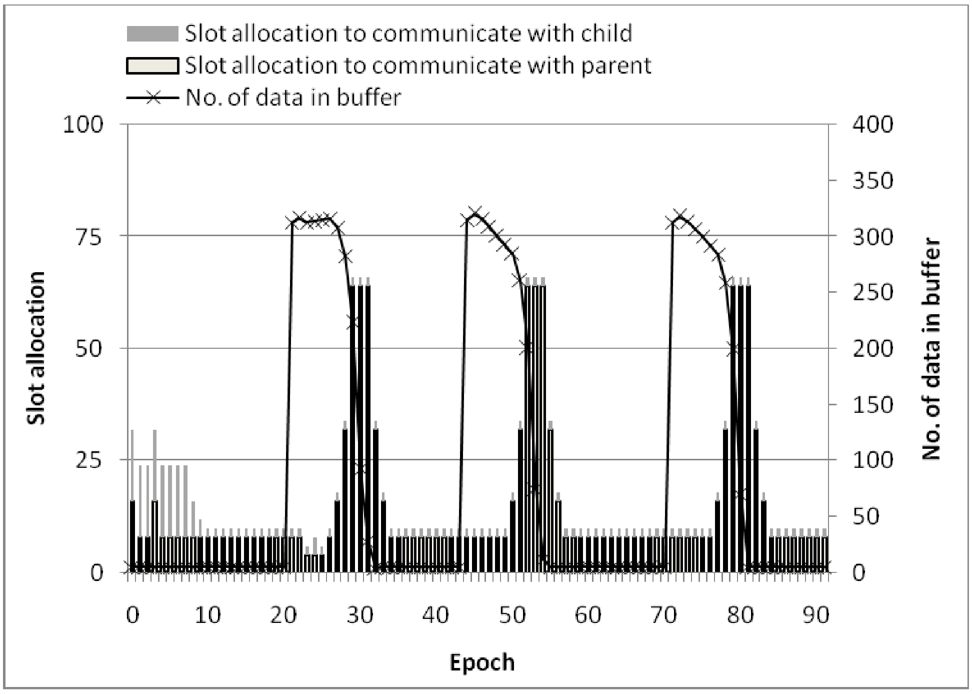

Figure 6.50. Slot allocation and buffer condition as a function of the number of cycles for Reinforcement CogWSN in Experiment 8.

For benchmarking algorithms, no adjustment of the power transmission is required since the RSSI is maintained between -85dBm to -75dBm as shown in Figure 6.51. For the slot allocation, benchmarking algorithms struggle to optimize performance as shown in Figure 6.52. When occupied buffer space (*OBS)* and packet per slot (*PPS)* are relatively high, the total slot allocation (*TSA)* is low; while occupied buffer space (*OBS)* is high and packet per slot (*PPS)* is relatively low, total slot allocation (*TSA)* is high. This causes the slot allocation to fluctuate when the buffer is downloading.

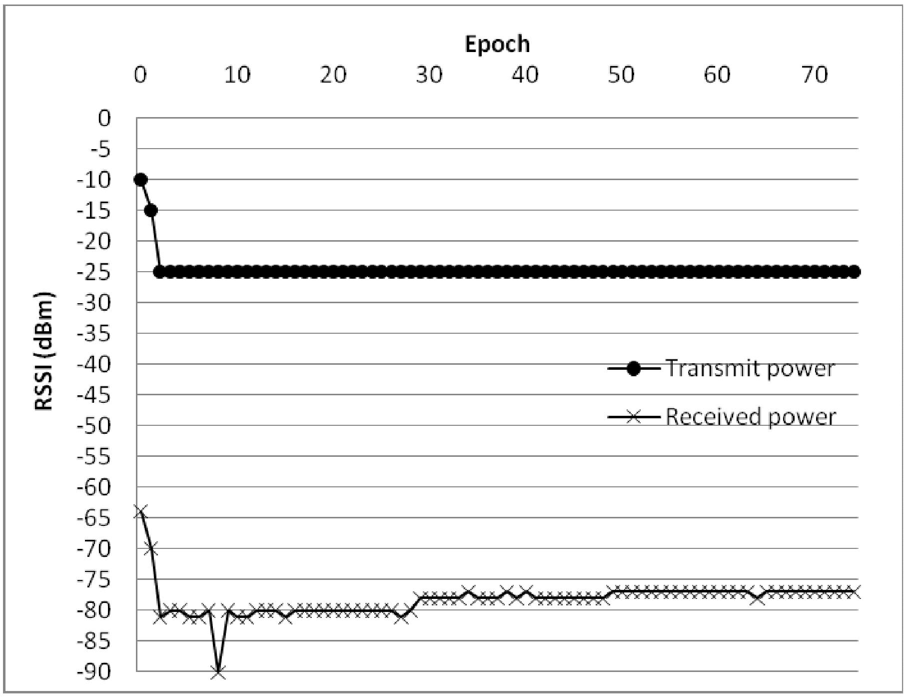

Figure 6.51. RSSI as a function of the number of cycles for Benchmarking Algorithms in Experiment 8.

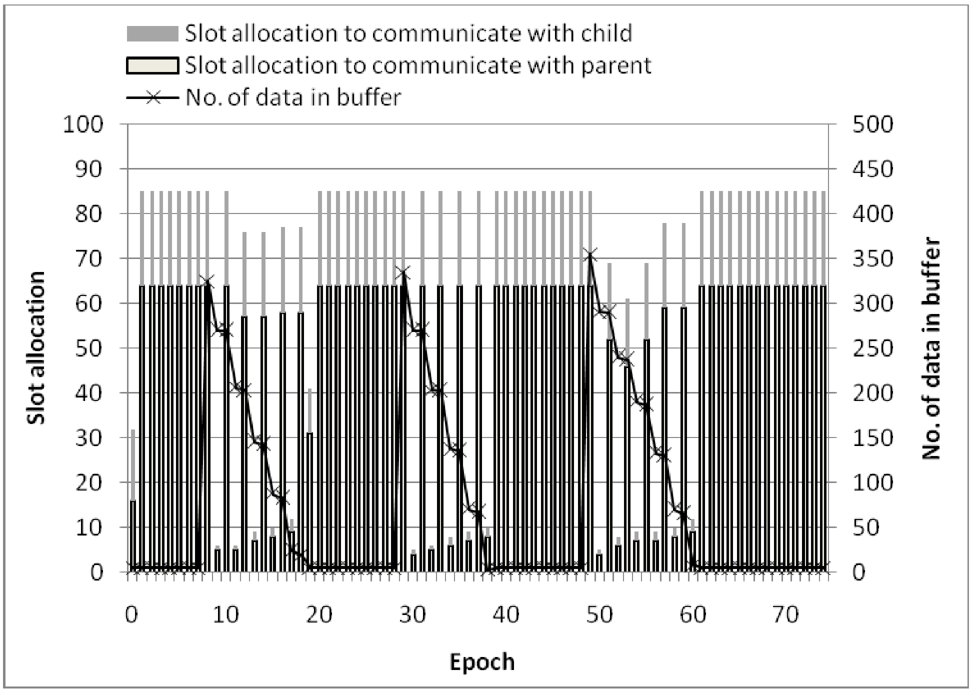

Figure 6.52. Slot allocation and buffer condition as a function of the number of cycles for Benchmarking Algorithms in Experiment 8.

In Experiment 9, a node is configured and placed in close proximity to the base station. The node is turned on until the goal is achieved. Then, a child node is configured and placed near to the deployed node (becoming its parent node). The child node is turned on and Target 1 is initiated. After a period (until the node achieves its goal or if not, around 20 cycles), the child node is turned off and Target 2 is initiated. This cycle is repeated three times.

For Rule-based CogWSN, the adjustment of transmission power is only performed when the received power falls outside the targeted range of -85dBm to -75dBm (Figure 6.53). When the child node is turned on, 6 to 7 cycles are required to adjust the slot allocation to communicate with the parent and child nodes to the maximum requirement; when the child node is switched off, 2 to 3 cycles are required to reduce the slot allocation to the minimum requirement (Figure 6.54).

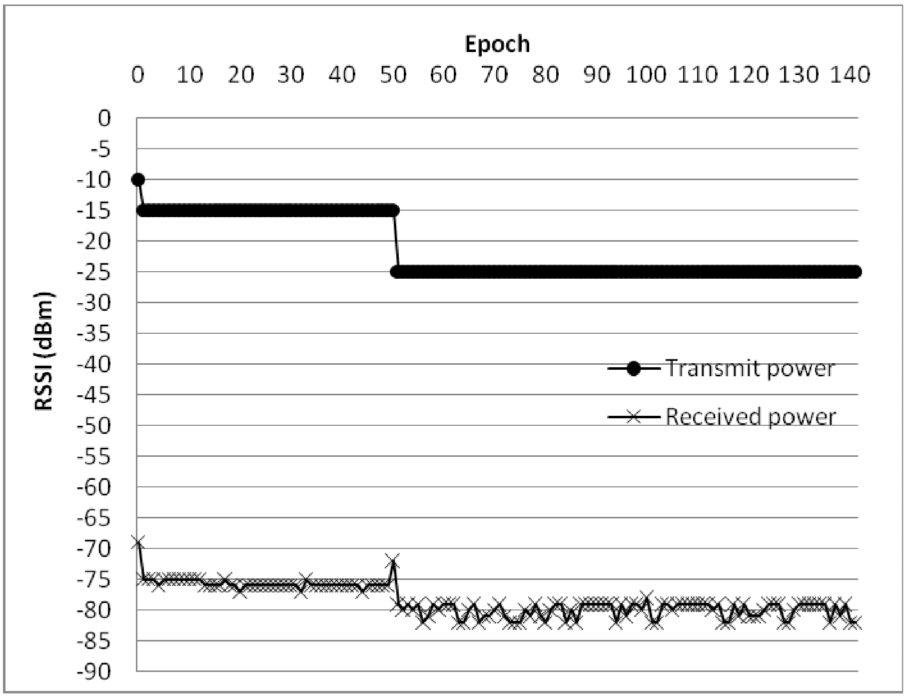

Figure 6.53. RSSI as a function of the number of cycles for Rule-based CogWSN in Experiment 9.

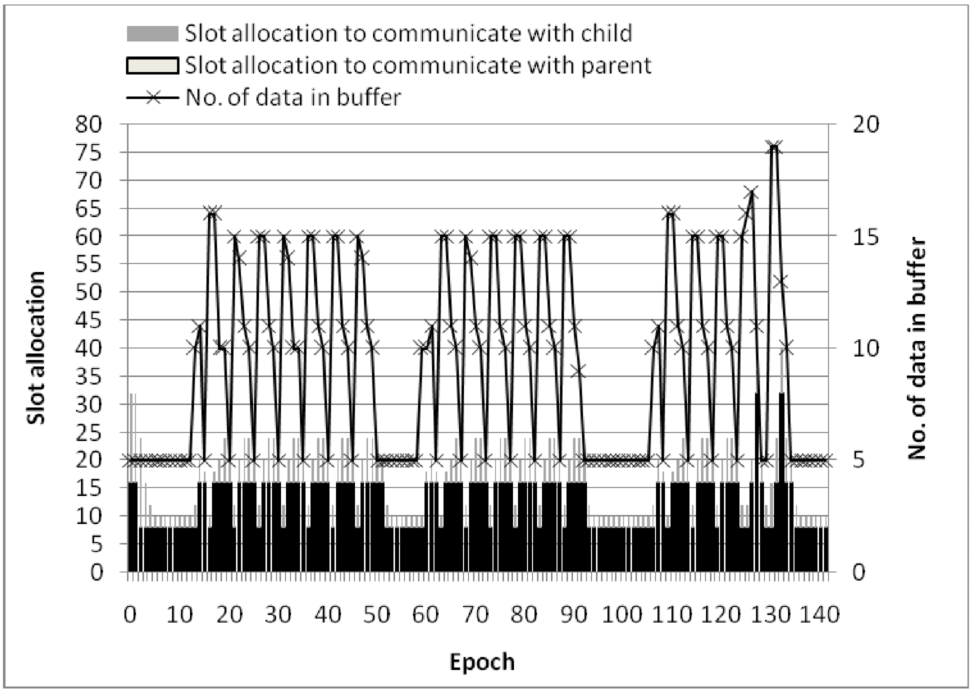

Figure 6.54. Slot allocation and buffer condition as a function of the number of cycles for Rule-based CogWSN in Experiment 9.

For Rule-based CogWSN with Greedy Scoring, the adjustment of the transmission power is only performed when the received power falls outside the targeted range of -85dBm to -75dBm (Figure 6.55). When the child node is turned on, 2 to 7 cycles are required to adjust the slot allocation to communicate with the parent and child nodes to the maximum requirement; while when the child node is switched off, 2 to 3 cycles are required to reduce the slot allocation to the minimum requirement (Figure 6.56).

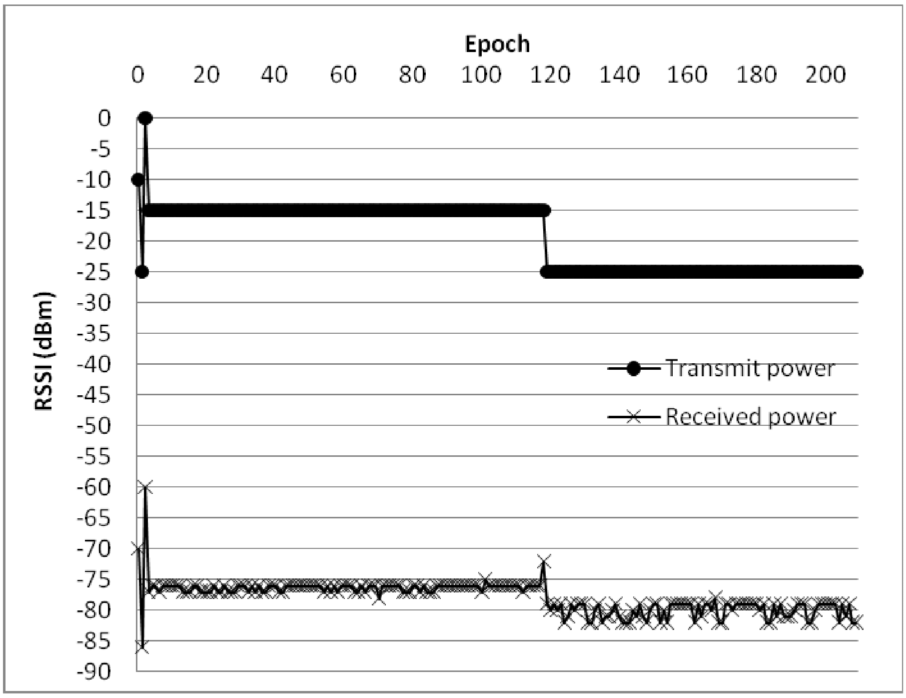

Figure 6.55. RSSI as a function of the number of cycles for Rule-based CogWSN with Greedy Scoring in Experiment 9.

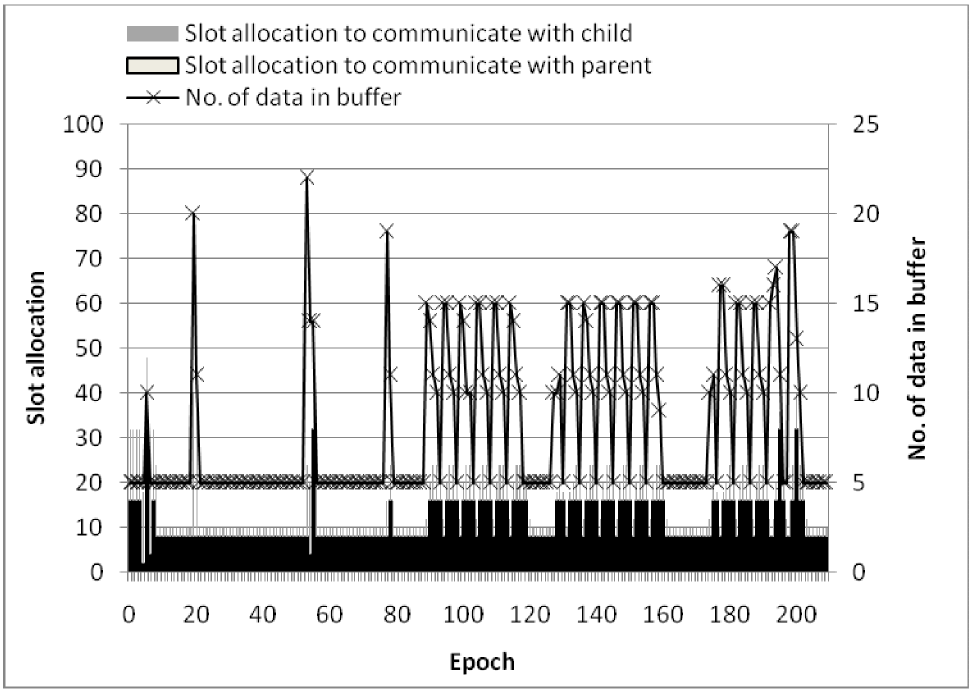

Figure 6.56. Slot allocation and buffer condition as a function of the number of cycles for Rule-based CogWSN with Greedy Scoring in Experiment 9.

For Supervised CogWSN, no adjustment is done for the power transmission when the child node is turned on and off as shown in Figure 6.57. When the child node is turned on, 2 to 5 cycles are required to adjust the slot allocation to communicate with parent and child nodes to the maximum requirement, while when switched off, 1 to 2 cycles are required to reduce the slot allocation to the minimum requirement (Figure 6.58).

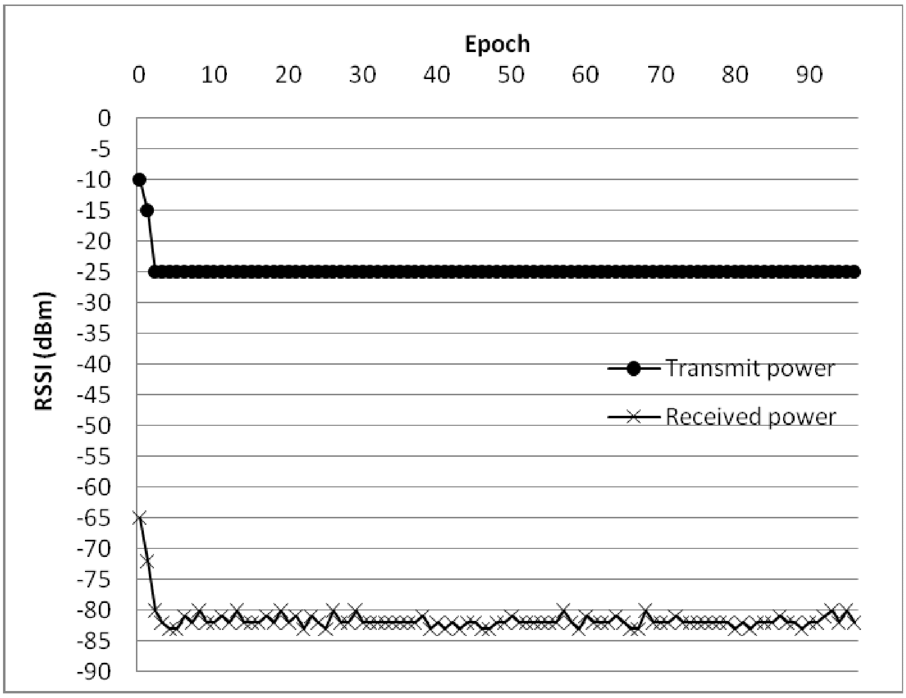

Figure 6.57. RSSI as a function of the number of cycles for Supervised CogWSN in Experiment 9.

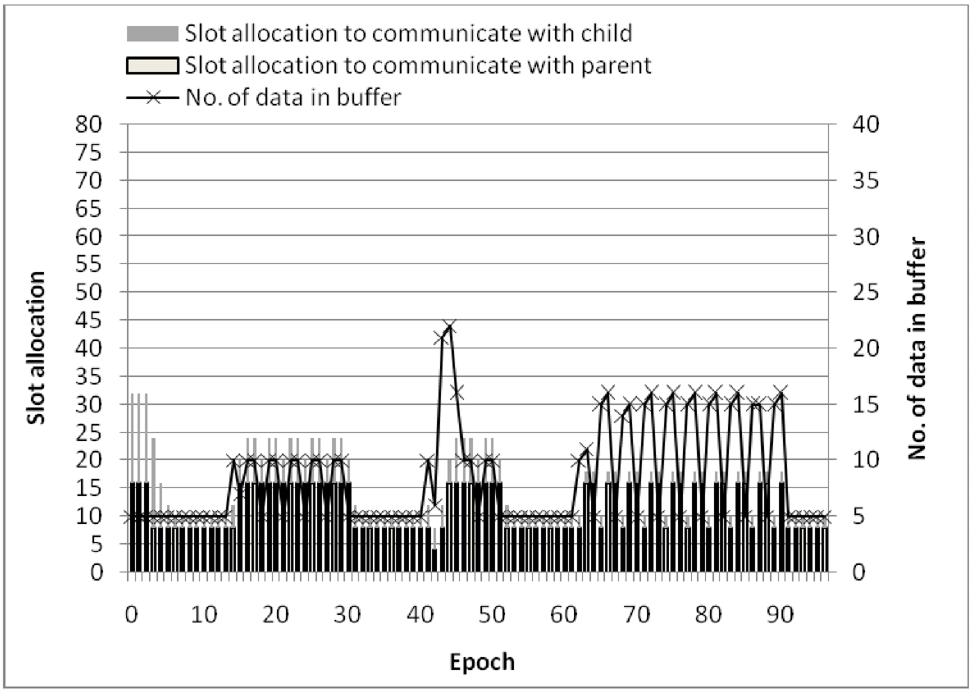

Figure 6.58. Slot allocation and buffer condition as a function of the number of cycles for Supervised CogWSN in Experiment 9.

For Reinforcement CogWSN, the adjustment of the transmission power only occurs when the received power falls outside the targeted range of -85dBm to -75dBm (Figure 6.59). When the child node is turned on, 2 to 20 cycles are required to adjust the slot allocation to communicate with parent and child nodes to the maximum requirement; while when the child node is switched off, 2 to 6 cycles are required to reduce the slot allocation to the minimum requirement (Figure 6.60).

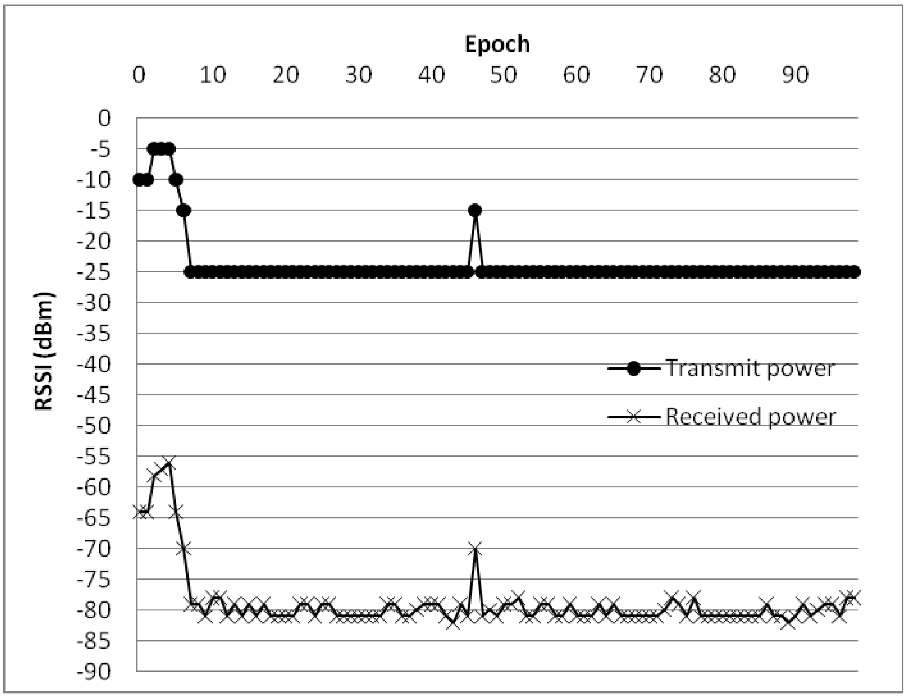

Figure 6.59. RSSI as a function of the number of cycles for Reinforcement CogWSN in Experiment 9.

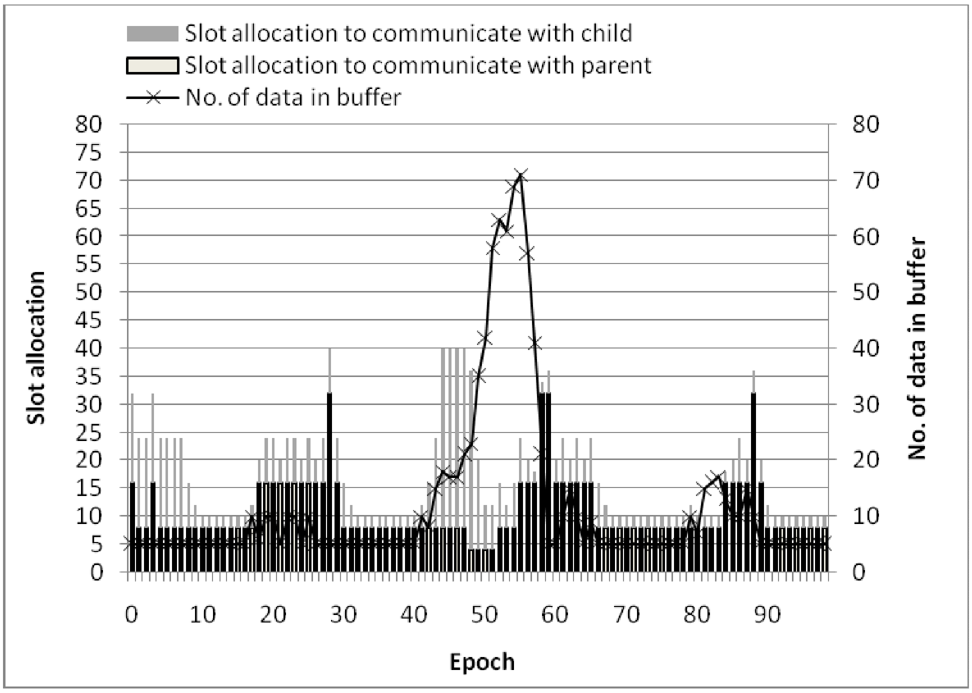

Figure 6.60. Slot allocation and buffer condition as a function of the number of cycles for Reinforcement CogWSN in Experiment 9.

For the Benchmarking Algorithms, no adjustment is required when the child node is turned on and off (Figure 6.61). When the child node is turned on, no appreciable adjustment is evident; however adjustment is performed continuously until the child node is switched off (Figure 6.62).

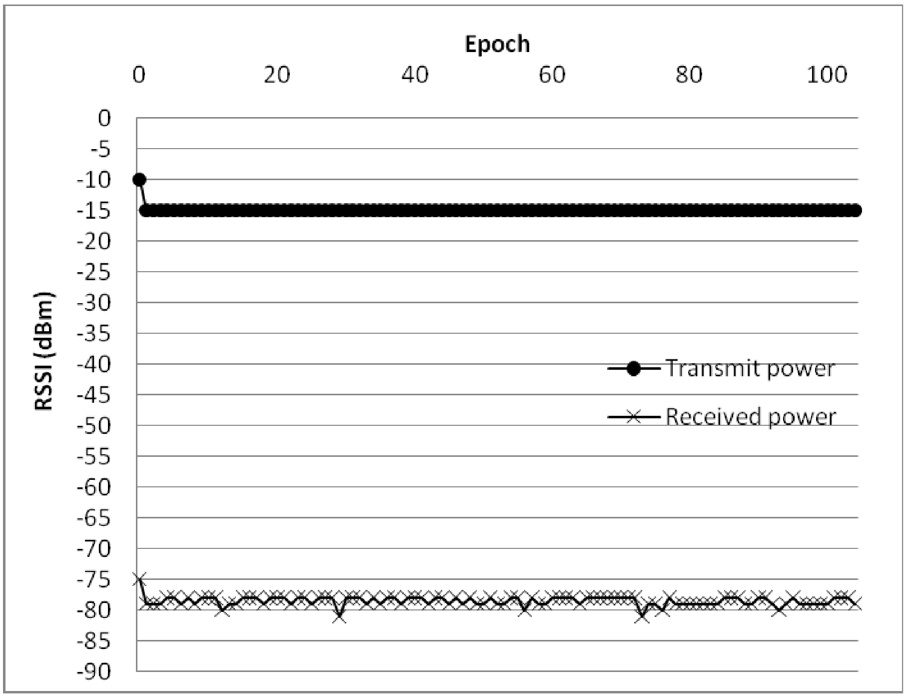

Figure 6.61. RSSI as a function of the number of cycles for Benchmarking Algorithms in Experiment 9.

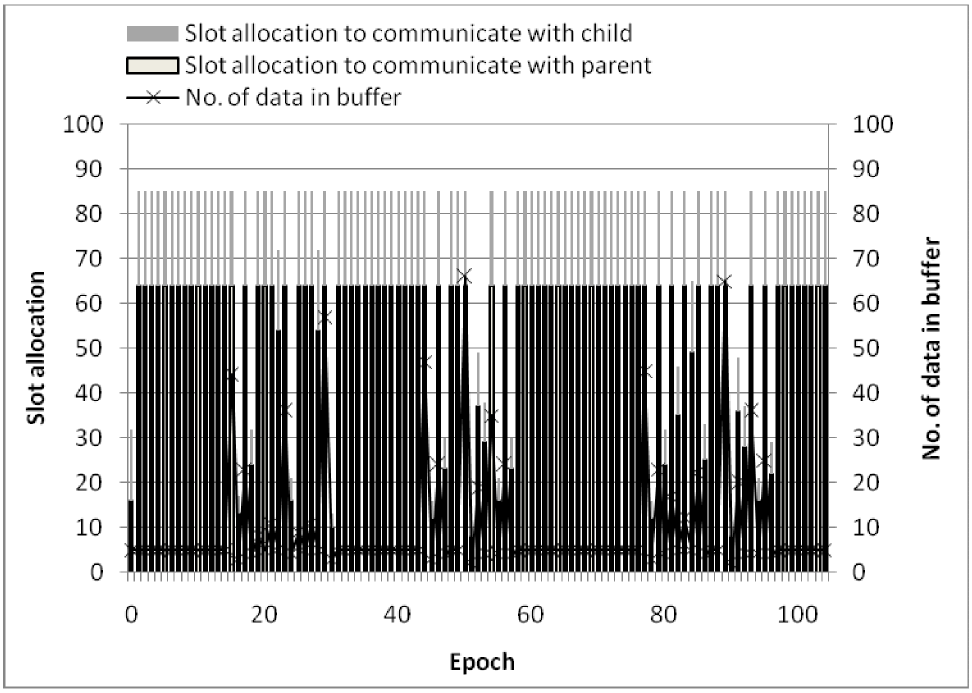

Figure 6.62. Slot allocation and buffer condition as a function of the number of cycles for Benchmarking Algorithms in Experiment 9.

A summary of the number of cycles needed to achieve the goal for each approach is summarised as in Table 6.5 for the three repeated scenarios.

Table 6.5. Number of cycles to achieve the goal for repeated scenarios.

	No. of cycles to achieve the goal (first time, second time, third time)				
	RBA	RBL	SL	RL	OT
Experiment 7 – Target 1	2, 2, 2	64, 2, 2	2, 2, 2	3, 6, 2	3, 4, 1
Experiment 7 – Target 2	2, 2, 2	1, 2, 2	2, 2, 2	2, 3, 2	2, 2, 2
Experiment 8 – Target 1	8, 7, 4	18, 10, 18	8, 3, 3	8, 8, 8	10, 8, 10
Experiment 8 – Target 2	3, 3, 3	1, 1, 1	3, 5, 6	3, 3, 3	1, 1, 1
Experiment 9 – Target 1	6, 7, 6	2, 7, 6	3, 5, 2	2, 20, 7	14, 12, 18
Experiment 9 – Target 2	3, 3, 2	3, 2, 2	2, 2, 1	6, 4, 2	1, 1, 1

Scenarios emulated in Experiments 7 to 9 have elements of behaviour encountered in real deployments. Experiment 7 is a scenario encountered in for example, cattle monitoring applications [5, 218]. A cow equipped with a collar (a node) can move close to and far from a base station many times in a day. Experiment 8 emulates for example, an agricultural field monitoring application [5] in which a node stores a certain number of packets and downloads to a base station several times over a period of a day or a week. In the latter application [5] most of the nodes are unable to reach a base station directly and consequently packets are transmitted to their parent nodes. Experiment 9 emulates unpredictable, dynamically shifting scenarios. Figure 6.63 shows the average cycles needed per each change to achieve the goal for the first to the third repeated run in Experiments 7 to 9. On average, Supervised CogWSN (SL) requires the fewest cycles per change to achieve the goal, closely followed by Rule-based CogWSN (RBA). Reinforcement CogWSN (RL) has similar performance to Benchmarking Algorithms (OT). Rule-based CogWSN with Greedy Scoring requires the most cycles per change.

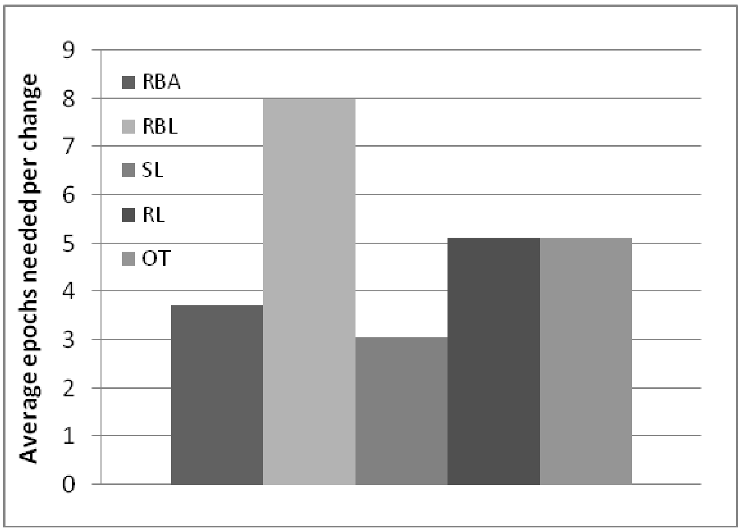

Figure 6.63. Average cycles per each change needed for 1st to 3rd repeated runs in Experiments 7 to 9.

Figure 6.64 shows the number of cycles needed for the 3rd repeat run in Experiment 7 to 9. CogWSN with learning capability (RBL, SL, and RL) require some time to learn and enhance its knowledge of the target goals. At the 3rd repeat, CogWSN with learning capability is able to reduce the number of cycles per change in a more efficient manner compared to the 1st or the 2nd cycle.

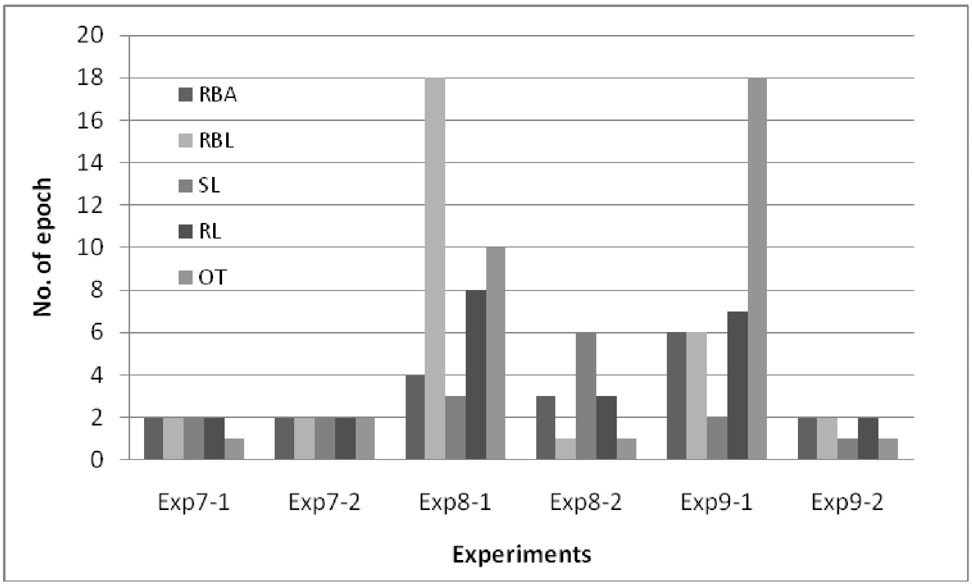

Figure 6.64. Number of cycles per change needed for the 3rd repeat run in Experiment 7 to 9.

Figure 6.65 shows the average number of cycles per each change for the 3rd repeat run. Supervised CogWSN (SL) provides the best performance, followed by Rule-based CogWSN (RBA), Reinforcement CogWSN (RL), Rule-based CogWSN with Greedy Scoring (RBL), and Benchmarking Algorithms (OT).

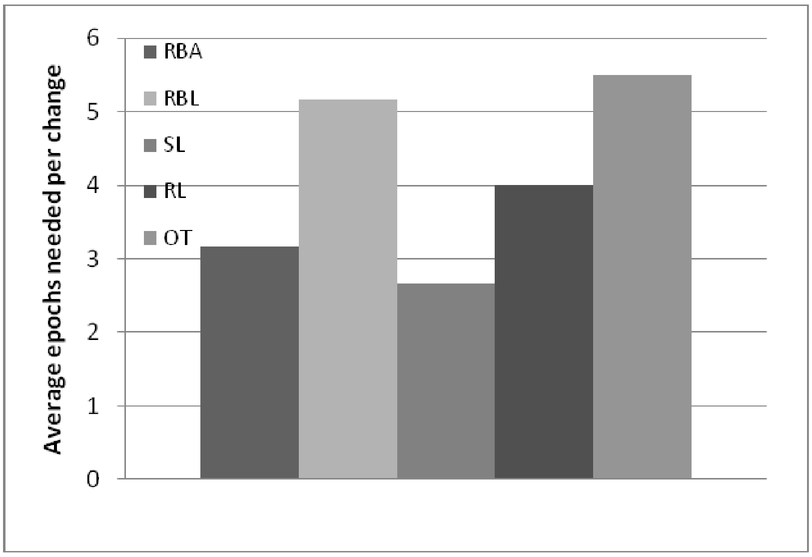

Figure 6.65. Average cycles per each change for the 3rd repeat run in Experiments 7 to 9.

6.8 Conclusions

In some WSN deployments [4, 5, 218], there are rules that cannot be defined since the optimal solution is unknown. However, it is still possible to identify suitably bounded targets or goals. For this scenario, it is most appropriate to apply Reinforcement Learning in CogWSN since the deployment cannot utilise *a priori* knowledge in the decision process.

In order to integrate Reinforcement Learning into CogWSN, 27 states and 6 actions are defined, where the states are formed from 6 states with 1 non-goal status, 12 states with 2 non-goal statuses, 8 states with 3 non-goal statuses, and 1 goal. The actions are formulated from 3 functions to adjust and in each function the adjustment can be increased or decreased. The Observe Phase is implemented using event-based principles. In the Plan Phase, all inputs are mapped into the defined states. The action is selected based on the policy. In the Implement Phase, the selected setting is 'tune up' or 'tune down'. In the Evaluation Phase, the quality of the state-action pair is updated accordingly based on goals.

Reinforcement CogWSN is always expected to take a longer time to achieve its targeted goal at the outset, as a consequence of the lack of pre-installed *a priori* knowledge. Once the goal is achieved, the experience acquired is foundation knowledge to be used in repeating scenarios. Results show that Reinforcement CogWSN is able to achieve the goals without any pre-installed knowledge during the deployment. The disadvantage is, on average, at least twice as many cycles are needed to reach the goal compared to other methods; in the first cycle, a longer period is consumed to explore its states in order to determine the correct solution. As more and more scenarios are repeated, the knowledge acquired is core to reducing the number of cycles required to find the solution. Based on the Experiments 7 to 9, in Figure 6.63, it is clear that Reinforcement CogWSN is able to achieve better performance as compared to the Rule-based CogWSN with Greedy Scoring and is on par with Benchmarking Algorithms. Furthermore in Figure 6.65, it has the best performance after Supervised CogWSN and Rule-based CogWSN in a repeated scenario.

Table 6.6 presents the program size and memory in bytes requirements for each solution. RBA, RBL, and OT are best from an implementation perspective since they consume relatively small levels of ROM and RAM resource. However in scenarios where optimal rules and solutions cannot or are too difficult to be pre-determined, SL and RL can be considered solutions at the expense of an additional 25% of ROM and 50% of RAM resource as compared to RBA, respectively. SL occupies more ROM compared to RL due to the resource allocated to artificial neurons in Supervised CogWSN which is more than the resource used to represent the states in Reinforcement CogWSN. The maximum allocation for ROM is 131072 bytes and RAM is 4096 bytes. The proposed algorithms only occupy a maximum 20% of ROM and 23% of RAM overall.

Table 6.6. Program size and memory for Rule-based CogWSN, Rule-based CogWSN with Greedy Scoring, Supervised CogWSN, Reinforcement CogWSN, and benchmarking algorithms.

	Program size and memory (Bytes)					
	Without any enhanced algorithm	RBA	RBL	SL	RL	OT
ROM	20198	20896	21876	25134	23848	21244
RAM	508	612	672	858	939	613

Chapter 7: Conclusions and Future Work

7.1 Conclusions

The demands of monitoring applications drive the design and development of WSNs, a system comprising a group of sensor nodes equipped with a short range communication capability which are often deployed over large scale at low cost. Due to the cost goals and market dynamics, low data rate transceivers operating in license-exempt ISM frequency bands are most often used.

The limited radio capability and uncertainty in physical operating environments pose fundamental constraints in optimizing WSN connectivity. Currently, a WSN lacks the ability to fine tune its radio configuration to meet the challenges of a dynamic operating environment. As a result, degradation in radio link performance and unreliable network connectivity plague operation.

Manual configurations and settings for WSN deployment are labour intensive and cumbersome. Cognition embedded in WSNs - CogWSNs - offers a route to enhancing WSN functionality, providing the ability to self-tune depending on changes in the operational environment without human intervention. Such evolution necessitates new concepts and designs to be developed in order to support the implementation.

In this Thesis, the concept of CogWSN is defined. The CogWSN decision process is formed using Problem Solving cognitive processes drawn from Layered Reference Model of the Brain (LRMB) in combination with the Polya algorithm. The CogWSN decision process is formulated through 4 phases; Observe, Plan, Implement, and Evaluate. Since the base WSN node consists of a processor, transceiver, transducer, and power unit, an architecture that comprises three core virtual modules; Transceiver, Transducer, and Power Supply is adopted. CogWSN operates co-operatively between these three virtual modules. Each module contains two elements, defined as its State Information (SI), which stores information about operating conditions, and its Tuneable Function (TF), which defines the actuating function. SI is used by the Observe Phase while TF is used by the Implement Phase. The Plan and Evaluate phases operate within the decision process on the selected action and feedback to the knowledge, respectively.

CogWSN are applicable in scenarios where:

1. all triggered conditions that are exact matches to the tuning actions are known, fine-grained, incremental corrections can be made to achieve goals
2. all triggered conditions that are exact matches to the tuning actions are known, sufficient additional information is available to enable more coarse-grained corrections to achieve goals
3. partial matches of tuning actions are known
4. the goals are known but the manner in which the goals are achieved is not known

Therefore, four types of CogWSN based on different methodologies are implemented and evaluated: Rule-based CogWSN, Rule-based CogWSN with Greedy Scoring, Supervised CogWSN, and Reinforcement CogWSN. Rule-based CogWSN requires full *a priori* knowledge of the target goals to be known (for the above item 1); Rule-based CogWSN with Greedy Scoring requires all possible actions with parameters to be defined but not the decision (for the above item 2); Supervised CogWSN only requires partial trained knowledge (for the above item 3); and Reinforcement CogWSN does not need any knowledge to be installed at the outset (for the above item 4).

Rule-based CogWSN is expected to present the ideal performance in any situation since the intelligence capturing the task is embedded in a complete set of rules on the sensor nodes before deployment. An extension to the cognition to enhance performance can be implemented by assigning a score to each rule, implemented through tuning a setting with a parameter, an example being the Rule-based CogWSN with Greedy Scoring. Since Rule-based CogWSN with Greedy Scoring requires feedback in order to converge to the correct rule, (as mentioned in

Section 4.5), a time penalty results to achieve the goal. The Observe Phase requires a routine to be performed that identifies any potential issue in the monitored conditions, implemented using event-based principles. To determine whether to trigger the Plan Phase, a mapping between the observed conditions and the pre-defined goals is executed. A plan is derived based on the first detected symptom in order, where a high priority symptom such as radio link connectivity is arranged as the beginning of the order. For rule-based approaches, the solution is pre-determined according to the conditions based on an `if: then` statement. The same recommendation is appropriate for Rule-based CogWSN with Greedy Scoring. This solution is determined by selecting the highest score in the action list. If there is more than one solution with the same highest score, the action with the greatest tuning is selected, allowing the node to solve the problem more rapidly. In the Implement Phase, an action is performed according to the derived solution. The solution could be in two forms viz. 'level-up' or 'level-down' a setting or 'tune a setting' with a parameter. The advantage of the 'level-up' or 'level down' a setting is the action is adjusted step by step; however the approach suffers in that the adjustment is slow and requires several cognitive cycles to achieve the goal. The advantage of 'tuning a setting' with a parameter is the rapidity of adjustment but the selection of the correct parameter is difficult without prior knowledge which has to be embedded into the system at the Plan Phase. For the Rule-based CogWSN, the option of 'level up' or 'level down' a setting is chosen for the implementation and for the Rule-based CogWSN with Greedy Scoring, 'tune a setting' with a parameter is preferred since the setting can be tagged with the score information. The Evaluation Phase provides feedback on the action taken to check the accuracy and validity of the derived solution. The evaluation result is stored so that it can be used as 'experience' for future plan phases. The rule-based approach is crafted as an ideal solution and as such feedback is not needed and no learning is involved. However, feedback is required for rule-based learning where each rule is assigned a score.

ANNs operate based on the training using key inputs and desired outputs. Ideally, the full set of inputs and desired outputs are required; for Supervised CogWSN, the full set of inputs and desired outputs are not compulsory. For inputs that have not been used in training, the ANN is able to recommend an output (may be not the correct output) based on the training set of inputs and desired outputs. The Observe Phase is implemented using event-based observation. For the Plan Phase, the trigger is best derived based on all detected symptoms since the Supervised CogWSN is able to make decisions using multiple inputs without the overhead of space complexity on the decision. For the Implement Phase, the option of 'level up' or 'level down' a setting is chosen. For the Evaluate Phase, a look up table to filter any incorrect decision made during the Plan Phase is implemented to increase the accuracy in the decision making. If an incorrect decision is made, a record is entered into the look up table. The size of the table is governed by memory capacity.

In some WSN deployments, there are rules that cannot be defined since the optimal solution is unknown. However, it is still possible to identify suitable bounded targets or goals. For this scenario, it is most appropriate to apply Reinforcement Learning in CogWSN since the deployment cannot utilise *a priori* knowledge in the decision process. In order to integrate Reinforcement Learning into CogWSN, 27 states and 6 actions are defined (as discussed in Section 6.6), where the states are formed from 6 states with 1 non-goal status, 12 states with 2 non-goal statuses, 8 states with 3 non-goal statuses, and 1 goal, while the actions are formulated from 3 functions to adjust and in each function, the adjustment can be increased or decreased. The Observe Phase is implemented using event-based principles. In the Plan Phase, all inputs are mapped into the defined states. The action is selected based on the policy (as discussed in Section 6.3). In the Implement Phase, the selected setting is tune up or down. In the Evaluation Phase, the quality of the state-action pair is updated based on goals. Reinforcement CogWSN is always expected to take more cycles to achieve its targeted goal at the outset, as a consequence of no embedded *a priori* knowledge. Once the goal is achieved, the experience acquired is the foundation knowledge to be used in repeating scenarios. Results show that Reinforcement CogWSN is able to achieve the goals without any pre-installed knowledge during deployment. The disadvantage is, on average, at least twice as many cycles are needed to reach the goal compared to other methods; in the first cycle, a longer time period is consumed to explore its states in order to determine the correct solution. As more and more scenarios are repeated, the knowledge acquired is core to reducing the number of cycles required to find the solution.

Verification is performed through several case studies, centering on the optimisation of transmission power and communication slot allocation. All four methods are able to achieve the goal over different periods of time. Rule-based CogWSN makes adjustment step by step. Each adjustment contributes some level of improvement in solving the problem, while Rule-based CogWSN with Greedy Scoring tends to solve the problem as fast as it can but, in some scenarios, the solutions are unstable over time (constant readjustment). Supervised CogWSN exhibits similar performance to the Rule-based CogWSN and Rule-based CogWSN with Greedy Scoring. At the outset, it is necessary to train the multi-layer ANN network with ground truth inputs and desired outputs. With more computation and memory resources, a greater number of neurons can be added in hidden layers to enhance precision. Reinforcement CogWSN is always expected to take a longer time to achieve its targeted goal at the outset, as a consequence of no embedded *a priori* knowledge. Once the goal is achieved, the experience acquired is the foundation knowledge to be used in repeating scenarios. Further enhancements through the addition of improvement criteria in the Evaluate Phase to reduce the training time are possible.

Comparisons have been performed and the results show that Supervised CogWSN requires the least number of cycles per change to achieve the specified goals, followed by Rule-based CogWSN. Reinforcement CogWSN exhibits similar performance to the Benchmarking Algorithms. Rule-based CogWSN with Greedy Scoring requires the most number of cycles per change. In the case of CogWSN with learning capability, at the 3rd repeated scenario, Supervised CogWSN provides the best performance, followed by Rule-based CogWSN, Reinforcement CogWSN, Rule-based CogWSN with Greedy Scoring, and Benchmarking Algorithms.

In terms of the program size and memory requirement, RBA, RBL, and OT are best from an implementation perspective since they consume relatively small levels of ROM and RAM resource. However in scenarios where optimal rules and solutions cannot or are too difficult to pre-determine, SL and RL are potential solutions but at the expense of an additional 25% of ROM and 50% of RAM resource compared to RBA, respectively. SL occupies more ROM compared to RL due to the resource used to allocate artificial neurons in Supervised CogWSN which is more than the resource used to represent the states in Reinforcement CogWSN. The maximum allocation for ROM is 131072 bytes and RAM is 4096 bytes. The proposed algorithms only occupies a maximum 20% of ROM and 23% of RAM overall.

In summary, the Thesis has provided robust evidence with which to answer the four main research challenges stated at the outset;

1. *What can be done to minimize human intervention in tuning WSN configuration during deployment?*
   The thesis has reviewed the existing cognitive approaches currently being reported in wireless communications. From the review, a concept of a Cognitive Wireless Sensor Network (CogWSN) is proposed that provides a solution to obviate or limit human intervention in tuning the WSN configuration during deployment in dynamically changing environments. The term CogWSN is defined and a CogWSN's decision process consisting of Observe, Plan, Implement, and Evaluate Phases is described.

2. *What kinds of modification or additional elements are required for WSNs in order to support the proposed solution?*
   The CogWSN decision processes are designed from a combination of Problem Solving cognitive processes inspired by A Layered Reference Model of the Brain, (LRMB) and Polya's concept. A CogWSN architecture is developed where Transceiver, Transducer, and Power Supply virtual modules are introduced, coordinated by CogWSN's decision process with the required intervention from the user. Each virtual module consists of State Information

(SI), which stores information about the operating conditions, and Tuneable Function (TF), which defines the actuating function.

3. *How to ensure the proposed solution is able to be aware of the configuration that it needs to tune?*
   The proposed CogWSN is equipped with four learning methods: Rule-based Approach, Rule-based Learning with Greedy Scoring, Supervised Learning, or Reinforcement Learning. Rule-based CogWSN requires full knowledge to be pre-installed in order to operate. Rule-based CogWSN with Greedy Scoring requires all possible actions with parameters to be defined but not the decision. The decision on actions is scored during operation, the highest score being the preferred action. Supervised CogWSN needs at least partial knowledge to be trained on a PC and to be transferred to the nodes upon link establishment with the parent node. Reinforcement CogWSN requires no *a priori* knowledge to be installed. The nodes learn based on experience during operation and gradually establish the required knowledge. As a consequence, this kind of learning is expected to occur over a longer period of time to achieve its goal from the outset. After several repeated runs, the time to goal improves markedly.

4. *How is the performance of the proposed solution?*
   The proposed CogWSN solutions performance has been verified in terms of transmission power and communication slot allocation. The solutions are benchmarked against Benchmarking Algorithms which combine the best reported research. After several repeated runs, Supervised CogWSN exhibits the best performance, followed by Rule-based CogWSN, Reinforcement CogWSN, Rule-based CogWSN with Greedy Scoring and Benchmarking Algorithms.

7.2 Future Work

Each CogWSN solution can be improved through enhancement of its algorithms and rules. For example, Rule-based CogWSN and Rule-based CogWSN with Greedy Scoring can be refined using case-based reasoning, where the process of solving new problems can be based on solutions of similar past problems; the problems are formed into a case. Supervised CogWSN can be enhanced by adding more neurons in its hidden layers and arranged in multiple layers. However all these software enhancements require additional computation power and memory capacity. For Reinforcement CogWSN, a heuristic approach [235] can be applied utilising partial knowledge pre-installed before deployment, continuing the operation using reinforcement learning.

There are still many other cognitive approaches worthy of consideration for the enhancement of CogWSN, such as Learning Automata [236] and Evolutionary Algorithms [237]. Those solutions are also able to provide adaptive control. CogWSN can be also applied in various areas of application such as routing, transport, security, resource management and spectrum sensing. All present unique research challenges.

References

1. I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless Sensor Networks: A Survey,” *Computer Networks (Elsevier)*, vol. 38, no. 4, pp. 393-422, Mar. 2002.

2. F. Wang and J. Liu J, “Networked Wireless Sensor Data Collection: Issues, Challenges, and Approaches”, *IEEE Communication Surveys and Tutorials*, vol. 13, no. 4, pp. 673-687, 2011.

3. A. Giorgetti, M. Lucchi, M. Chiani, and M. Z. Win, “Throughput per Pass for Data Aggregation from a Wireless Sensor Network via a UAV,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 47, no. 4, pp. 2610-2626, Oct. 2011.

4. K. H. Kwong, T. T. Wu, H. G. Goh, K. Sasloglou, B. Stephen, I. Glover, C. Shen, W. Du, C. Michie, and I. Andonovic, “Implementation of Herd Management Systems with Wireless Sensor Networks,” *IET Wireless Sensor Systems*, vol. 1, no. 2, pp. 55-65, Jun. 2011.

5. H. G. Goh, Toni, H. Y. Lee, C. F. Leong, C. S. Kuek, S. Y. Liew, and K. H. Kwong, “Practical Implementation of Self-powered Wireless Sensor Networks for Paddy Field Monitoring,” *Proceedings of the IET International Conference on Wireless Communications and Applications (ICWCA)*, Oct. 2012.

6. J. M. L. P. Caldeira, J. J. P. C. Rodrigues, and P. Lorenz, “Toward Ubiquitous Mobility Solutions for Body Sensor Networks on Healthcare,” *IEEE Communications Magazine*, vol. 50, no. 5, pp. 108-115, May 2012.

7. J. Byun, B. Jeon, J. Noh, Y. Kim, and S. Park, “An Intelligent Self-Adjusting Sensor for Smart Home Services Based on ZigBee Communications,” *IEEE Transactions on Consumer Electronics*, vol. 58, no. 3, pp. 794-802, Aug. 2012.

8. D. Caicedo and A. Pandharipande, “Distributed Illumination Control with Local Sensing and Actuation in Networked Lighting Systems”, *IEEE Sensors Journal*, vol. 13, no. 3, pp. 1092-1104, Mar. 2013.

9. V. C. Gungor and G. P. Hancke, “Industrial Wireless Sensor Networks: Challenges, Design Principles, and Technical Approaches,” *IEEE Transactions on Industrial Electronics*, vol. 56, no. 10, pp. 4258-4265, Oct. 2009.

10. J. A. Gutierrez, M. Naeve, E. Callaway, M. Bourgeois, V. Mitter, and B. Heile, “IEEE 802.15.4: A Developing Standard for Low-Power Low-Cost Wireless Personal Area Networks,” *IEEE Network*, vol. 15, no. 5, pp. 12-19, Sep./Oct. 2001.

11. J. L. Hill and D. E. Culler, “Mica: A Wireless Platform for Deeply Embedded Networks,” *IEEE Micro*, vol. 22, no. 6, pp. 12-24, Nov.-Dec. 2002.

12. S. Xiao, A. Dhamdhere, V. Sivaraman, and A. Burdett, “Transmission Power Control in Body Area Sensor Networks for Healthcare Monitoring,” *IEEE Journal on Selected Areas in Communications*, vol. 27, no. 1, pp. 37-48, Jan. 2009.

13. M. Ringwald and K. Romer, “Deployment of Sensor Networks: Problems and Passive Inspection,” *Proceedings of the 5th IEEE Workshop on Intelligent Solutions in Embedded Systems*, Jun. 2007, pp. 179-192.

14. J. Mitola, “Cognitive Radio: An Integrated Agent Architecture for Software Defined Radio,” Ph.D. Dissertation, Royal Institute of Technology (KTH), Sweden, May 2000.

15. R. W. Thomas, “Cognitive Networks,” Ph.D. Dissertation, Virginia Polytechnic Institute and State University, United States, Jun. 2007.

16. O. B. Akan, O. B. Karli O, and O. Ergul, “Cognitive Radio Sensor Networks,” *IEEE Network*, vol. 23, no.4, pp. 34-40, Jul./Aug. 2009.

17. L. Bixio, L. Ciardelli, M. Ottonello, and C. S. Regazzoni, “Distributed Cognitive Sensor Network Approach for Surveillance Applications,” *Proceedings of the 6th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, Sep. 2009, pp. 232-237.

18. K. Shenai and S. Mukhopadhyay, “Cognitive Sensor Networks,” *Proceedings of the 26th International Conference on Microelectronics (MIEL)*, May 2008, pp. 315-320.

19. H. G. Goh, K. H. Kwong, C. Shen, C. Michie, and I. Andonovic, “CogSeNet: A Concept of Cognitive Wireless Sensor Network,” *IEEE Consumer Communications and Networking Conference (CCNC 2010)*, Las Vegas, Nevada, United States, Jan. 2010. (short paper)

20. H. G. Goh, S. Y. Liew, K. H. Kwong, C. Michie, and I. Andonovic, “Cognitive Wireless Sensor Network,” *International Conference on Advanced Infocomm Technology (ICAIT 2010)*, Hainan, China, Jul. 2010. (abstract)

21. K. -L. A. Yau, H. G. Goh, D. Cheng, K. H. Kwong, “Application of Reinforcement Learning to Wireless Sensor Networks: Models and Algorithms,” *Computing (Springer)*, Feb. 2014. (under review)

22. A. Mahapatro and P. Khilar, “Fault Diagnosis in Wireless Sensor Networks: A Survey”, *IEEE Communications Surveys & Tutorials*, 2013. (accepted for inclusion in a future issue of this IEEE journal)

23. Y. Wang, Y. Wang, S. Patel, and D. Patel, "A Layered Reference Model of the Brain (LRMB)," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 36, no. 2, pp. 124-133, 2006.

24. C. -Y. Chong and S. P. Kumar, "Sensor Networks: Evolution, Opportunities, and Challenges," *Proceedings of the IEEE*, vol. 91, no. 8, Aug. 2003, pp. 1247-1256.

25. M. Di Francesco, G. Anastasi, M. Conti, S. K. Das, and V. Neri, "Reliability and Energy-Efficiency in IEEE 802.15.4/ZigBee Sensor Networks: An Adaptive and Cross-Layer Approach," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 8, pp. 1508-1524, 2011.

26. T. M. Mitchell, *Machine Learning (1st Edition)*. McGraw-Hill, Mar. 1997.

27. G. Vijay, E. Bdira, and M. Ibnkahla, "Cognition in Wireless Sensor Networks: A Perspective," *IEEE Sensors Journal*, vol. 11, no. 3, pp. 582-592, Mar. 2011.

28. J. Wang, M. Ghosh, K. Challapali, "Emerging Cognitive Radio Applications: A Survey," *IEEE Communications Magazine*, vol. 49, no. 3, pp. 74-81, Mar. 2011.

29. K. Sohraby, D. Minoli, and T. Znati, *Wireless Sensor Networks: Technology, Protocols, and Applications*. John Wiley & Sons, 2007.

30. University of Rhode Island, Graduate School of Oceanography. (2002-2012). *DOSITS: The Cold War: History of the SOund SUrveillance System (SOSUS)*. [Online] Available: http://www.dosits.org/people/history/SOSUShistory/ [Accessed February 28th 2014].

31. University of Rhode Island, Graduate School of Oceanography. (2002-2012). *DOSITS: How is sound used to navigate underwater?* [Online] Available: http://www.dosits.org/people/navigation/navigateunderwater/ [Accessed February 28th 2014].

32. University of Rhode Island, Graduate School of Oceanography. (2002-2012). *DOSITS: Hydrophone Arrays*. [Online] Available: http://www.dosits.org/technology/basictechnology/hydrophonearrays/ [Accessed February 28th 2014].

33. C. E. Nishimura and D. M. Conlon, "IUSS Dual Use: Monitoring Whales and Earthquakes using SOSUS," *Marine Technology Society Journal*, vol. 27, no. 4, pp. 13-21, 1994.

34. The United States Air Force. (2010, Mar. 29). *Factsheets: Tethered Aerostat Radar System*. [Online] Available: http://www.af.mil/information/factsheets/factsheet.asp?id=3507 [Accessed February 28th 2014].

35. T. W. Nine, "The Future of USAF Airborne Warning and Control: A Conceptual Approach," Research Report, Maxwell Air Force Base, Alabama, Apr. 1999.

36. Federation of American Scientists. (2000, Feb. 24). *Tethered Aerostat Radar System – United States Nuclear Forces*. [Online] Available: http://www.fas.org/nuke/guide/usa/airdef/tars.htm [Accessed February 28th 2014].

37. IEEE. (2012). *Robert Kahn – GHN: IEEE Global History Network*. [Online] Available: http://www.ieeeghn.org/wiki/index.php/Robert_Kahn [Accessed February 28th 2014].

38. S. J. Lukasik, "Why the Arpanet Was Built", *IEEE Annals of the History of Computing*, vol. 33, no. 3, pp. 4-21, 2011.

39. *Proceedings of the Distributed Sensor Nets Workshop*. Pittsburgh, PA: Department of Computer Science, Carnegie Mellon University, 1978.

40. R. Rashid and G. Robertson, "Accent: A Communication Oriented Network Operating System Kernel," *Proceedings of the 8th Symposium Operating System Principles*, 1981, pp. 64-75.

41. C. Myers, A. Oppenheim, R. Davis, and W. Dove, "Knowledge-based Speech Analysis and Enhancement," *International Conference Acoustics, Speech, and Signal Processing*, San Diego, CA, 1984.

42. C. Y. Chong, K. C. Chang, and S. Mori, "Distributed Tracking in Distributed Sensor Networks," *The American Control Conference*, Seattle, WA, 1986.

43. C. Y. Chong, S. Mori, and K. C. Chang, "Distributed Multitarget Multisensor Tracking," in *Multitarget Multisensor Tracking: Advanced Applications*, Y. Bar-Shalom, Ed. Norwood, MA: Artech House, 1990, pp. 247-295.

44. R. T. Lacoss, "Distributed Mixed Sensor Aircraft Tracking," *The 6th American Control Conference*, Minneapolis, MN, 1987, pp. 1827-1830.

45. V. R. Lesser and D. D. Corkill, "The Distributed Vehicle Monitoring Testbed: A Tool for Investigating Distributed Problem Solving Networks," *AI Magazine*, vol. 4, no. 3, pp. 15-33, 1983.

46. D. S. Alberts, J. J. Garska, and F. P. Stein. (1999). *Network Centric Warfare: Developing and Leveraging Information Superiority*. [Online] Available: http://www.dodccrp.org/NCW/ncw.html [Accessed February 28th 2014].

47. J. Hopkins, "The Cooperative Engagement Capability," *APL Technical Digest*, vol. 16, no. 4, pp. 377-396, 1995.

48. J. Pike. (1999, Oct. 26). *Advanced Deployable System (ADS)*. [Online] Available: http://www.fas.org/irp/program/collect/ads.htm [Accessed February 28th 2014].

49. J. Pike. (1997, Jun. 21). *Fixed Distributed System (FDS)*. [Online] Available: http://www.fas.org/irp/program/collect/fds.htm [Accessed February 28th 2014].

50. Federation of American Scientists. (2000, Feb. 21). *Remote Battlefield Sensor System (REMBASS)*. [Online] Available: http://www.fas.org/man/dod-101/sys/land/rembass.htm [Accessed February 28th 2014].

51. The United States Marine Corps. (n. a.). *INTEL – Surveillance and Target Identification – Tactical Remote Sensor Systems System-of-Systems*. [Online] Available: http://www.marcorsyscom.usmc.mil/sites/cins/INTEL/SURVEILLANCE%20&%20TARGET%20IDENTIFICATION/TRSS.html [Accessed February 28th 2014].

52. D. A. Koester, K. W. Markus, and M. D. Walters, "MEMS: Small Machines for the Microelectronics Age," *Computer*, vol. 29, no. 1, pp. 93-94, Jan. 1996.

53. R. J. Weber, *Introduction to Microwave Circuits: Radio Frequency and Design Applications*. Wiley-Blackwell, Jan. 2001.

54. S. Kumar and D. Shepherd, "SensIT: Sensor Information Technology for the Warfighter," *Proceedings of the 4th International Conference on Information Fusion*, TuC1-3–TuC1-9, 2001.

55. J. Corella, "Tactical Automated Security System (TASS): Air Force Expeditionary Security," *SPIE Conference on Unattended Ground Sensor Technologies and Applications*, Orlando, FL, 2003.

56. K. Sohrabi, J. Gao, V. Ailawadhi, and G. J. Pottie, "Protocols for Self-Organization of a Wireless Sensor Network," *IEEE Personal Communications*, vol. 7, no. 5, pp.16-27, Oct. 2000.

57. B. Warneke, M. Last, B. Liebowitz, and K. S. J. Pister, "Smart Dust: Communicating with a Cubic-Millimeter Computer," *Computer*, vol. 34, no. 1, pp. 44-51, Jan. 2001.

58. J. M. Kahn, R. H. Katz, and K. S. J. Pister, "Mobile Networking for Smart Dust," *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'99)*, Seattle, WA, Aug. 1999, pp. 271-278.

59. K. S. J. Pister, J. M. Kahn, and B. E. Boser B. E, "Smart Dust: Wireless Networks of Millimeter-Scale Sensor Nodes," Electronics Research Laboratory Research Summary, 1999.

60. A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, "Wireless Sensor Networks for Habitat Monitoring," *Proceedings of the ACM International Workshop on Wireless Sensor Networks and Applications (WSNA'02)*, Atlanta, GA, Sep. 2002.

61. Dawson Lab at the University of California at Berkeley. (n. a.). *Dawsonlab's Site*. [Online] Available: http://dawsonlab.synthasite.com/ [Accessed February 28th 2014].

62. U. S. Government Printing Office. (2013, Jul. 11). *eCFR – Code of Federal Regulations*. [Online] Available: http://www.ecfr.gov/cgi-bin/text-idx?c=ecfr&SID=a4df538869b44dc2e55b7bbb7956b4fa&rgn=div8&view=text&node=47:1.0.1.1.18.3.239.1&idno=47 [Accessed February 28th 2014].

63. D. S. James. (n. a.). *Notes on Industrial, Scientific and Medial (ISM) Frequency Band*. [Online] Available: http://www.ofcom.org.uk/static/archive/ra/topics/research/rrac/members/rrac01_19.doc [Accessed February 28th 2014].

64. J. L. Hill, "System Architecture for Wireless Sensor Networks," Ph.D. Dissertation, University of California, Berkeley, United States, 2003.

65. S. Lan, M. Qilong, and J. Du, "Architecture of Wireless Sensor Networks for Environmental Monitoring," *2008 International Workshop on Education Technology and Training & 2008 International Workshop on Geoscience and Remote Sensing*, vol. 1, 2008, pp. 579-582.

66. A. Arora, R. Ramnath, E. Ertin, P. Sinha, S. Bapat, V. Naik, V. Kulathumani, H. Zhang, H. Cao, M. Sridharan, S. Kumar, N. Seddon, C. Anderson, T. Herman, N. Trivedi, M. Nesterenko, R. Shah, S. Kulkami, M. Aramugam, L. Wang, M. Gouda, Y. Choi, D. Culler, P. Dutta, C. Sharp, G. Tolle, M. Grimmer, B. Ferriera, and K. Parker, "ExScal: Elements of an Extreme Scale Wireless Sensor Network," *Proceedings of the 11th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications*, 2005, pp. 102-108.

67. N. M. Freris, H. Kowshik, and P. R. Kumar, "Fundamentals of Large Sensor Networks: Connectivity, Capacity, Clocks, and Computation," *Proceedings of the IEEE*, vol. 98, no. 11, Nov. 2010, pp. 1828-1846.

68. J. Ryckaert, C. Desset, A. Fort, M. Badaroglu, V. De Heyn, P. Wambacq, G. Van der Plas, S. Donnay, B. Van Poucke, and B. Gyselinckx, "Ultra-Wide-Band Transmitter for Low-Power Wireless Body Area Networks: Design and Evaluation," *IEEE Transactions on Circuits and Systems – I: Regular Papers*, vol. 52, no. 12, pp. 2515-2525, Dec. 2005.

69. S. Savazzi, U. Spagnolini, L. Goratti, D. Molteni, M. Latva-aho, and M. Nicoli, "Ultra-Wide Band Sensor Networks in Oil and Gas Explorations," *IEEE Communications Magazine*, vol. 51, no. 4, pp. 150-160, Apr. 2013.

70. T. Kaiser, F. Zhang, and E. Dimitrov, "An Overview of Ultra-Wide-Band Systems with MIMO," *Proceedings of the IEEE*, vol. 97, no. 2, Feb. 2009, pp. 285-312.

71. Texas Instruments Incorporated. (1995-2013). *Proprietary RF – 2.4 GHz – CC2543 – TI.com*. [Online] Available: http://www.ti.com/product/cc2543 [Accessed February 28th 2014].

72. C. Bormann, A. P. Castellani, and Z. Shelby, "CoAP: An Application Protocol for Billions of Tiny Internet Nodes," *IEEE Internet Computing*, vol. 16, no. 2, pp. 62-67, Mar./Apr. 2012.

73. S. Hong, D. Kim, M. Ha, S. Bae, S. J. Park, W. Jung, and J. -E. Kim, "SNAIL: An IP-Based Wireless Sensor Network Approach to the Internet of Things," *IEEE Wireless Communications*, vol. 17, no. 6, pp. 34-42, Dec. 2010.

74. L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Computer Networks*, vol. 54, no. 15, pp. 2787-2805, Oct. 2010.

75. S. Sudevalayam and P. Kulkarni, "Energy Harvesting Sensor Nodes: Survey and Implications," *IEEE Communications Surveys & Tutorials*, vol. 13, no. 3, pp. 443-461, 2011.

76. MEMSIC, Inc. (2013). *MEMSIC, Inc – Wireless Sensor Networks*. [Online] Available: http://www.memsic.com/wireless-sensor-networks/ [Accessed February 28th 2014].

77. Atmel Corporation. (2013). *Atmel Corporation – Microcontrollers, 32-bit, and Touch Solutions*. [Online] Available: http://www.atmel.com/ [Accessed February 28th 2014].

78. Atmel Corporation. (2013). *ATmega 128*. [Online] Available: http://www.atmel.com/devices/atmega128.aspx [Accessed February 28th 2014].

79. Adesto Technologies Corporation, Inc. (2013). *Serial Flash | Adesto Technologies*. [Online] Available: http://www.adestotech.com/products/serial-flash [Accessed February 28th 2014].

80. Texas Instruments Incorporated. (1995-2013). *Analog, Embedded Processing, Semiconductor Company, Texas Instruments – TI.com*. [Online] Available: http://www.ti.com [Accessed February 28th 2014].

81. W. Stallings, *Wireless Communications & Networks (2nd Edition)*. Prentice Hall, Nov. 2004.

82. Texas Instruments Incorporated. (1995-2013). *CC1110 and CC1111 Development Kit – CC1110-CC1111DK – TI Tool Folder*. [Online] Available: http://www.ti.com/tool/cc1110-cc1111dk [Accessed February 28th 2014].

83. J. N. Al-Karaki and A. E. Kamal, "Routing Techniques in Wireless Sensor Networks: A Survey," *IEEE Wireless Communications*, vol. 11, no. 6, pp. 6-28, Dec. 2004.

84. F. Hu, Y. Xiao, and Q. Hao, "Congestion-Aware, Loss-Resilient Bio-Monitoring Sensor Networking for Mobile Health Applications," *IEEE Journal on Selected Areas in Communications*, vol. 27, no. 4, pp. 450-465, May 2009.

85. O. A. Postolache, J. M. D. Pereira, and P. M. B. S, Girao, "Smart Sensors Network for Air Quality Monitoring Applications," *IEEE Transactions on Instrumentation and Measurement*, vol. 58, no. 9, pp. 3253-3262, Sep. 2009.

86. T. Torfs, T. Sterken, S. Brebels, J. Santana, R. van den Hoven, V. Spiering, N. Bertsch, D. Trapani, and D. Zonta, "Low Power Wireless Sensor Network for Building Monitoring," *IEEE Sensors Journal*, vol. 13, no. 3, pp. 909-915, Mar. 2013.

87. N. Bouabdallah, M. E. Rivero-Angeles, and B. Sericola, "Continuous Monitoring Using Event-Driven Reporting for Cluster-Based Wireless Sensor Networks," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 7, pp. 3460-3479, Sep. 2009.

88. J. He, S. Ji, Y. Pan, and Y. Li, "Reliable and Energy Efficient Target Coverage for Wireless Sensor Networks," *Tsinghua Science and Technology*, vol. 16, no. 5, pp. 464-474, Oct. 2011.

89. Y. Liu, L. Ni, and C. Hu, "A Generalized Probabilistic Topology Control for Wireless Sensor Networks," *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 9, pp. 1780-1788, Oct. 2012.

90. Y. Sadi and S. C. Ergen, "Optimal Power Control, Rate Adaptation, and Scheduling for UWB-Based Intravehicular Wireless Sensor Networks," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 1, pp. 219-234, Jan. 2013.

91. T. S. Rappaport, *Wireless Communications: Principles and Practice (2nd Edition)*. Prentice Hall, Jan. 2002.

92. I. Glover and P. Grant, *Digital Communications (3rd Edition)*. Pearson Education Canada, Aug. 2009.

93. A. F. Molisch, *Wireless Communications (2nd Edition)*. Wiley, Dec. 2010.

94. I. Demirkol, C. Ersoy, and F. Alagoz, "MAC Protocols for Wireless Sensor Networks: A Survey," *IEEE Communications Magazine*, vol. 44, no. 1, pp. 115-121, Apr. 2006.

95. P. Huang, L. Xiao, S. Soltani, M. W. Mutka, N. Xi, "The Evolution of MAC Protocols in Wireless Sensor Networks: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 1, pp. 101-120, 2013.

96. W. Ye, J. Heidemann, and D. Estrin, "Medium Access Control with Coordinated Adaptive Sleeping for Wireless Sensor Networks," *IEEE/ACM Transactions on Networking*, vol. 12, no. 3, pp. 493-506, Jun. 2004.

97. W. Ye, J. Heidemann, and D. Estrin, "An Energy-Efficient MAC Protocol for Wireless Sensor Networks," *Proceedings of the IEEE 21st Annual Joint Conference of the Computer and Communications Societies (INFOCOM)*, vol. 3, Jun. 2002, pp. 1567-1576.

98. T. Van Dam and K. Langendoen, "An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks," *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys)*, Nov. 2003, pp. 171-180.

99. V. Rajendran, K. Obraczka, and J. Garcia-Luna-Aceves, "Energy-Efficient Collision-Free Medium Access Control for Wireless Sensor Networks," *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys)*, Nov. 2003, pp. 181-192.

100. J. Polastre, J. Hill, and D. Culler, "Versatile Low Power Media Access for Wireless Sensor Networks," *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys)*, Nov. 2004, pp. 95-107.

101. C. Schurgers, V. Tsiatsis, S. Ganeriwal, and M. Srivastava, "Optimizing Sensor Networks in the Energy-Latency-Density Design Space," *IEEE Transaction on Mobile Computing*, vol. 1, no. 1, pp. 70-80, Jan.-Mar. 2002.

102. I. Rhee, A. Warrier, M. Aia, J. Min, and M. L. Sichitiu, "Z-MAC: A Hybrid MAC for Wireless Sensor Networks," *IEEE/ACM Transactions on Networking*, vol. 16, no. 3, pp. 511-524, Jun. 2008.

103. N. A. Pantazis, S. A. Nikolidakis, and D. D. Vergados, "Energy-Efficient Routing Protocols in Wireless Sensor Networks: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 2, pp. 551-591, 2013.

104. D. Braginsky and D. Estrin, "Rumor Routing Algorithm for Sensor Networks," *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, Atlanta, USA, 2002, pp. 22-31.

105. H. Pucha, S. Das, and Y. Hu, "The Performance Impact of Traffic Patterns on Routing Protocols in Mobile Ad Hoc Networks," *Computer Networks*, vol. 51, no. 12, pp. 3595-3616, 2007.

106. D. B. Johnson and D. A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," *Mobile Computing, Springer US*, pp. 153-181, 1996.

107. C. E. Perkins and E. M. Royer, "Ad-hoc On-demand Distance Vector Routing," *Mobile Computing Systems and Applications*, 1999.

108. W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An Application-Specific Protocol Architecture for Wireless Microsensor Networks," *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 660-670, Oct. 2002.

109. C. -H. Lung and C. Zhou, "Using Hierarchical Agglomerative Clustering in Wireless Sensor Networks: An Energy-Efficient and Flexible Approach," *Ad Hoc Networks*, vol. 8, no. 3, pp. 328-344, May 2010.

110. C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva, "Directed Diffusion for Wireless Sensor Networking," *IEEE/ACM Transactions on Networking*, vol. 11, no. 1, pp. 2-16, Feb. 2003.

111. N. Sadagopan, B. Krishnamachari, and A. Helmy, "Active Query Forwarding in Sensor Networks (ACQUIRE)", *Journal of Ad Hoc Networks, Elsevier*, vol. 3, no. 1, pp. 91-113, Jan. 2005.

112. J. Kulik, W. R. Heinzelman, and H. Balakrishnan, "Negotiation-Based Protocols for Disseminating Information in Wireless Sensor Networks," *Wireless Networks*, vol. 8, no. 2/3, pp. 169-185, Mar.-May. 2002.

113. V. Jolly and S. Latifi, "Comprehensive Study of Routing Management in Wireless Sensor Networks - Part-2," *Proceedings of the International Conference on Wireless Networks*, Las Vegas, Nevada, USA, 2006, pp. 49-62.

114. Y. Yu, R. Govindan, and D. Estrin, "Geographical and Energy Aware Routing: A Recursive Data Dissemination Protocol for Wireless Sensor Networks," UCLA Computer Science Department Technical Report, 2001, pp. 1-11.

115. M. Zimmerling, W. Dargie, and J. M. Reason, "Energy-Efficient Routing in Linear Wireless Sensor Networks," *Proceedings of the 4th IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS)*, Pisa, Italy, 2007, pp. 1-3.

116. M. Chen, V. C. M. Leung, S. Mao, Y. Xiao, and I. Chlamtac, "Hybrid Geographic Routing for Flexible Energy-Delay Tradeoffs," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 9, pp. 4976-4988, Nov. 2009.

117. Y. Xu, J. Heidemann, and D. Estrin, "Geography-Informed Energy Conservation for Ad-Hoc Routing," *Proceedings of the 7th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, 2001, pp. 70-84.

118. M. Chen, S. Gonzalez, and V. C. M. Leung, "Applications and Design Issues for Mobile Agents in Wireless Sensor Networks," *IEEE Wireless Communications*, vol. 14, no. 6, pp. 20-26, Dec. 2007.

119. K. Akkaya and M. Younis, "Energy and QoS Routing for Wireless Sensor Networks," *Cluster Computing*, vol. 8, no. 2, pp. 179-188, 2005.

120. F. Ye, G. Zhong, S. Lu, and L. Zhang, "GRAdient Broadcast: A Robust Data Delivery Protocol for Large Scale Sensor Networks," *Wireless Networks (WINET)*, vol. 11, no. 3, pp. 285-298, 2005.

121. E. Felemban, C. -G. Lee, and E. Ekici, "MMSPEED: Multipath Multi-SPEED Protocol for QoS Guarantee of Reliability and Timeliness in Wireless Sensor Networks," *IEEE Transactions on Mobile Computing*, vol. 5, no. 6, pp. 738-754, Jun. 2006.

122. G. Shen and A Ortega, "Joint Routing and 2D Transform Optimization for Irregular Sensor Network Grids using Wavelet Lifting," *Proceedings of the 7th International Conference on Information Processing in Sensor Networks (IPSN)*, 2008, pp. 183-194.

123. S. Kulkarni, A. Iyer, and C. Rosenberg C, "An Address-light, Integrated MAC and Routing Protocol for Wireless Sensor Networks," *IEEE/ACM Transactions on Networking*, vol. 14, no. 4, pp. 793-806, Aug. 2006.

124. C. Wang, K. Sohraby, B. Li, M. Daneshmand, and Y. Hu, "A Survey of Transport Protocols for Wireless Sensor Networks," *IEEE Network*, vol. 20, no. 3, pp. 34-40, May/Jun. 2006.

125. F. Stann and J. Heidemann, "RMST: Reliable Data Transport in Sensor Networks," *Proceedings of the 1st IEEE International Workshop on Sensor Network Protocols and Applications*, 2003, pp. 102-112.

126. C. -Y. Wan, A. T. Campbell, and L. Krishnamurthy, "Pump-Slowly, Fetch-Quickly (PSFQ): A Reliable Transport Protocol for Sensor Networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 4, pp. 862-872. Apr. 2005.

127. O. B. Akan and I. F. Akyildiz, "Event-to-sink Reliable Transport in Wireless Sensor Networks," *IEEE/ACM Transactions on Networking*, vol. 13, no. 5, pp. 1003-1016, Oct. 2005.

128. F. A. Silva, L. B. Ruiz, T. R. M. Braga, J. M. S. Nogueira, and A. A. F. Loureiro, “Defining a Wireless Sensor Network Management Protocol,” *Proceedings of the 4th Latin American Network Operations and Management Symposium (LANOMS)*, Aug. 2005, pp. 39-50.

129. W. Stallings, *SNMP, SNMPv2, SNMPv3, and RMON 1 and 2*. Addison-Wesley, 1999.

130. W. Chen, N. Jain, and S. Singh, “ANMP: Ad Hoc Network Management Protocol,” *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 8. pp. 1506-1531, Aug. 1999.

131. C. S. Raghavendra, K. M. Sivalingam, and T. Znati, *Wireless Sensor Networks*. Kluwer Academic, Jul. 2004.

132. H. S. AbdelSalam, S. R. Rizvi, and S. Olariu, “Energy-Aware Task Assignment and Data Aggregation Protocols in Wireless Sensor Networks,” *Proceedings of the 6th IEEE Consumer Communications and Networking Conference (CCNC)*, 2009, pp. 1-5.

133. X. Du and F. Lin, “Efficient Energy Management Protocol for Target Tracking Sensor Networks,” *Proceedings of the 9th IFIP/IEEE International Symposium on Integrated Network Management*, 2005, pp. 45-58.

134. T. Landstra, S. Jagannathan, and M. Zawodniok, “Energy-Efficient Hybrid Key Management Protocol for Wireless Sensor Networks,” *International Journal of Network Security*, vol.9, no.2, pp. 121-134, Sep. 2009.

135. S. Persia and D. Cassioli, “IPv4 Wireless Multimedia Sensor Networks,” *Proceedings of the 3rd International Workshop on Software Engineering for Sensor Network Applications (SESENA)*, 2012, pp. 58-63.

136. I. F. Akyildiz and I. H. Kasimoglu, “Wireless Sensor and Actor Networks: Research Challenges,” *Ad Hoc Networks*, vol. 2, no. 4, pp. 351-367, Oct. 2004.

137. A. A. Abbasi, M. F. Younis, and U. A. Baroudi, “Recovering from a Node Failure in Wireless Sensor-Actor Networks with Minimal Topology Changes,” *IEEE Transactions on Vehicular Technology*, vol. 62, no. 1, pp. 256-271, Jan. 2013.

138. M. Gregori and M. Payaro, “Energy-Efficient Transmission for Wireless Energy Harvesting Nodes,” *IEEE Transactions on Wireless Communications*, vol. 12, no. 3, pp. 1244-1254, Mar. 2013.

139. S. Shakkottai, T. S. Rappaport, and P. C. Karlsson, “Cross-layer Design for Wireless Networks,” *IEEE Communications Magazine*, vol. 41, no. 10, pp. 74-80, Oct. 2003.

140. V. Srivastava and M. Motani, “Cross-layer Design: A Survey and The Road Ahead,” *IEEE Communications Magazine*, vol. 43, no. 12, pp. 112-119, Dec. 2005.

141. G. Xylomenos and G. C. Polyzos, “Quality of Service Support over Multi-Service Wireless Internet Links,” *Computer Networks*, vol. 37, no. 5, pp. 601-615, Nov. 2001.

142. L. -A. Larzon, U. Bodin, and O. Schelen, “Hints and Notification [for Wireless Links],” *Proceedings of the IEEE Conference on Wireless Communications and Networking (WCNC)*, vol. 2, Mar. 2002, pp. 635-641.

143. G. Dimic, N. D. Sidiropoulos, and R. Zhang, “Medium Access Control - Physical Cross-Layer Design,” *IEEE Signal Processing Magazine*, vol. 21, no. 5, pp. 40-50, Sep. 2004.

144. T. Elbatt and A. Ephremides, “Joint Scheduling and Power Control for Wireless Ad Hoc Networks,” *IEEE Transactions on Wireless Communications*, vol. 3, no. 1, pp. 74-85, Jan. 2004.

145. L. Tong, V. Naware, and P. Venkitasubramaniam, “Signal Processing in Random Access,” *IEEE Signal Processing Magazine*, vol. 21, no. 5, pp. 29-39, Sep. 2004.

146. Q. Liu, S. Zhou, and G. B. Giannakis, “Cross-Layer Combining of Adaptive Modulation and Coding with Truncated ARQ Over Wireless Links,” *IEEE Transactions on Wireless Communications*, vol. 3, no. 5, pp. 1746-1755, Sep. 2004.

147. V. Kawadia and P. R. Kumar, “A Cautionary Perspective on Cross-layer Design,” *IEEE Wireless Communications*, vol. 12, no. 1, pp. 3-11, Feb. 2005.

148. M. C. Vuran and I. F. Akyildiz, “XLP: A Cross-Layer Protocol for Efficient Communication in Wireless Sensor Networks,” *IEEE Transactions on Mobile Computing*, vol. 9, no. 11, pp. 1578-1591, Nov. 2010.

149. L. Song and D. Hatzinakos, “A Cross-layer Architecture of Wireless Sensor Networks for Target Tracking,” *IEEE/ACM Transactions on Networking*, vol. 15, no. 1, pp. 145-158, Feb. 2007.

150. P. Langley, *Elements of Machine Learning*. Morgan Kaufmann, 1995.

151. S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. Pearson, ISBN-13: 978-0132071482, Apr. 2010.

152. Q. H. Mahmoud, *Cognitive Networks: Towards Self Aware Networks*. Wiley-Blackwell, ISBN-13: 978-0470061961, July 2007.

153. E. Alpaydin, "Supervised Learning," in *Semi-Supervised Learning*. MIT Press, 2010, pp. 21-45.

154. O. Brock, J. Trinkle, and F. Ramos, "Classifying Dynamic Objects: An Unsupervised Learning Approach," in *Robotics: Science and Systems IV*, 2009, pp. 270-277.

155. O. Chapelle, B. Schölkopf, and A. Zien, "Introduction to Semi-Supervised Learning," in *Semi-Supervised Learning*. MIT Press, 2006, pp. 1-12.

156. T. M. Mitchell, S. Mahadevan, and L. I. Steinberg, "LEAP: A Learning Apprentice for VLSI Design," *Proceedings of the 9th International Joint Conference on Artificial Intelligence*, 1985, pp. 573-580.

157. P. Langley, "User Modeling in Adaptive Interfaces," *Proceedings of the 7th International Conference on User Modeling*, 1999, pp. 357-370.

158. A. Cypher, *Watch What I Do: Programming by Demonstration*. MIT Press, Cambridge, MA, 1993.

159. C. Sammut, S. Hurst, D. Kedzier, and D. Michie, "Learning to Fly," *Proceedings of the 9th International Conference on Machine Learning*, 1992, pp. 385-393.

160. J. A. Boyan and M. L. Littman, "Packet Routing in Dynamically Changing Networks: A Reinforcement Learning Approach," *Advances in Neural Information Processing Systems*, vol. 6, pp. 671-678, 1994.

161. D. Sleeman, P. Langley, and T. Mitchell, "Learning from Solution Paths: An Approach to the Credit Assignment Problem," *AI Magazine*, vol. 3, pp. 48-52, 1982.

162. N. S. Flann and T. G. Dietterich, "A Study of Explanation-based Methods for Inductive Learning," *Machine Learning*, vol. 4, pp. 187-226, 1989.

163. G. Drastal, R. Meunier, and S. Raatz, "Error Correction in Constructive Induction," *Proceedings of the 6th International Workshop on Machine Learning*, Jun. 1989, pp. 81-83.

164. G. DeJong, "Toward Robut Real-world Inference: A New Perspective on Explanation-based Learning," *Lecture Notes in Computer Science, Springer Verlag*, vol. 4212, 2006, pp. 102-113.

165. A. Stolcke and S. Omohundro, "Inducing Probabilistic Grammars by Bayesian Model Merging," *Proceedings of the 2nd International Colloquium on Grammatical Inference*, 1994, pp. 106-118.

166. D. Ourston and R. Mooney, "Changing the Rules: A Comprehensive Approach to Theory Refinement," *Proceedings of the 8th National Conference on Artificial Intelligence*, 1990, pp. 815-820.

167. Cambridge University Press. (2013). *Cambridge Free English Dictionary and Thesaurus*. [Online] Available: http://dictionary.cambridge.org [Accessed February 28th 2014].

168. R. W. Thomas, L. A. DaSilva, and A. B. Mackenzie, "Cognitive Networks," *Proceedings of the 1st IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN)*, 2005, pp. 352-360.

169. Y. -C. Liang, K. -C. Chen, G. Y. Li, and P. Mahonen, "Cognitive Radio Networking and Communications: An Overview," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 7, pp. 3386-3407, Sep. 2011.

170. J. Mitola, "The Software Radio Architecture," *IEEE Communications Magazine*, vol. 33, no. 5, pp. 26-38, May 1995.

171. J. Mitola and G. Q. Maguire, "Cognitive Radio: Making Software Radios More Personal," *IEEE Personal Communications*, vol. 6, no. 4, pp. 13-18, Aug. 1999.

172. W. H. W. Tuttlebee, "Software-Defined Radio: Facets of A Developing Technology," *IEEE Personal Communications*, vol. 6, no. 2, pp. 38-44, Apr. 1999.

173. T. Ulversoy, "Software Defined Radio: Challenges and Opportunities," *IEEE Communications Surveys & Tutorials*, vol. 12, no. 4, pp. 531-550, 2010.

174. J. Xiao, R. Q. Hu, Y. Qian, L. Gong, and B. Wang, "Expanding LTE Network Spectrum with Cognitive Radios: From Concept to Implementation", *IEEE Wireless Communications*, vol. 20, no. 2, pp. 12-19, Apr. 2013.

175. A. S. Zahmati, S. Hussain, X. Fernando, and A. Grami A, "Cognitive Wireless Sensor Networks: Emerging Topics and Recent Challenges," *Proceedings of the IEEE Toronto International Conference on Science and Technology for Humanity (TIC-STH)*, 2009, pp. 593-596.

176. G. Vijay, E. Bdira, and M. Ibnkahla, "Cognition in Wireless Sensor Networks: A Perspective," *IEEE Sensors Journal*, vol. 11, no. 3, pp. 582-592, Mar. 2011.

177. O. Akan, O. Karli, and O. Ergul, "Cognitive Radio Sensor Networks," *IEEE Network*, vol. 23, no. 4, pp. 34-40, Jul.-Aug. 2009.

178. S. Men, Y. Zhou, X. Sun, and Z. Zhou, "A Cooperative Spectrum Sensing Scheme in Wireless Cognitive Sensor Network," *Proceedings of the IET International Conference on Communication Technology and Application (ICCTA)*, 2011, pp. 577-581.

179. M. Baumgarten and M. Mulvenna, “Cognitive Sensor Networks: Towards Self-Adapting Ambient Intelligence for Pervasive Healthcare,” *Proceedings of the 5th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth) and Workshop*, 2011, pp. 366-369.

180. F. Aalamifar, G. Vijay, P. A. Khoozani, and M. Ibnkahla, “Cognitive Wireless Sensor Networks for Highway Safety,” *Proceedings of the 1st ACM International Symposium on Design and Analysis of Intelligent Vehicular Networks and Applications (DIVANet)*, Nov. 2011, pp. 55-60.

181. S. Haykin, “Cognitive Radio: Brain-empowered Wireless Communications,” *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 2, pp. 201-220, Feb. 2005.

182. R. W. Thomas, D. H. Friend, L. A. Dasilva, and A. B. Mackenzie, “Cognitive Networks: Adaptation and Learning to Achieve End-to-end Performance Objectives,” *IEEE Communications Magazine*, vol. 44, no. 12, pp. 51-57, Dec. 2006.

183. IEEE-SA. (2013). *IEEE-SA – The IEEE Standards Association - Home*. [Online] Available: http://standards.ieee.org [Accessed February 28th 2014].

184. R. B. Marks, I. C. Gifford, and B. O’Hara, “Standards in IEEE 802 Unleash the Wireless Internet,” *IEEE Microwave Magazine*, vol. 2, no. 2, pp. 46-56, Jun. 2001.

185. “IEEE Standard for Information Technology-Telecommunications and Information Exchange Between Systems-Local and Metropolitan Area Networks-Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications”, IEEE Std 802.11-2012 (Revision of IEEE Std 802.11-2007), pp. 1-2793, Mar. 2012.

186. “IEEE Standard for Telecommunications and Information Exchange Between Systems - LAN/MAN - Specific Requirements - Part 15: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Wireless Personal Area Networks (WPANs)”, IEEE Std 802.15.1-2002, pp. 1-473, Jun. 2002.

187. “IEEE Standard for Local and metropolitan area networks--Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)”, IEEE Std 802.15.4-2011 (Revision of IEEE Std 802.15.4-2006), pp. 1-314, Sep. 2011.

188. “IEEE Standard for Local and metropolitan area networks - Part 15.6: Wireless Body Area Networks”, IEEE Std 802.15.6-2012, pp. 1-271, Feb. 2012.

189. "IEEE Standard for Local and Metropolitan Area Networks--Part 15.7: Short-Range Wireless Optical Communication Using Visible Light", IEEE Std 802.15.7-2011, pp. 1-309, Sep. 2011.

190. "IEEE Standard for Air Interface for Broadband Wireless Access Systems", IEEE Std 802.16-2012 (Revision of IEEE Std 802.16-2009), pp. 1-2542, Aug. 2012.

191. D. Geer, "Users Make a Beeline for ZigBee Sensor Technology," *Computer*, vol. 38, no. 12, pp. 16-19, Dec. 2005.

192. A. Wheeler, "Commercial Applications of Wireless Sensor Networks Using ZigBee," *IEEE Communications Magazine*, vol. 45, no. 4, pp. 70-77, Apr. 2007.

193. J. W. Hui and D. E. Culler, "Extending IP to Low-Power, Wireless Personal Area Networks," *IEEE Internet Computing*, vol. 12, no. 4, pp. 37-45, Jul./Aug. 2008.

194. J. W. Hui and D. E. Culler, "IPv6 in Low-Power Wireless Networks," *Proceedings of the IEEE*, vol. 98, no. 11, pp. 1865-1878, Nov. 2010.

195. G. Mulligan, "The 6LoWPAN Architecture," *Proceedings of the 4th Workshop on Embedded Networked Sensors (EmNets)*, Jun. 2007, pp. 78-82.

196. A. P. Castellani, G. Ministeri, M. Rotoloni, L. Vangelista, and M. Zorzi, "Interoperable and Globally Interconnected Smart Grid Using IPv6 and 6LoWPAN," *Proceedings of the IEEE International Conference on Communications (ICC)*, 2012, pp. 6473-6478.

197. M. Sherman, A. N. Mody, R. Martinez, C. Rodriguez, and R. Reddy, "IEEE Standards Supporting Cognitive Radio and Networks, Dynamic Spectrum Access, and Coexistence," *IEEE Communications Magazine*, vol. 46, no. 7, pp. 72-79, Jul. 2008.

198. C. Stevenson, G. Chouinard, Z. Lei, W. Hu, S. J. Shellhammer, and W. Caldwell, "IEEE 802.22: The First Cognitive Radio Wireless Regional Area Network Standard," *IEEE Communications Magazine*, vol. 47, no. 1, pp. 130-138, Jan. 2009.

199. The Wireless Innovation Forum. (2012). *Software Defined Radio (SDR), Cognitive Radio (CR), and Dynamic Spectrum Access (DSA)*. [Online] Available: http://www.wirelessinnovation.org [Accessed February 28th 2014].

200. ITU. (2013). *ITU Radiocommunication Sector*. [Online] Available: http://www.itu.int/en/ITU-R/pages/default.aspx [Accessed February 28th 2014].

201. R. V. Prasad, P. Pawelczak, J. A. Hoffmeyer, and H. S. Berger, "Cognitive Functionality in Next Generation Wireless Networks: Standardization Efforts," *IEEE Communications Magazine*, vol. 46, no. 4, pp. 72-78, Apr. 2008.

202. G. Pólya, *How to Solve It*. Princeton University Press, ISBN: 0-691-08097-6, 1945.

203. G. Pólya and S. Sloan, *How to Solve It: A New Aspect of Mathmetical Method*. Ishi Press, ISBN: 4-871-87830-9, Jun. 2009.

204. V. Chiew and Y. Wang, "Formal Description of the Cognitive Process of Problem Solving," *Proceedings of the 3rd IEEE International Conference on Cognitive Informatics (ICCI)*, 2004, pp. 74-83.

205. K. Sasloglou, I. A. Glover, H. G. Goh, K H. Kwong, M. P. Gilroy, C. Tachtatzis, C. Michie, and I. Andonovic, "Antenna and Base-station Diversity for WSN Livestock Monitoring," *Wireless Sensor Network, Scientific Research Publishing*, vol. 1, no. 5, pp. 383-396, Dec. 2009.

206. H. G. Goh, K .H. Kwong, C. Michie, and I. Andonovic, "Performance Evaluation of Priority Packet for Wireless Sensor Network," *Proceedings of the 2nd IARIA International Conference on Sensor Technologies and Applications (SENSORCOMM)*, Cap Esterel, France, Aug. 2008.

207. MEMSIC, Inc. (2013). *MPR2400CB - 2.4GHz MICAz Processor Board Data Sheet*. [Online] Available: http://www.memsic.com/userfiles/files/Datasheets/WSN/micaz_datasheet-t.pdf [Accessed February 28th 2014].

208. MEMSIC, Inc. (2013). *MEMSIC, Inc - Wireless Sensor Networks*. [Online] Available: http://www.memsic.com/wireless-sensor-networks/ [Accessed February 28th 2014].

209. H. G. Goh, S. Y. Liew, K. H. Kwong, C. Michie, and I. Andonovic, "Abstract Reporting and Reformation Schemes for Wireless Sensor Networks," *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering (LNICST)*, vol. 72, pp. 69-74, Aug. 2011.

210. B. Simmons. (Aug. 2012). *Mathwords: Arithmetic Sequence*. [Online] Available: http://www.mathwords.com/a/arithmetic_sequence.htm [Accessed February 28th 2014].

211. B. Simmons. (Aug. 2012). *Mathwords: Geometric Sequence*. [Online] Available: http://www.mathwords.com/g/geometric_sequence.htm [Accessed February 28th 2014].

212. S. Madden, J. Hellerstein, and W. Hong. (Sep. 2003). *TinyDB: In-Network Query Processing in TinyOS*. [Online] Available:

http://telegraph.cs.berkeley.edu/tinydb/tinydb.pdf [Accessed February 28th 2014].

213. D. Wang, J. Liu, and J. Xu, "Data Sweeper: A Proactive Filtering Framework for Error-Bounded Sensor Data Collection," *Proceedings of the 18th International Workshop on Quality of Service (IWQoS)*, 2010, pp. 1-2.

214. X. Yu, K. Niyogi, S. Mehrotra, and N. Venkatasubramanian, "Adaptive Middleware for Distributed Sensor Environments," *IEEE Distributed Systems Online*, vol. 4, no. 5, May 2003.

215. Crossbow Technology, Inc, "MTS/MDA Sensor Board Users Manual, Revision A", Crossbow User's Manual, Jun. 2007.

216. Toni, H. G. Goh, and S. Y. Liew, "Performance Study of Zeroth-, First- and Second- order Data Abstraction and Reformation Algorithms for Wireless Sensor Networks," *Proceedings of the IET International Conference on Wireless Communications and Applications (ICWCA)*, Oct. 2012.

217. Crossbow Technology, Inc, "MPR-MIB Users Manual, Revision A", Crossbow User's Manual, Jun. 2007.

218. K. H. Kwong, T. T. Wu, H. G. Goh, K. Sasloglou, B. Stephen, I. Glover, C. Shen, W. Du, C. Michie, and I. Andonovic, "Practical Considerations for Wireless Sensor Networks in Cattle Monitoring Applications," *Computers and Electronics in Agriculture*, vol. 81, pp. 33-44, Feb. 2012.

219. C. F. Eick and N. N. Mehta, "Decision Making Involving Imperfect Knowledge," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 23, no. 3, pp. 840-851, May/Jun. 1993.

220. W. Grimson and R. Patil, "Knowledge-Based Systems: The View in 1986," *AI in the 1980s and Beyond: An MIT Survey*, 1989, pp. 13-41.

221. C. Clancy, J. Hecker, E. Stuntebeck, and T. O'Shea, "Applications of Machine Learning to Cognitive Radio Networks," *IEEE Wireless Communications*, vol. 14, no. 4, pp. 47-52, Aug. 2007.

222. R. Haider, M. Y. Javed, and N. S. Khattak, "EAGR: Energy Aware Greedy Routing in Sensor Networks," *Future Generation Communication and Networking*, vol. 2, 2007.

223. M. Shamaiah, S. Banerjee, and H. Vikalo, "Greedy Sensor Selection under Channel Uncertainty," *IEEE Wireless Communications Letters*, vol. 1, no. 4, pp. 376-379, Aug. 2012.

224. M. Cinque, D. Cotroneo, and A. Pecchia, "Event Logs for the Analysis of Software Failures: A Rule-Based Approach," *IEEE Transactions on Software Engineering*, vol. 39, no. 6, pp. 806-821, Jun. 2013.

225. N. D. Tripathi, J. H. Reed, and H. F. VanLandinoham, "Handoff in Cellular Systems," *IEEE Personal Communications*, vol. 5, no. 6, pp. 26-37, Dec. 1998.

226. L. Cheng, X. Zhang, and A. G. Bourgeois, "GTS Allocation Scheme Revisited," *Electronics Letters*, vol. 43, no. 18, pp. 1005-1006, Aug. 2007.

227. B. Kröse and P. van der Smagt, *An Introduction to Neural Networks*. University of Amsterdam, 1996.

228. J. Bemley, "Neural Network and the XOR Problem," *Proceedings of International Joint Conference on Neural Networks (IJCNN)*, vol. 1, 2001, pp. 540-543.

229. K. Gnana Sheela and S. N. Deepa, "Review on Methods to Fix Number of Hidden Neurons in Neural Networks," *Mathematical Problems in Engineering*, vol. 2013, Article ID 425740, 11 pages, 2013.

230. Nii O. Attoh-Okine, "Analysis of Learning Rate and Momentum Term in Backpropagation Neural Network Algorithm Trained to Predict Pavement Performance," *Advances in Engineering Software*, vol. 30, no. 4, pp. 291-302, Apr. 1999.

231. R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction (Adaptive Computation and Machine Learning)*. A Bradford Book, ISBN: 0-262-19398-1, Mar. 1998.

232. M. Wiering and M. Otterlo, *Reinforcement Learning: State-of-the-Art (Adaptation, Learning, and Optimization)*. Springer, ISBN: 3-642-27644-X, Mar. 2012.

233. K. -L. A. Yau, P. Komisarczuk, and P. D. Teal, "Reinforcement Learning for Context Awareness and Intelligence in Wireless Networks," *Elsevier Journal of Network and Computer Applications*, vol. 35, no. 1, pp. 253-267, 2012.

234. T. -C. Huang, P. -H. Lu, and H. -Y. Chi, "Time-slot Allocation for Increasing the Energy-efficient of Multi-hop Layered WSN," *Proceedings of the IEEE International Conference on Intelligent Computing and Integrated Systems (ICISS)*, 2010, pp. 836-839.

235. V. U. Cetina, "Supervised Reinforcement Learning using Behavior Models," *Proceedings of the Sixth International Conference on Machine Learning and Applications (ICMLA)*, 2007, pp. 336-341.

236. M. Thathachar and P. S. Sastry, “Varieties of Learning Automata: An Overview,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 32, no. 6, pp. 711-722, Dec. 2002.

237. R. C. Barros, M. P. Basgalupp, A. C. P. L. F. de Carvalho, and A. A. Freitas, “A Survey of Evolutionary Algorithms for Decision-Tree Induction,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 42, no. 3, pp. 291-312, May 2012.