SPRINGER BRIEFS IN
APPLIED SCIENCES AND TECHNOLOGY

Jameel Ahmed
Mohammed Yakoob Siyal
Muhammad Tayyab
Menaa Nawaz

RFID-WSN Integrated Architecture for Energy and Delay-Aware Routing

A Simulation Approach

Springer

SpringerBriefs in Applied Sciences and Technology

More information about this series at http://www.springer.com/series/8884

Jameel Ahmed · Mohammed Yakoob Siyal
Muhammad Tayyab · Menaa Nawaz

RFID-WSN Integrated Architecture for Energy and Delay-Aware Routing

A Simulation Approach

Springer

Jameel Ahmed
Department of Electrical Engineering
HITEC University
Taxila-Cantt
Pakistan

Mohammed Yakoob Siyal
School of Electrical and Electronic Engineering
Nanyang Technological University
Singapore
Singapore

Muhammad Tayyab
Department of Electrical Engineering
HITEC University
Taxila-Cantt
Pakistan

Menaa Nawaz
Department of Electrical Engineering
HITEC University
Taxila-Cantt
Pakistan

ISSN 2191-530X ISSN 2191-5318 (electronic)
SpringerBriefs in Applied Sciences and Technology
ISBN 978-981-287-413-9 ISBN 978-981-287-414-6 (eBook)
DOI 10.1007/978-981-287-414-6

Library of Congress Control Number: 2015934437

Springer Singapore Heidelberg New York Dordrecht London

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made.

Printed on acid-free paper

Springer Science+Business Media Singapore Pte Ltd. is part of Springer Science+Business Media (www.springer.com)

Preface

Radio frequency identification (RFID) and Wireless sensor networks (WSN) are the two key wireless technologies that have diversified applications in the present and the upcoming systems in this area. RFID is a wireless automated recognition technology which is primarily used to recognize objects or to follow their position without providing any sign about the physical form of the substance. On the other hand, WSN not only offers information about the state of the substance and environment but also enables multi-hop wireless communications. The integration of the promising technologies of RFID and WSN increases their overall functionality and capability and gives a novel outlook to a wide variety of useful applications. As per the literature survey, a need of the simulator for the integrated environment of RFID and WSN was felt and thus presented in this book with its true perspective.

The resource-constrained nature of sensor network impelled the respective research community to address various challenges in its design and operations that degrade its performance. On the other hand, protocols and varying number of applications having different limitations in their nature make it further challenging for such resource-constrained networks to attain application expectations. These challenges appear at various layers of OSI model starting from physical layer up to application layer. At routing-layer, routing protocols are mainly concerned with sensor network operation. Various performance parameters are applied to smart node in order to analyze and optimize querying protocols.

This book is focused on identifying the performance challenges of WSN and RFID analyzing their impact on the performance of routing protocols. For this purpose, a thorough literature survey is performed to identify the issues affecting the routing protocols performance. Then to validate the impact of identified challenges from the literature, a mathematical model is presented to calculate the end-to-end delays of a routing protocol ACQUIRE and a comparison is provided between two routing protocols (ACQUIRE and DIRECTED DIFFUSION) for evaluation. On the basis of achieved results and the literature review, recommendations are made for better selection of protocol regarding the application nature in the presence of considered challenges. In addition, this book also covers a proposed simulator that integrates both RFID and WSN technologies. Therefore, the manuscript is divided into two major parts: integrated architecture of smart node and power-optimized protocol for query and information interchange. An application scenario has been run in order to test the functionality of RFID and sensor networks. Further, for better understanding, an integrated RFID tag with a sensor node has been communicated with the base station/end system.

This book consists of six chapters. Chapter 1 introduces the RFID and WSN technologies, their evolution, and the main differences between them.

Chapter 2 elaborates the two technologies in terms of their components, memory hierarchy, characteristics of routing protocols, and their techniques.

Chapter 3 highlights the challenges and issues in the area of RFID and WSN. It also underlines the contemporary research carried out in this field and what is expected in the near future.

Chapter 4 is all about the delay model for ACQUIRE, its energy estimation, and analysis.

Chapter 5 explains the simulator for smart node in which proposed solutions to existing challenges are addressed with reasonable results.

Finally, Chap. 6 is based on the simulation-based case study and analysis of network model and simulation model. Analysis of case study simulation has been carried out, and subsequently, recommendations are made on the basis of simulated results.

Contents

Abbreviations

ACQUIRE	ACtiveQUery forwarding InsensoRnEtworks Protocol
ADC	Analog to Digital Convertor
AODV	Adhoc On-demand Distance Vector
BAP	Battery Assisted Passive
BER	Bit-Error-Rate
BFSA	Basic Frame Slot Aloha
CLT	Central Limit Theorem
CPU	Central Processing Unit
DARPA	Defense Advanced Research Project Agency
DSDV	Dynamic Source Distance Vector
DSR	Dynamic Source Routing
ELIMA	Environmental Life Cycle Information Management and Acquisition
EPC	Electronic Product Code
EWMA	Exponential Weighted Moving Average
FPGA	Field Programmable Gate Array
GAF	Geographical Adaptive Fidelity
GPS	Global Positioning System
GSM	Global System for Mobile communication
GUI	Graphical User Interface
IFF	Identification: Friend or Foe
ISM	Industrial Scientific and Medical
LEACH	Low Energy Adaptive Clustering Hierarchical Protocol
LLC	Logic Link Control
MAC	Media Access Control
NPDA	Non-deterministic Pushdown Automata
OSI	Open System Interconnection
PDF	Probability Density Function
QoS	Quality of Service
RFID	Radio Frequency Identification
RV	Random Variable

SE-RFID	Sensor Embedded-RFID
SPIN	Sensor Protocol for Information via. Negotiation
TDMA	Time Division Multiple Access
UCODE	Universal Code
UHF	Ultra High Frequency
WSN	Wireless Sensor Network

Chapter 1
Introduction

Abstract Computing technology is expected to interact with the physical environment through small wireless and communication devices in future. In this regard, Radio Frequency Identification (RFID) and Wireless Sensor Networks (WSN) are the most promising technologies in our daily needs and usage of mobile devices. RFID which has the ability to identify and track the objects with its unique code provides information about the presence or absence of the object. Similarly the WSN make use of sensor nodes together and process information about the physical environment which results in, to calculate the state of the object. Both of these technologies are implied separately in a number of applications like asset monitoring, public transportation, supply chain, controlling building access etc. This chapter highlights the integration of these two technologies and number of useful applications enhancing their capabilities and effectiveness. The process of integration of these two technologies has been taken up in this book to envision the simulation environment of RFID and WSN. This approach will help a lot in understanding the real scenarios of the integration and will highlight some attractive areas for engineering and research community.

Keywords RFID · WSN · MAC · LLC

1.1 Radio Frequency Identification

RFID is a wireless automated identification technology that has the capability to accumulate and recover data through electromagnetic communication using radio frequency readable integrated circuit [1]. This technology is termed under the group Automatic Identification (Auto ID), such as bar code, magnetic stripes, biometrics (voice, finger printing, and retina scanning), smart cards, voice recognition, optical character recognition etc. [2]. But RFID system not only enables the feature of unique identification for tracking objects but also overcomes the challenges of the above mentioned identification system. This is because the

J. Ahmed et al., *RFID-WSN Integrated Architecture for Energy and Delay-Aware Routing*, SpringerBriefs in Applied Sciences and Technology,
DOI 10.1007/978-981-287-414-6_1

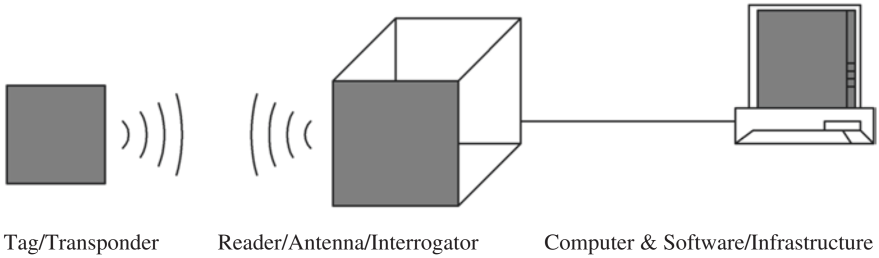

Tag/Transponder
Reader/Antenna/Interrogator
Computer & Software/Infrastructure

Fig. 1.1 Overview of a RFID system [7]

object to be scanned need not to be in line of sight with the reader, preserves tough physical environment, maintains a cost and power-efficient operation, and allows for simultaneous tag identification. An RFID system typically consists of these main parts: A set of tags, a reader, an application host (Fig. 1.1).

The interrogator antenna radiates a field of electromagnetic waves, which serves as a mean of communication between RFID tag and reader. The tag absorbs the radiating energy and power up its microchip to backscatter the signal including the tag unique identification number. The reader then updates a background system about the presence of tagged item in the range which usually runs software that stands between RFID readers and applications. This software is known as RFID middleware.

1.1.1 RFID System Components

Typically RFID systems have three main components: Tag, Reader and Application Host.

1.1.2 Tag

An RFID tag (also referred to as a “transponder, smart tag, smart label, or radio barcode”) has a unique identification number (ID) and memory that is designed to store certain unique information (such as “manufacturer name, product type, and environmental factors including temperature, humidity, etc.”) about the physical object to which the tag is attached, the size of which varies between 32 bits and 32,000 bytes. This tag attached to any physical object can be read and/or written wirelessly with the help of a reader to ascertain its identity, position, or state. The tag consists of a “silicon chip or an integrated circuit” and an antenna. The silicon chip holds an inimitable recognition number and the antenna can launch and take delivery of radio waves. These two components are typically attached to a smooth plastic card which can then be attached to any substantial object. The physical size of a tag can be quite small, thin (like a grain of rice) and can be easily embedded in items like plastic cards, tickets, clothing labels, books etc. [3].

1.1.3 *Reader*

The reader also refers to as interrogator or scanner may have a number of antennas that accounts the process of sending and receiving RF data to and from tags wirelessly [4]. The readers may be deployed stationary or as mobile to notify or energies the tags to “wake it up”.

1.1.4 *Application Host*

The host computer responsible for processing the received data form the reader to obtain some useful information after mapping between IDs and objects via consulting a background database.

1.2 Wireless Sensor Networks

Developments in the field of wireless communication have made possible the advancement of WSN, consisting of many low-cost sensor nodes of low power autonomous devices that are small self-organized, able to collect information, cooperating with each other. They consist of four main components, the CPU for processing data, the battery for the sake of energy, memory for data storage and transceiver to send and receive data between the nodes. The wireless sensor node is also known as “mote”, its size can be application-specific and may vary as per requirement. For example, in applications like military surveillance it could be of microscopic size. Speed factors affect unavoidably the processing cost, size of battery memory [5].

Currently, WSN are extensively applicable in most areas including industrial and commercial benefits. For example, in a military area, WSN can be used for monitoring activity and surveillance. Sensor nodes get activated and start detection, event-triggering, and then by communicating with other nodes in the network, the information to the sink or base station is sent. A sensor network is easily available and reliable system. The intelligence of the network is programmable through the software and due to reliability it can survive for longer period of time, which in long-term is profitable investment.

There are other different types of WSN applications:

- The most common application of WSN is area monitoring. In this scenario, WSN motes are distributed over a region of interest. This scenario is widely used in military for surveillance.
- The advancement of WSN also gives new opportunities in the health system. In the traditional method, the patient should see a doctor at regular intervals and report the symptoms to the doctor by himself. But the smart home care WSN collects data on the basis of specifications and provides continuous medical

record to help diagnosis. This method is also used for emergency medicine and reminder.

- Another area of use of WSN can be agriculture. Many jobs can be done with WSN, like controlling gravity water supply and the pump can be controlled using the device I/O wireless.

1.2.1 Evolution of Sensor Network

Development of wireless sensor networks was originated during Cold War, by the United States [2]. The first WSN was designed and used in the 70s, by the military during the Vietnam War. WSN consists of nodes, from a few to several one, working together to capture data from a region of the environment and send this data to a base station. These sensor nodes are used to track and monitor parameters like heat, temperature, vibratory-motion etc. A network of acoustic sensors placed at strategic locations on the ocean-floor to detect and track Soviet submarines. This system consisted of sound sensors, called the Sound Surveillance System (SOSUS). An important role was played by human operators in these systems. The sensor network was wired network that does not have the limitations of energy bandwidth of a wireless system. Advanced research in the area of sensor networks started around 1980 with the (DSN) program distributed sensor networks Projects Agency Defense Advanced Research (DARPA). This communication includes acoustic sensors (a high-level protocol that link work processes in a common application on a network resource sharing), processing techniques, algorithms (including algorithms for self-localization sensors) and software distributed (dynamically modifiable distributed systems and language design).

1.2.2 Wireless Sensor Network Components

Unlike their predecessors (ad-hoc networks), WSNs have limited resources which are inclined to failure. The number of wireless sensor nodes in a network is superior to a number of ad hoc networks orders having WSN network topology changing continuously. WSN uses media broadcasting within an adhoc network and doesn't provide global identification to avoid traffic collision [6]. Figure 1.2 shows a typical sensor network which comprises:

- *Sensor Node*: Sensor nodes are the heart of the network as they are responsible for data collection and routing of this information to a sink. A sensor is a small device that has a micro-sensor technology, low computing-power and short-range communication capacity. Sensor nodes are conventionally formed by four basic components: a sensor, a processor, a supply/battery and a transceiver [6]. The signal obtained from the location of a sensor is passed to the ADC (Analog

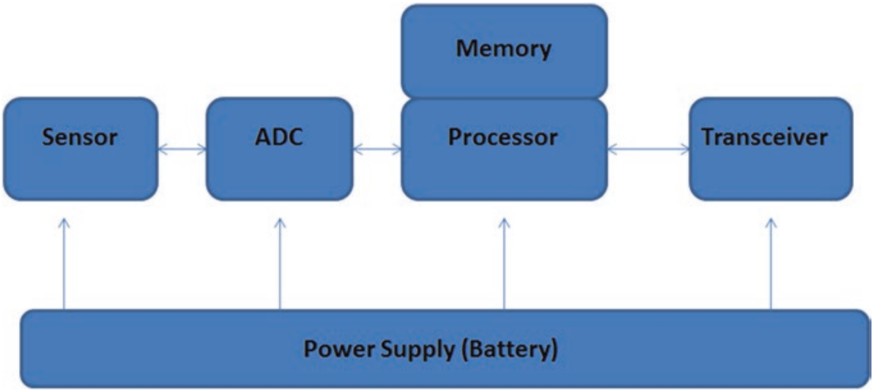


Fig. 1.2 WSN node components

to Digital Convertor) in order to precisely gather the coordinate's information in the system. The ADC digitizes the analog signals measured by the sensors, in turn, fed to the processor. The latter and its associated memory, RAM, is commonly used to manage the procedures of making the sensor node perform its detection and assigned association tasks. The radio transceiver connects to the network node and serves as the mean of node's communication. Memories like EEPROM or Flash is used to store program code. The power supply or battery is the most crucial component of the sensor node as it determines the entire network's lifetime indirectly. Due to the reason that AA batteries or quartz have limited size, the main power sources are cells. To keep it aware of its energy, normal sensor node will spend approximately 4.8 mA signal in order to receive a message, 12 mA, 5 µA when it transmits a packet and sleep respectively [6]. In addition, in active mode the CPU uses an average 5.5 mA current.

- *Sensor Field*: A sensor field can be considered as the area in which the nodes are placed.
- *Base Station*: The central point of control within the network, which extracts information from the network and disseminates controlled information back into the network, is base station. It also serves as a gateway to other networks, a powerful data processing and storage centre and an access point for a human interface. The base station is either a laptop or a workstation.
- *Sink*: A sensor node with the specific task of receiving, processing and storing data from the other sensor nodes, is called sink node. They serve to reduce the total number of messages that need to be sent, hence reducing the overall energy requirements of the network. Sinks are also known as data aggregation points.

Data is flooded to these workstations either via wireless channels, internet or satellite etc. Therefore, hundreds to several thousand nodes are positioned throughout a sensor field in order to create a network that is wireless multi-hop network as shown in Fig. 1.2. Nodes can use wireless communication media such as radio, infrared, Bluetooth or optical media for their communications. The Fig. 1.3 shows

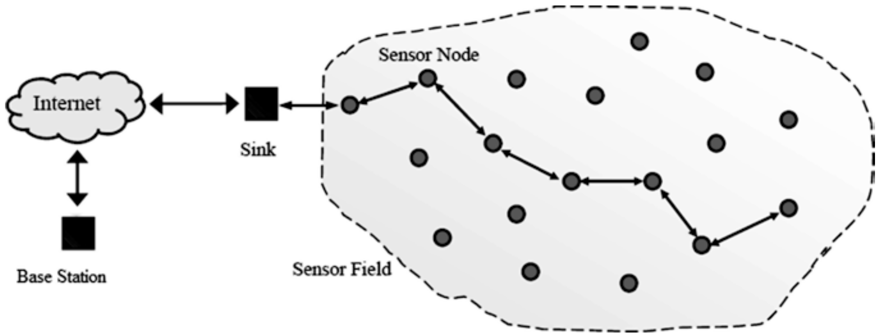


Fig. 1.3 Components of WSN

the variation in transmission range of the nodes occurs according to the communication protocol used.

1.2.3 Architecture of WSN

Sensor network communication follows the basic OSI (Open System Interconnection) layer model. It implements five out of seven layers of the model, which are, application layer, transport layer, network layer, data link layer and physical layer.

1.2.3.1 Physical Layer

This layer is the fundamental layer of network and it consists of networking hardware technologies. This layer works as an electrical and a mechanical interface to the transmission medium. It is responsible for media and signal communication. In OSI architecture, the physical layer translates the logical address that arrives from data link layer, to the hardware specific operations as shown in Fig. 1.4.

1.2.3.2 Data Link Layer

The second layer of OSI model is responsible for physical addressing and it provides functional resources for data broadcasting among networks. It also identifies the errors of physical layer and tries to correct them. The other task of this layer is frame synchronization. The encoding and decoding of data into bits are the main functionality of this layer. OSI Data Link Layer has two sub layers:

- *Logical Link Control* (*LLC*) layer is responsible for frame management and error checking. It provides multiplexing mechanism that makes it possible for several network protocols to transport over the same network medium.

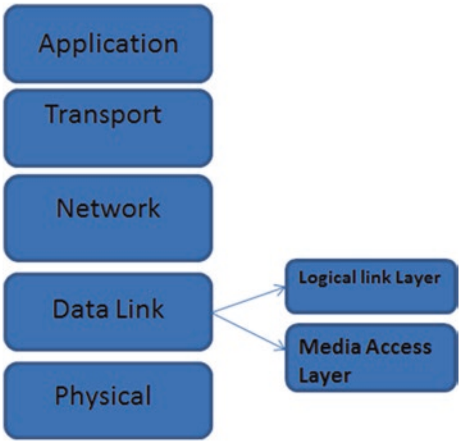


Fig. 1.4 WSN architecture

- *Media Access Control* (*MAC*) is responsible for addressing and channel access control mechanism. It makes possible, for several nodes in a network to communicate within a multiple access network. The media access control is applied when one frame of data ends and the next one starts.

1.2.3.3 Network Layer

OSI Network Layer is used for logically address the communications inside a virtual circuits, so it is used to transmit data from node to node and to determine the path that this data should follow. This layer offers routing and switching technologies. The error handling, packet sequencing, addressing and congestion control are the main functionality of Network layer. It also provides best quality of service on request of transport layer.

1.2.3.4 Transport Layer

The transport layer provides transparent transfer of data and reliable data transfer service to the upper layer. It also provides acknowledgment of the successful data transmission.

1.2.3.5 Application Layer

The OSI model defines application layer as the user interface. It is responsible for displaying data and images to the user in a human-recognizable format, traffic management and provides software for different applications that translate the data in an understandable form.

1.2.3.6 MAC Protocol

Different scheduling modes are defined in WSNs; the sensing node is in active-mode if it is involved in reception and transmission activities, it consumes energy in this state. In idle mode when it is just listening the transmission but does not participate, the consumption of energy is almost same as that in active mode. On the contrary, when it is in sleep mode, the radios are shutdown to save the energy. In this way, the energy can be saved by scheduling the sensor nodes efficiently. Another way of saving the energy is that if the transmission range between the sensor nodes is changed somehow, less transmission range will consume less energy and vice versa. Routing is very important when it comes to network lifetime. Energy can be saved to a larger extent if data is collected and routed efficiently between the nodes i.e. the selected route should contain lesser nodes. Also sensor nodes waste most of their energy when they are overhearing, especially when the data is highly unwanted. So avoiding this situation would make the situation better in the form of increased network life time.

1.2.4 WSN Characteristics

The popularity of cell phones, laptop, PDAs, devices with GPS, RFID, and intelligent computing is continuously increasing, making the things more accessible, more distributed, more mobile, and more pervasive in today’s life. In this scenario, the evolution of wireless sensor networks is basically headed for the compactness and miniaturization of computing devices due to the reason that sensor networks comprise thousands of sensor nodes or motes which are resource constrained and also due to the availability of some resourced base stations. Communication among all the nodes in a network takes place wirelessly.

Furthermore, the usage of WSN is continuously increasing in today’s world but at the same time it is facing the dilemma of energy constraints in the form of limited lifetime of battery resources. The energy essentially needed in transmission of a message is almost two times greater than the energy required in reception of the same message.

As each sensor node is dependent on its energy for all the activities, this has emerged as a major issue in wireless sensor networks. If one of the nodes in the network fails, the whole system gets interrupted. The path taken by each message to reach the destination is termed as “route”. The route plays a very important part in preserving the lifetime of the network, for example, if the routes selected by sensor nodes to reach the destination or the base station, consist of nodes with exhausted batteries may result in decreased network lifetime. On the other hand, if the selected route is longer, consisting of many sensor nodes, the network delay will get increased significantly. WSNs have some other unique characteristics, which include:

- Sensor nodes lie in the category of small-scale devices having their volumes approaching a cubic millimetre in the near future. The energy in such small devices is very limited they can harvest or store energy from the environment.

- In WSN a great degree of dynamics is produced due to factors like mobility of a node, failure of node due to energy constraints, and other environmental obstructions. This also includes numerous changes in network partitions and network topology. Despite partitions, however, mobile nodes are capable of transporting information across partitions by moving between nodes physically.
- The failure in sensor nodes occur due to depletion of batteries or, more generally, due to environmental effects. Limited energy and size of sensor node also typically means restricted resources.
- The paths found as a result of flow of information might have abundant delays and are unidirectional potentially. One of the typical WSN problems is communication failure.

1.2.5 Routing Protocols in WSN

As it is described before, one of the tasks of network layer is the routing. This layer defines the most optimum path that packet should take from source to the destination. Routing algorithm is a logic used to decide for each incoming packet that which output link should be chosen to transmit the data. Routing algorithms can be classified into two groups:

- **Static**: Routing decisions are fixed and nothing can affect on that like traffic load or network topology.
- **Dynamic**: Routing decision depends on network topology and traffic load.

In WSN one of the challenges is the energy consumption problem, since it is not feasible to recharge the limited battery, once it is depleted. So when we want to choose routing algorithm for network, one should take into account to choose the energy-efficient one. There are several routing algorithms which support the idea of energy-efficiency, these will be explained in later chapter.

1.2.6 Energy Aware Routing in WSN

So far, we have got knowingly the importance of battery for the survival of sensor node is that, in most of the cases the replacement of batteries which are drained or depleted out of energy is not desirable. The most important quality of sensor networks is that they are application-specific, which means that the requirements in designing a network vary according to the application. Awareness of the exact location of the sensor node is crucial, as to transfer the data the information about the position of destination node is required. The mechanisms adopted for routing do consider the natural features of WSNs and the architecture and application requirements. The route finding task of WSN and then maintaining it is difficult since the energy is limited and rapid changes in status of node cause unpredictable and frequent changes in topology.

Many routing techniques have been proposed in the field of WSNs to minimize the consumption of energy using some well-known routing policies as well as policies specially designed for WSNs. According to the structure of network, the routing protocols for WSNs can be classified as location based hierarchical and flat routing schemes. In Location-based protocols the information about the position is utilized to transmit the data to the desired area rather than moving it to the whole network. In hierarchical protocols, the nodes are got together in the form of clusters and one node of these clusters is made cluster head which has the responsibility of aggregating the data, reducing the data in order to conserve the energy. Flat networks differ from location based and hierarchical networks as in these types of networks, all nodes play the same role. The various routing protocols proposed for WSNs include Directed Diffusion, SPIN, ACQUIRE, COUGAR, RR, LEACH, TEEN, PEGASIS, APTEEN, GAF and GEAR.

1.3 Why Integration of RFID and WSN

The integration of RFID and WSN technology will enhance the capabilities and functionalities of each other. A broad range of useful advantages can be achieved by merging them into each other because both technologies represent two complementary to each other. Few of them are listed below.

- RFID labels are less costly as compared to sensor nodes; it is reasonable to use RFID tags to swap some of the sensor nodes in WSNs.
- RFID technology can be used to track objects that otherwise are difficult to detect.
- WSN can provide RFID system with a variety of good judgment capabilities to produce intelligent RFID tags.
- In WSN environment, RFID system can be made capable of operating in multihop fashion (Adhoc network) that potentially will extend the applications of this technology.
- Flexible retrieving of information can be achieved by avoiding the wired communication with the base station by make use of portable readers.

1.4 Difference Between RFID and WSN

WSNs are normally deployed to observe objects in areas of interest or to sense environment while RFID systems are used to detect presence or absence of objects that have RFID tags. A summary of both technologies is given in Table 1.1. As both the technologies are equally opposite to each other and the effectiveness of broad range useful applications are just a step ahead with the merging of two technologies.

Table 1.1 WSN versus RFID systems [8]

Attribute	WSNs	RFID systems
Purpose	Sense parameters in environment or provide information on the condition of attached objects’	Detect presence of tagged objects
Component	Sensor nodes, relay nodes, sinks	Tags, readers
Standards	Zigbee	EPC protocol architecture
Communication	Multihop	Single-hop
Mobility	Sensor nodes are usually static	Tags move with attached objects
Power-supply	Battery-powered	Tags are battery-powered or passive
Programmability	Programmable	Usually closed systems
Price	Sensor node—medium Sink—expensive	Reader—expensive Tag—cheap
Deployment	Random or fixed	Fixed, usually requires careful placement
Design goal	WSNs are general-purpose	Tags are optimized to perform a single operation, such as read

References

1. Zhang Y, Yang LT, Chen J (eds) RFID and sensor networks: architectures, protocols, security and integrations, pp 511–536
2. Chong C, Kumar SP (2003) Sensor networks: evolution, opportunities, and challenges. In Proceedings of the IEEE, vol 91, no 8
3. Zhang B, Hu K, Zhu Y (2010) Network architecture and energy analysis of the integration of RFID and wireless sensor network. In: Proceedings of Chinese control and decision conference, p 1381
4. Liu H, Bolic M, Nayak A, Stojmenovic I (2009) Integration of RFID and wireless sensor networks (Chap. 13). Bentham Science Publishers, Sharjah
5. Römer K, Mattern F (2004) The design space of wireless sensor networks. IEEE Wirel Commun
6. Akyildiz IF, Su W, Sankarasubramaniam Y, Cayirci E (2002) A survey on sensor networks. In: Proceedings of IEEE communication magazine
7. http://www.aimglobal.org/technologies/rfid/what_is_rfid.asp. Accessed 20 Apr 2010
8. Thompson D (2006) RFID technical tutorial. J Comput Sci Coll 21(5):8–9

Chapter 2
RFIDs and WSNs

Abstract This chapter presents in-depth knowledge about the two technologies i.e. RFID and WSN. The characterization, working principles and applications of these technologies are discussed in a useful manner.

Keywords Tags · Reader · Routing

2.1 Radio Frequency IDentification (RFID)

The computing technology is expected to interact with the physical environment through small wireless and communication devices in future. In this regard, the RFID (Radio Frequency Identification) is one of the most promising technologies for influencing the real and virtual world. RFID has the ability to identify and track the objects with its unique code and provides information about the presence or absence of the object. RFID technology is implied separately in a number of applications like asset monitoring, public transportation, supply chain, controlling building access etc.

The history of RFID can be tracked as far back as the 1920s with the birth of radar systems (the word radar is an acronym for radio detection and ranging). The development of the technology, a combination of radar and radio broadcast technology, is messy and convoluted but there is consensus that it developed from the work carried out during WW2 to identify enemy aircraft, known as ‘Identification: Friend or Foe’ (IFF) systems [1]. The radio frequency part of RFID is the communication medium between tags and readers. With passive RFID tags, radio frequency is also used to deliver power to the tag, as they do not have on-board power systems [2].

RFID systems are designed to be asymmetric: readers are expensive and power hungry, whilst tags are cheap and require comparatively low levels of energy [3].

J. Ahmed et al., *RFID-WSN Integrated Architecture for Energy and Delay-Aware Routing*, SpringerBriefs in Applied Sciences and Technology,
DOI 10.1007/978-981-287-414-6_2

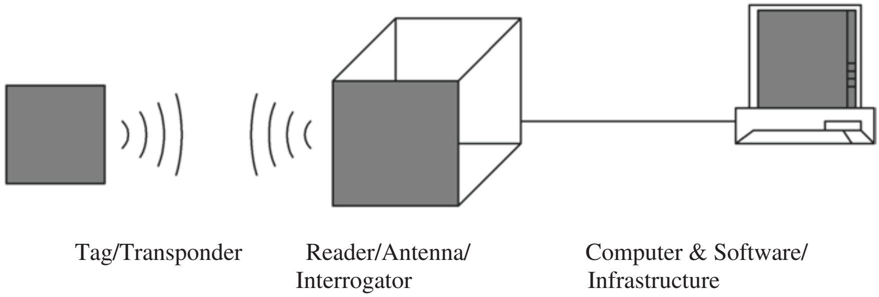

Tag/Transponder
Reader/Antenna/ Interrogator
Computer & Software/ Infrastructure

Fig. 2.1 Overview of a RFID system [4]

In addition, there are three key elements that need to be borne in mind while discussing RFID systems: energy source (which determines if a tag is passive or active), frequency and memory (Fig. 2.1).

2.1.1 Frequency

The RFID operating frequencies band is divided into four major classes: “like Low Frequency (LF) 125–134 kHz, High Frequency (HF) 13.56 MHz, Ultra High Frequency (UHF) 315–433 MHz, 865–956 MHz” and 2.45 GHz and Microwave Frequency 2.45 GHz [5]. Based on these frequency bands, the communication range greatly depends upon factors like “the operating environment, the detail of the antenna design and the available system power” [6]. RFID is fundamentally based on wireless communication, making use of radio waves, which form part of the electromagnetic spectrum (i.e. frequencies from 300 kHz to 3 GHz). RFID operates in unlicensed spectrum space, sometimes referred to as ISM (Industrial, Scientific and Medical) but the exact frequencies that constitute ISM may vary depending on the regulations in different countries (Table 2.1).

2.1.2 Tag-Reader Communication

Tag-reader communication is handled by common procedures often specified in RFID standards such as the “ISO 15693 and ISO 18000-3 for HF or the ISO 18000-6 and EPC for UHF” [7]. Tag-reader communication process is initiated by the reader once it is powered on. The reader broadcasts signals at a special frequency band. The corresponding tags within the reader’s range will absorb the signal energy to power up their internal integrated circuits. The tags then respond to the reader after decoding the signal as valid and continue RF transaction indicating its presence.

Table 2.1 RFID operating frequencies and associated characteristics [1]

Band	Frequency	RFID frequency (typical)	Read range (approx)	Transfer rate of data	Characteristics	Applications
Low frequency	30–300 kHz	125–134 kHz	<0.5 m	Less than 1 kbit/s	Short-range, low data transfer rate, penetrates water but not metal	Animal ID car immobilizer
High frequency	3–30 MHz	13.56 MHz	Up to 1.5 m	Approximately 25 kbit/s	Higher ranges, reasonable data rate (similar to GSM phone), penetrates water but not metal	Smart labels contact-less travel cards access and security
Ultra high frequency	300 MHz–3 GHz	433 MHz or 865–956 MHz 2.45 GHz	Up to 100 m for 433 MHz and 0.5–5 m for 865–956 MHz	For 433–956; 30 kbit/s For 2.45; 100 kbit/s	Long ranges, high data transfer rate, concurrent read of <100 items, cannot penetrate water or metals	Specialist animal tracking
Microwave	2–30 GHz	2.45 GHz	Up to 10 m	Up to 100 kbit/s	Long range, high data transfer rate, cannot penetrate water or metal	Logistics moving vehicle toll

2.2 RFID System Components

An RFID system has two main components: the RF reader (known also as the base-station or interrogator) and the RF tag (or transponder). When RFID tags are attached to physical objects they enable those objects to identify themselves to RFID readers through the use of radio frequency communication [8].

2.2.1 Tag

An RFID tag (also referred to as a “transponder, smart tag, smart label, or radio barcode”) has a unique identification number (ID) and memory that is designed to store certain unique information (such as “manufacturer name, product type, and environmental factors including temperature, humidity, etc”.) about the physical object to which the tag is attached, the size of which varies between 32 bits and 32,000 bytes. This tag attached to any physical object can be read and/or written wirelessly with the help of a reader to ascertain its identity, position, or state. The tag consists of a “silicon chip or an integrated circuit” and an antenna. The silicon chip holds an inimitable recognition number and the antenna can launch and take delivery of radio waves. These two components are typically attached to a smooth plastic card which can then be attached to any substantial object. The physical size of a tag can be quite small, thin (like a grain of rice) and can be easily embedded in items like plastic cards, tickets, clothing labels, books, etc [9].

2.2.2 Reader

The reader also refers to as interrogator or scanner may have a number of antennas that accounts the process of sending and receiving RF data to and from tags wirelessly [10]. The readers may be deployed stationary or as mobile to notify or energies the tags to “wake it up”. The reader is a hand-held or fixed unit that can interrogate nearby RFID tags and obtain their ID numbers using radio frequency (RF) communication (i.e. the process does not require contact). When a passive tag is within range of a reader, the tag’s antenna absorbs the energy being emitted from the reader, directs the energy to ‘fire up’ the integrated circuit on the tag, which then uses the energy to beam back the ID number and any other associated information.

2.3 Types of Tags

RFID tags have been classified into a number of categories based on the power source, memory type and wireless communication signal. Each of these classifications is mentioned below [11].

2.3.1 Tags by the Power Source

Based on power source, Liu et al. [12] classified RFID tags into three major classes: “active tags, passive tags, and semi-passive (semi-active) tags”.

2.3.1.1 Active Tags

These tags contain their own power resource that supplies power to the radio transceiver and on-board circuitry. These tags have more processing power than rest of tags. These tags can communicate with readers in a distance of 100 m or more. They can respond to low power signal form RFID reader than other tags. Due to its advanced processing power, these tags can also be set for incorporated sensors for reporting environmental factors such as temperature, humidity etc. Active tags have a significant amount of memory than passive tags and are best suited for environment where a number of tags need to be read simultaneously. However, these tags are relatively expensive than the passive/semi-passive ones and have a finite battery life which must be replaced periodically [13].

2.3.1.2 Passive Tags

A passive tag does not use any power source of its own. It utilizes the same signal for appending information as a power source and contains a low power integrated circuit. This integrated circuit is attached to an antenna. With the help of this antenna, the tag collects electromagnetic energy from the reader transmitted signal which induces a current in the tag antenna. This current wakes up the tag circuit that reflects a piece of energy reverse to the transceiver adding information to the reflected signal with the help of modulation. These tags have power only when in communication with an RFID reader. Generally, the low power constraint restricts these tags to a short read range up to 3 m or less. This restriction also results in small amount of memory which can store manufacturer unique data in the range of 64 bits. However, these tags have much longer life cycle because they require energy only for its processing operations which is utilized from the received signal. These kinds of tags are cheaper than powered tags because of their nominal involved circuitry. These types of tags are more suitable for applications of individual products such as super markets checkouts and smart cards.

2.3.1.3 Semi-passive Tags

Semi-passive tags are set with energy source to sustain information in the tags or energized some supplementary tasks. This category of tags utilizes the radio waves of source as a power source for their communication like other passive tags. This category of tags has more reliability and larger communicating range than pure passive tags because more power is available for other purposes. Nevertheless, its life cycle gets reduced due to its dependence on battery source and results in an expensive range than other passive tags. Generally, the terms "Semi-passive" and "Semi-active" are used interchangeably in literatures.

2.3.2 Tags by the Memory Type

Another classification of RFID tags is based on memory type, tags with read/write memory, and tags with read-only memory.

2.3.2.1 Read Only Tags

RFID tag with read/only memory is programmed once by the manufacturers and cannot be modified thereafter. As limited size of static information can be stored so these tags are easy to integrate with a data collection system. Usually, these devices are cheaper than others.

2.3.2.2 Read/Write Tags

This class of tags performs both types of memory operations. The information on the memory can be altered dynamically after the manufacturing process. These tags can store larger amount of information (usually in the range of 32–128 kbytes) than Read-only tags but are quite expensive and are not suitable for application of inexpensive-items.

2.3.3 Tags by the Wireless Communication Signal

Another classification of RFID tags is based on the technique of wireless signal used for exchange of information among the two nodes of communication. i.e. (Near field RFID and Far field RFID) [14].

2.4 Routing in WSNs

The process of determining a path from source to destination when a data transmission request is made, called routing. In WSNs, network-layer is the layer which is responsible for routing of all the data. In particular case when the sink node is away or does not lie in the range of source node, multi-hop routing technique is applied and their packets are relayed by intermediate sensor nodes. The solution lies in implementing the routing tables. Routing table comprises lists which have the information about node options, as destination of any packet. It is function of the routing algorithm along with routing protocol's support for their creation and maintenance.

2.4.1 Routing Challenges and Design Issues

Different constraints and objectives have been considered for WSN depending on the different architectures, applications and designs. Though the analysis of the routing-protocol performance is diligently associated with the architectural model yet the considerable limitations are [15]:

- **Network dynamics**: Sensor nodes are supposed to be stationary in maximum sensor networks, because the configurations with mobile sensor nodes are very few in numbers. Sometimes the mobility support of sinks or cluster head is very essential. The other important factor is route optimization stability, in addition to bandwidth, energy constraints etc. The transfer of routing messages between mobile nodes is more challenging. Thus, depending on particular application, the sensed event may be either dynamic or static.
- **Node deployment**: The deployment of node is application specific and performance of routing protocol is highly dependent on it. The deployment of node is either self-organizing or deterministic. In situations where deployment is deterministic, the placement of sensors is done manually and the routing paths for data transmission are predetermined. When nodes are deployed in the self-organizing manner, an ad-hoc infrastructure is created by scattering the nodes. The other factors affecting the performance and energy efficiency is position of the sink or cluster-head. When the nodes are not distributed uniformly, optimal clustering becomes an urgent issue for efficient operation of the power system.
- **Energy considerations**: While an infrastructure is created, setting up process of the routes is greatly under effect of energy constraints. Subsequently, the power of transmission of wireless radio is relative to square of the distance or higher than that sometimes when obstacles are present, lesser energy will be consumed in multi-hop routing than direct communication. Nevertheless, multi-hop routing leads major overhead for medium access control and topology management. The performance of direct routing would be well enough if all the nodes lie close enough to the sink. Sensors are dispersed over an interested area, randomly most of the time, and multi-hop routing becomes inevitable.
- **Data delivery models**: The transmission of data from source to the sink can be uninterrupted, driven by some event, hybrid and driven by some query. It is highly dependent on the application scenario of the sensor network. Considering the uninterrupted model for data delivery, data is sent by each

sensor periodically. In query-driven and event-driven models, thus whenever an even occurs or the sink generates some query, the transmission of data is activated. Sometimes a hybrid model is applied by sensors by combining continuous, query-driven and event-driven delivery of data. The choice of routing protocol used is highly inclined by the model for delivering data, particularly when reduction of route stability and energy consumption is under consideration.

- **Data aggregation/fusion**: Multiple nodes can produce similar types of packets which are aggregated to minimize the transmission. Substantial amount of redundant data is generated by sensor nodes for this purpose. The aggregation of data is the process of data combination from various sources by using different functions.
- **Node capabilities**: In a sensor network, different functionalities can be linked with the sensor nodes. A sensor node is very application specific, thus a node can be devoted to a certain special task such as sensing of data, relaying and collection of data. The usage of these functionalities all at once on a single node might make the energy of that node drained very quickly.

2.4.2 Routing Objectives

There are some of the applications of sensor network which require just the successful data delivery between a source node and a destination node. Nevertheless, there are some of the applications where more assurance is required. These are the requirements in real-time for maximizing the life time of network and for delivering the data [16].

- **Non-real time delivery**: It is essential for all routing protocols to consider the guarantee of data delivery. In clear meanings, if there exists a routing path between the interactive nodes, the routing protocol should always find it. This property of precision can be proven in a way which is more recognized, while the performance evaluating parameter is the data delivery ratio.
- **Real-time delivery**: Some of the sensor network application scenarios deal with in-time data delivery so they require in-time delivery of that message, if failed to do that, the message becomes useless and it might losses informative content after the time limit. Hence, thorough control of network's delay is the key objective of these routing protocols. The time constraint message delivery ratio is the parameter for evaluating the performance of these protocols.
- **Network life Span**: The applications where the sensor nodes in the network must run as long as possible, this objective is crucial. The protocols which are concerned about network lifetime, by considering the remaining energy levels, try to balance the energy consumption among the nodes equally. Though,

like every other sensor node application, the network lifetime parameter is also application dependent. Maximum routing protocols deal with the assumption that every node is important equally and one of the parameter they use is the time, until the first node drains off, and the energy consumed by nodes on average, as another parameter. If nodes are not equally important, then the time until the last or high-priority nodes die can be a reasonable metric [17, 18].

2.4.3 Characteristics of Routing Protocols

Generally, wireless sensor routing protocols are:

- Capable of collecting data.
- Application dependent.
- Data centric.
- Capable of optimizing energy consumption.

2.5 Routing Techniques in Wireless Sensor Networks

WSN Routing Protocols can generally be classified in three ways, according to the way of routing paths are established, the network structure, the protocol operation. Figure 2.2 shows the WSN routing protocols classification.

There can be three ways for the establishment of a routing path, namely proactive, reactive or hybrid. Routing protocols capable of calculating all the routes before they are really needed and then store these routes in a routing table of each node, is proactive routing protocol. Whenever there is a change in routing path, the change must be circulated through the entire network. Since a WSN may consist of several hundreds or thousands of sensor nodes, each sensor node is keeping the routing table, it could be huge in size and therefore proactive protocols [19] are not much suitable for sensor networks. Reactive protocols [19] calculate routes if they are needed. Hybrid protocols use a combination of these two ideas. But in general, considering the network structure, the routing in networks of sensors can be divided into categories described as hierarchical-based routing, flat-based routing and location-based routing. In hierarchy-based routing, however, nodes play different roles in the network. In the flat routing, all nodes have the same role. In the location-based routing, the positions of the sensor nodes are operated for routing data in the network. Furthermore, these protocols can be classified into query-based, multipath-based, QoS-based, negotiation-based or coherent-based routing techniques depending on the protocol operation.

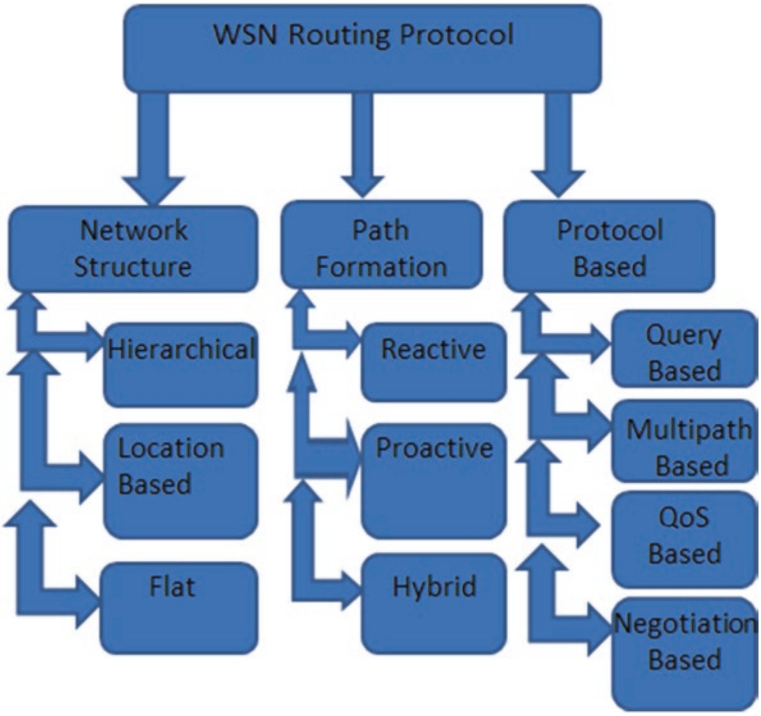


Fig. 2.2 Classification of WSN routing protocols

2.5.1 *Flat Routing*

Assigning individual identifiers to every node in a network is not feasible due to higher density of sensor nodes present in a network for wide variety of applications. Thus random deployment of sensor nodes in a network with no global identifiers makes selection of set of sensor nodes, difficult for a query. Therefore, within a deployment area data transmitted form sensor node is generally in redundant form. This realization helped in emergence of data-centric routing [20].

2.5.2 *Hierarchical Protocols*

Scalability is one of the key design attributes of sensor networks. Due to the reason that the sensors cannot communicate over longer distances, there is no scalable architecture for single gateway sensors. In order to cope with extra load on a network, some approaches have adopted aggregation procedure which maintain the service and enable the network to cover large area of deployed interests.

Hierarchical routing works in two layers, Cluster-heads are selected on the first layer and second layer is used for routing the information. Hierarchical protocols maximize the overall system lifetime, scalability, and energy efficiency [21].

2.5.3 Location-Based Protocols

For geographically deployed sensor networks, exact location information of sensors is very important. It enables the distance calculation between two particular nodes in order to estimate the energy consumption and thus total energy of the network. Typically, two techniques have been used in the literature to find the location, one is by using Global Positioning System (GPS) and other is by finding the coordinates of the nodes in the neighborhood. Due to unavailability of some addressing system for sensor networks such as assigning IP addresses to individual sensors and their spatial deployment in a region, the location information may be accessed which is useful in routing data efficiently with energy consideration [21].

2.5.4 Multipath Routing Protocols

Several ways are used to maximize the performance of a network. On failure of the primary link between the source and the destination, there is another way which measured fault tolerance (resilience) of a protocol. This can be increased by maintaining multiple paths between the source and the destination. This increases the cost of the energy consumption and the generation of traffic. Alternative paths are kept alive by sending periodic messages. Therefore, the reliability can be increased [21].

2.5.5 Query Based Routing Protocols

A REQ (request for data) meta-data has been propagated by destination nodes through the network and a node with requested data returns the data to the node. The forwarded data is then matched with the query to initiative query. Generally, local languages are used to describe the queries, or languages having high-level queries [21].

2.5.6 Negotiation Based Routing Protocols

High-level data descriptors are used eliminate duplicate data transmissions, via negotiation. Communicating decisions are made depending on the available resources. Using flooding for data dissemination for providing implosion and overlapping between the sent data is the motivation behind this. The consumption of energy is increased and more processing is required to send the same data to different sensor nodes. Thus, removal of duplicate information and prevention of redundant data is the main idea of the negotiation based routing in sensor networks [21].

2.5.7 QoS-Based Routing Protocols

Quality of Service (QoS) parameters, such as delay, power, and bandwidth is satisfied upon delivery of data to the base station, the network has a balance between energy consumption and data quality [21].

2.6 Routing Protocols for WSN

One of the most important and efficient routing protocols of WSNs is Directed Diffusion (DD). It is a typical data-centric protocol for WSN which laid the essential foundation for routing design of wireless network and is leading the way in data centric protocol design. In next section DD and another routing protocol ACQUIRE is discussed in detail. ACQUIRE is an efficient query based routing protocol especially designed for energy aware efficient routing in WSNs.

2.6.1 Directed Diffusion Protocol

Direct Diffusion [22] is a data centric protocol. It is the first protocol proposed for wireless sensor networks scenarios that works better than the flooding techniques. It consists of several elements: interests, data messages, gradients, and reinforcements. First, request is sent by the sink node by sending data interest. The message of interest is a query, which specifies that the user wants its neighbors to name the data. Data is named using attribute-value pairs and is information is collected or processed. Interests are flooded throughout the network. This data may be an event which is a brief description of the phenomenon detected. Whenever a node receives an interest, it will check if this interest already exists or it's new. If it is a new interest, the sensor node set up a gradient to the sender to "draw" the data down which corresponds to the interest. Each pair of neighboring nodes establishes a gradient in the other. After creating step gradient, the source node starts sending data corresponding to the related interests of the sink. The data is generally distributed to all its neighbors' gradient. The events are propagated to the initiators of interest along a variety of paths gradient. The sensor networks reinforce some of these paths. The reinforcing pattern is usually designed for the minimum or maximum packet delay received during a certain period of time as shown in Fig. 2.3.

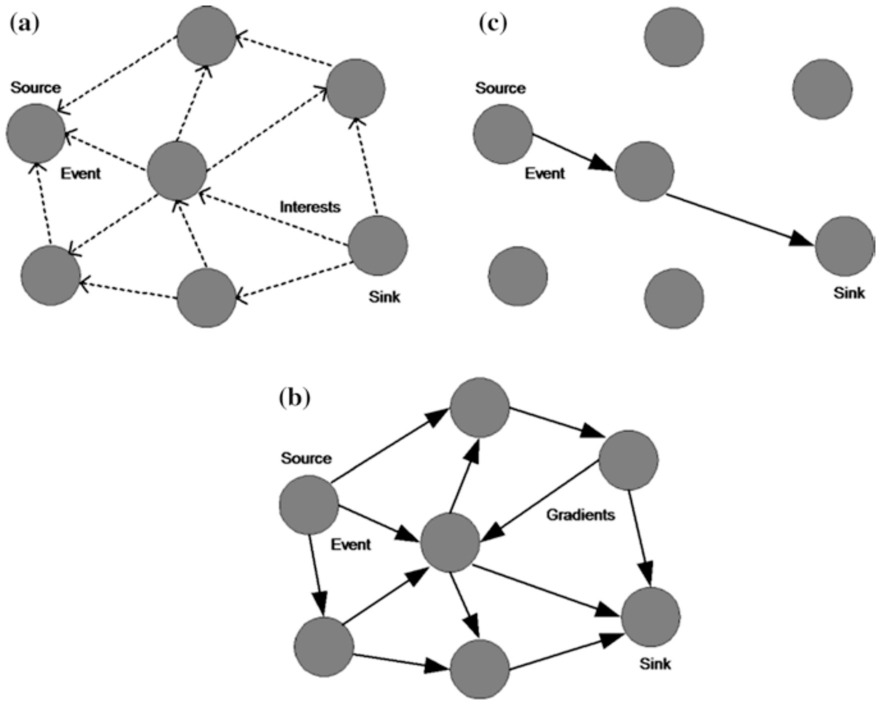


Fig. 2.3 Direct diffusion; **a** propagation of interest, **b** initial gradient setup, **c** data delivery [22]

Directed Diffusion tasks are named using attribute-value pairs. For the naming scheme based on the value attributes, a range of values is associated with each attribute. Some other choices are also available for attribute-value pair arrangement and naming systems. Somehow, the choice of naming scheme and arrangement can affect the performance of diffusion by affecting the appearance of the task.

The interest is usually introduced into the network through sink. For each active task, the sink periodically broadcasts an interest in all of its neighbors. Since in the sensor network it is highly inconvenient to locate the sources accurately, interest must necessarily be distributed over a wider section of the network. Accordingly, if the selected sink has data rate that is initially high, the energy consumption could be high due to the wider distribution of sensor data. The desired higher flow rate data may be obtained by reinforcement.

The interest is trivial state and sink is responsible for updating it periodically. Periodic interest propagated by sinks is necessary because there is a fair chance that it might not be received. To overcome this problem, the same interest is simply re-sent by the sink, thus increasing timestamp attribute. Each node is responsible for maintaining a cache of interests. Each item in the cache is related as a separate interest. Interest contains information about the immediate previous hop only so, the state of interest related to the number of distinct active interests. Few fields are associated with interests in cache. A timestamp field shows the timestamp of the last corresponding interest received. A gradient field contains a data rate requested by the corresponding neighbor range, and the derivative of interest and a duration field attribute and derived from the timestamp expires, to attributes of interest. The duration field indicates the approximate life of the gradient and interest. Data is propagated by gradients. For event-triggered applications, every gradient contains a type gradient instead of a data rate gradient. There are two types of gradient: Exploratory gradient and Data gradient. Exploratory gradients are for the path setup and repair while gradients data for sending actual data. Gradient type by default is exploratory.

On receiving interest, a node first checks its availability in cache. If there is no corresponding interest, the node creates an entry of interest and each field of the entry in the interest is determined. This entry contains a single gradient towards the neighbor whose interest was received, with the rate specified event data. Thus, it is necessary to distinguish individual neighbors. Then the timestamp fields and duration is updated by node. Expired gradient will be removed from the entry of interest, but not all gradients will expire at the same time.

When an interest is received, a node may decide to refer the interest in a subset of its neighbors. To its neighbors, this interest seems to come from the sending node, although a distant well may be the true origin. With such a completely local interaction, interest is distributed across the network. Interests However, all receipts are sent again. Using the cache of interest, a node can delete a received interest if it has recently re-sent a corresponding interest. In general, there are several possible choices for the neighbors to return the interest.

The best alternative in a simple way would be retransmitting the same interest of all neighbors, like flooding the interest. Since interest is flooded, gradients are established by all nodes. Unlike simplified in Fig. 2.3b description, each pair of neighboring nodes establishes a gradient towards each other. Sine there is no information about sink in received interest, when a node receives an interest, node is unable to find out that if the interest is delivered to the node because it is in the form of loop, also that the interest is delivered using a different path, or the same interest is newly generated from another sink. Such bidirectional gradients can cause a node to receive a copy of events of low data rate from each of its neighbors. However, this technique can enable fast recovery paths or reinforcing the empirical failure best ways and do not incur persistent loops.

2.6.2 *Low Energy Adaptive Clustering Hierarchical Protocol (LEACH)*

LEACH protocol is the most widely used protocol [23] and can be described as a combination of multi-hop routing and cluster architecture. The sensors with LEACH protocol work by forming cluster-heads and cluster members among the group. The communication between clusters, cluster-head and base stations are done by multi-hop routing. The operations that are conducted in the protocol LEACH is divided into two phases, the setup phase and the steady state phase.

In the setup phase, all of the sensors within a network get together make groups in some region of the cluster. They communicate with each other by exchanging short messages. One sensor node can act as a cluster head at a time and is capable of communicating to all other remaining sensors. Primarily, similar type of sensors depending on the signal strength of messages sent by the cluster heads chooses to form clusters. Interested sensors meet again at the head of the cluster when they send a signal indicating their acceptance to join as a response. Thus, this is the end of setup phase. The group leader can decide the optimal number of cluster members, it can handle or requires.

Before entering the first phase, parameters such as relative computation cost and network topology are taken into consideration. A TDMA schedule is applied to all members of the cluster group to send messages to the cluster head and cluster head to the base station. Figure 2.4 shows two phases of a sensor in a LEACH protocol, all sensors formed as members of the cluster head and the cluster heads in the second phase of ammunition perform data transmission to the sink in a multi-hop structure. A direct transmission system is also provided below.

As soon as a cluster head is selected for a region, all the sensed and collected data is sent using TDMA slots allotted to the cluster head. Then this data is compressed and transmitted to base station by cluster head, which completes the

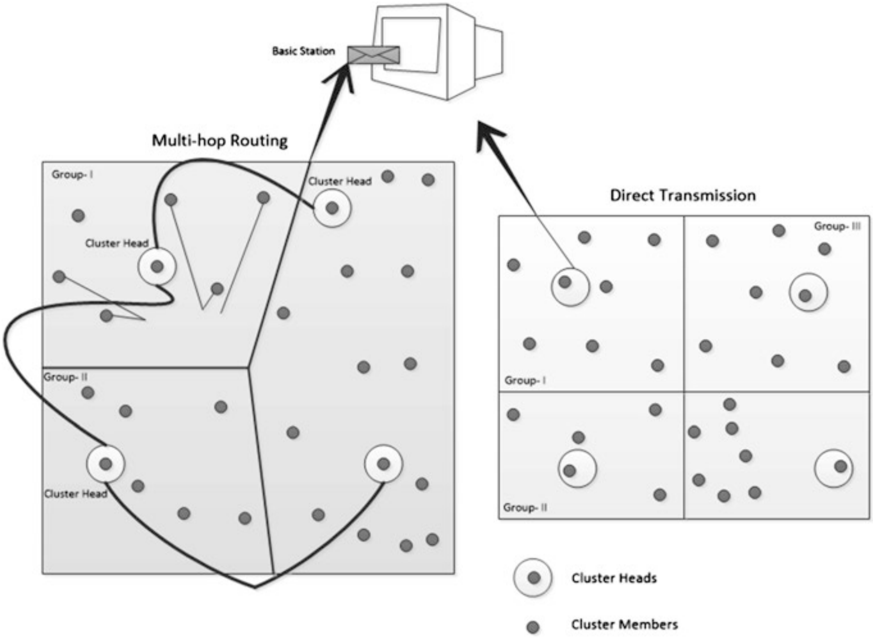


Fig. 2.4 LEACH operation showing set-up, steady state phases using multi-hop, also showing direct transmission [23]

second phase, called the **Steady State Phase**. Once the steady-state finishes the data transmission to the sink, the whole process comes to an end and a new search for the forming of cluster heads for a region and new cluster-member formation begins.

Since the network is distributed, there is a possibility that all the sensors might not be too close or at equal distance to the cluster head, so the cost of energy consumption by the farther sensor is not equal to the cost of energy consumption by the nearest node. In order to minimize this, procedure of cluster heads formation is rotated among all the nodes in the cluster. LEACH minimizes global energy usage by distributing the load of the network to all the nodes or cluster members at different intervals.

The data sent by all the cluster heads in a network is received at base station in compressed format. An important point is that the cluster heads which are away from base station, cannot send their data to base station directly therefore, they forward their data to nearest cluster heads and they eventually forwards the information to the base station, forming a multi-hop routing network. Making cluster head responsible for forwarding the data improves lifetime of network by improving the lifetime of sensors [24].

2.6.3 Sensor Protocol for Information via Negotiation (SPIN)

Distribution of data became the main motivation behind developing SPIN. Dissemination or distribution of data is the process of collecting the observations from all the deployed sensors in a network, treating every sensor as sink node. Life time of the sensors is made prolonged by sensibly controlling consumption of energy during computation and communication. SPIN was proved better protocol by removing drawbacks in the sophisticated protocols like: overlapping, implosion and resource blindness [23]. These issues of flooding and resource blindness were solved by using negotiations and resource adaptation.

SPIN uses three types of messages for transmitting data, also known as Meta data before data transmission among the neighbors in the network. First the interest is propagated when a node sends this meta-data to its neighboring node. Since information is exchanged by using meta-data, it helps the node to choose particular type of data from particular node, saving the energy. While sharing data directly between two nodes might result in greater energy consumption. Nodes cannot respond to all the data messages sent by all other nodes because of the changing topology or network structure.

Characteristics of shared information are described in meta-data. The size of meta-data is made smaller than the original data and it should be distinct from other data types. Meta-data are application-specific and they always consider their geographic location or a unique ID while communicating with the neighbor nodes. Three types of meta-data messages are exchanged between the nodes:

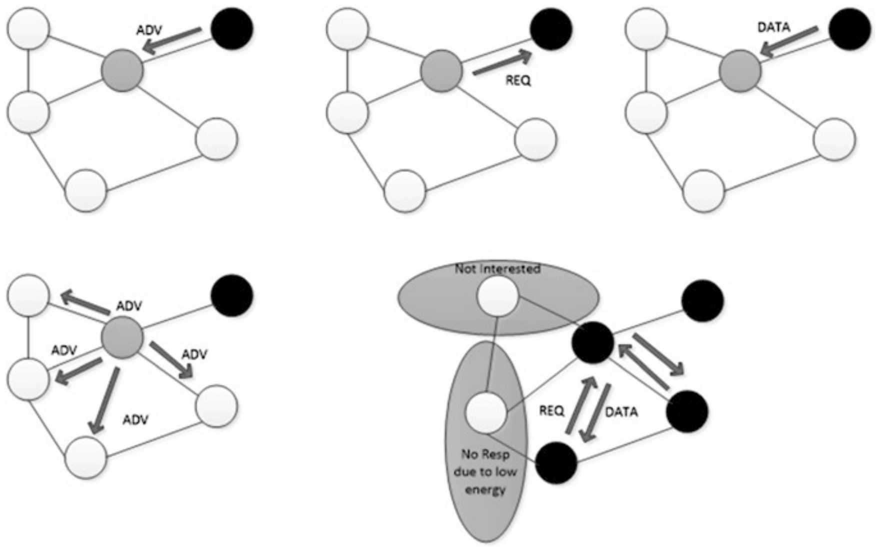


Fig. 2.5 Five stages of SPIN showing the three-way handshaking [23]

ADV: This type of meta-data is used when a SPIN node has something new to exchange it with other neighbouring nodes in the network. So, this is advertisement meta-data.
REQ: This is the request meta-data and any node interested in exchanging new information, sent this type of message to the node having this new interest or information.
DATA: The actual data that has to be shared between two nodes which are involved in exchange.

Considering the scenario that a node A has gone through update and it needs to forward this new information to other neighboring nodes in a network. For this purpose first message it send is an ADV message to the nearest neighbors, containing required fields describing data type computation and communication type etc. Only interested nodes will respond in this updated information will send REQ message to source node as a response. On receiving REQ message, the source note transmits the requested information to interested node. This is how the data dissimilates (Fig. 2.5).

2.6.4 ACtive QUery Forwarding In sensoR nEtworks Protocol (ACQUIRE)

ACQUIRE [23] is a data centric, query-based protocol. Recently ACQUIRE has appeared as a technique that is energy efficient and highly scalable query-based protocol for use in real-world sensor networks. The motivation behind ACQUIRE is the injection of active queries into the network that are capable of triggering when local updates are performed similar to COUGAR [25], it considers the network as a distributed database. The query is sent by the sink and each node that receives the query by processing the existing information. After that it forwards the query to a neighboring sensor. If the existing information in the node needs to be updated, the node looks for the information from the neighbors who are at most *d* hops far away.

ACQUIRE, when comes to compare with other alternative routing protocols seems to beats all other strategies, keeping its parameters values optimum. The reason behind that performance would be that it is especially designed for complex, one-shot queries, even when the other schemes too are enhanced with cached updates. In particular, optimal ACQUIRE performs many orders of magnitude better than flooding-based schemes (such as Directed Diffusion) for such queries in large networks. Energy consumption can be reduced to 60 % by using ACQUIRE with optimum settings and parameters (Fig. 2.6).

The traditional flooding-based techniques must be taken as overview in order to know mechanism of ACQUIRE better. Two main stages are clearly separated in those techniques, which are response gathering stage and dissemination stage. Several copies of queries in the form of interests for named data are flooded in the network first. Nodes carrying data relevant to those queries respond. For non-persistent queries, the costs lined with queries are dominated by flooding. Similarly, in the process of aggregation data is collected in suboptimal way as a result of duplicate responses, causing increased energy cost.

On the other hand, query and response stages are not separated in ACQUIRE. The querier node generates an active, one shot, rather complex query, carrying

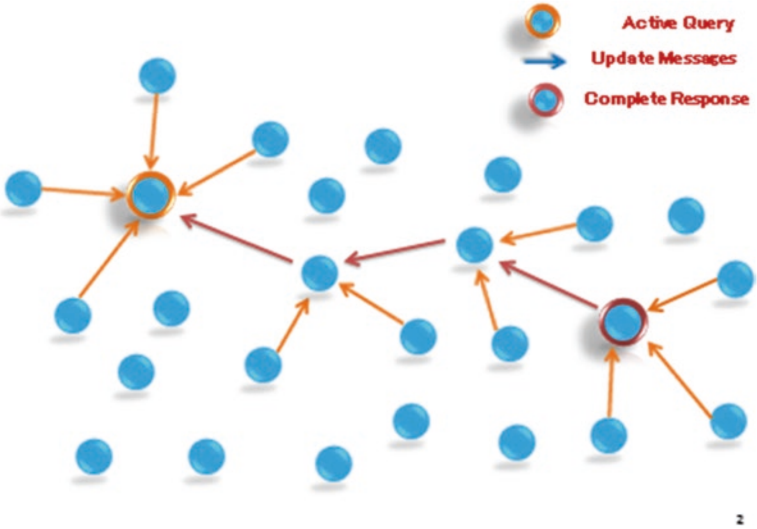


Fig. 2.6 Working of ACQUIRE protocol

multiple sub-queries capable of corresponding to different interests. The query generated is then forwarded to other neighboring nodes following certain sequence. The updates received by all nodes are then utilized by active node at each intermediate step, within *d* hop distance in order to resolve it partially. Whenever a node receives an active query, it triggers for updates if it is holding old/expired information. First active node tries to resolve the query partially with information it is already holding, then it forwards the remaining unresolved query to another node randomly selected within *d* hop distance. The choice of node can be either way by random walk or directed intelligently based on other information. Hence as the active complex query moves through the network, it keeps getting "smaller" as pieces of it become resolved gradually, until it reaches a node which has every information to resolve it completely. The forwarded query becomes a complete response at this point where it is resolved completely and it is forwarded back to source node as complete rote.

ACQUIRE protocol is therefore well-suited for complex, one-shot queries for replicated data. When it comes to discuss persistent queries, flooding based techniques can work better than ACQUIRE because of their low cost of interest flooding rate.

2.6.5 Geographical Adaptive Fidelity Protocol (GAF)

It locates network nodes and makes the best use of them for better fidelity. All nodes use a location-identification technique lie within the network, together with its nearest neighbors by using information location systems like GPS. In GAF, all nodes are commanded according to the networks also called the virtual networks. All network nodes are divided into the virtual networks and all nodes that are under the same grid coordinate with each other to see who will enter a state of sleep and for how long. Load balancing is performed and a single node will not get drawn with the work of others. It can also be very simple to define virtual networks and all the nodes in the window A can communicate with all nodes in Bare adjacent window. Sleep time decided or information depends on the application and system.

There are three state variances in GAF, i.e. discovery, active and sleep. At start all nodes begin with the discovery state. Radio of the node is turned on and starts sending discovery messages to find the adjacent its nodes in same network. Each discovery message is a collection of certain parameters, e.g:

Node State: Discovery, Active or Sleeping
Node ID: The node itself or its current location
Grid ID: Each node in a network utilizes its location information gathered from GPS and the size of its grid in order to query its grid id
(*enat*): Estimated node active time, this is the node lifetime. This is the total remaining energy in the node.

These parameters such as node move to discovery mode, which sets the time Td (Time Discovery) and send discovery messages to all its neighbors in the network. After transmitting a message to this discovery, it enters the active state. This may include sleep mode if there are other nodes in the network, corresponding to the processing of fidelity before falling in the active state. In active mode, the node sets the timeout value Ta showing the remaining time that the node is intended to remain in an active state. During the active state, the node retransmits the message to the discovery of a certain time interval Td and enters sleep mode if it detects another node, which corresponds to a node or a node of higher rank, which can handle the communication and routing. The three types of state processing is represented in Fig. 2.7, which states performance of the node during discovery, active and sleep state [26].

Sleeping state of a node can be achieved from discovery state or the active state. Prior to the sleeping state, all the timers like Ta and Td are cancelled and radio is put down to power off.

The node completes time frame Ts to return to the discovery state, which is application or system input.

GAF follows a load-balancing mechanism to maintain a constant communication medium or routing path between the nodes. This is done to make the nodes in the grid work efficiently and to analyze the nodes increase in lifetime. This is dependent on the assumption that all nodes in a specific grid are equal and no node is completely used; this is repeated till all the nodes are died out. There is possibility of nodes available in a pool with more energy resources or in other words with higher ranks. To deal with these nodes there is slot available so that they may handle some other process among other nodes. This is the case where nodes are commanded to shift back to discovery state after the completion of active state

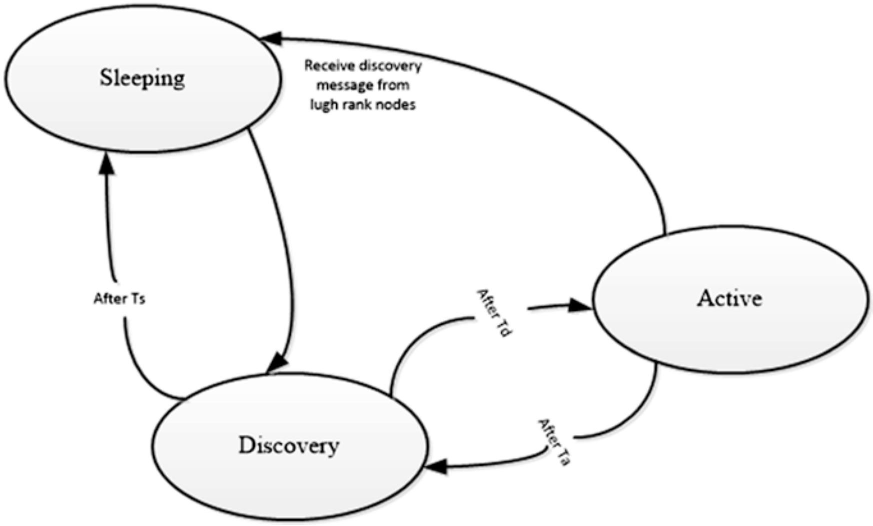


Fig. 2.7 Showing three state transitions in geographic adaptive fidelity [26]

time interval Ta. Timer Ta is settled at *enat* and discovery message advertisement is started. Timer Ts is set to sleeping time which is again equal to *enat*, i.e. the sleeping time.

2.6.6 Dynamic Source Routing (DSR)

Dynamic Source Routing protocol is one of the reactive protocols. It states the way for all nodes in a network to find a route to multiple destinations in the grid. Overall overhead of network bandwidth is minimized using DSR. DSR does not broadcast routing updates periodically. Battery power is also saved by stopping the periodic large routing update messages broadcast. In case of failure however there is an option available at MAC layer to inform the routing protocol [23, 26].

Following are properties of Dynamic source routing:

1. DSP takes advantage of saving space in nodes memory by avoiding the up-to date routing information in intermediate nodes.
2. No periodic update messages broadcast saves network bandwidth.
3. Battery is also saved by avoiding periodic updates in DSR.
4. Information is gathered by sniffing the routes in received packets.
5. Piggybacking new request allow unidirectional communication to the source node.
6. Interface address filtering turned off in order to scan all packets. This allows the interface to run in a promiscuous mode free environment. An intruder can sniff to all the information in the packets for some valuables such as credit card information and passwords.

Route Discovery: DSR stores all known routes in a cache. RREQ is used by the source node to broadcast messages to start conversation between nodes. Other nodes search their own cache in order to find the route to source node as soon as they receive RREQ. RREQ is forwarded in case of route unavailability and current node address is stored in sequence of hops. RREQ propagation is a recursive process and kept on broadcast till the destination is available by itself. In this scenario a RREP is unicasted to the source node. RREP packet contents itself contain hops sequence in the grid to reach the destination [23, 26] (Fig. 2.8).

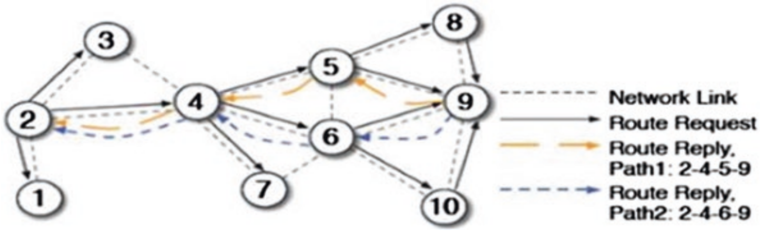


Fig. 2.8 DSR route discovery for target node [27]

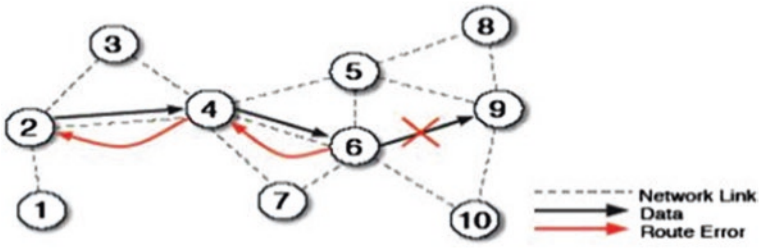


Fig. 2.9 DSR maintenance for error route [27]

Route Maintenance: Error packet is generated and sent to source node in case of an invalid path discovery. The sent route does not contain the hop that has error in the path and it is removed from the memory i.e. cache of host. All related hop routes also are deleted (Fig. 2.9).

References

1. Ward M (2006) RFID: Frequency, standards, adoption and innovation. JISC Technol Stand Watch. Available at http://www.jisc.ac.uk/techwatch
2. Zhang Y, Yang LT, Chen J (eds) (2009) RFID and sensor networks: architectures, protocols, security and integrations, pp 511–536
3. http://www.rfidc.com/docs/introductiontorfid_technology.html. Accessed 20 Apr 2010
4. http://www.aimglobal.org/technologies/rfid/what_is_rfid.asp. Accessed 20 Apr 2010
5. Dressen D (2004) Considerations for RFID technology selection. Atmel Appl J (Corporate Communication: Atmel Corporation)
6. http://en.wikipedia.org/wiki/File:Sensornode.svg. Accessed 20 Apr 2010
7. Laran (2004) RFID: a basic introduction to RFID technology and its use in the supply chain
8. http://en.wikipedia.org/wiki/File:WSN.svg Accessed 20 Apr 2010
9. Beiwei Z, Kunyuan H, Yunlong Z (2010) Network architecture and energy analysis of the integration of RFID and wireless sensor network. In: Proceedings of Chinese control and decision conference
10. Liu H, Bolic M, Nayak A, Stojmenovi I (2009) Integration of RFID and wireless sensor networks. Bentham Science Publishers, Sharjah
11. Ilie-Zudor E, Kemeny Z, Egri P, Monostori P (2006)The RFID technology and its current applications. In: Proceedings of the modern information technology in the innovation processes of the industrial enterprises-MITIP, pp 29–36
12. Liu H, Bolic M, Nayak A, Stojmenovi I (2008) Taxonomy and challenges of the integration of RFID and wireless sensor networks. IEEE Netw 26–32
13. Thompson D (2006) RFID technical tutorial. J Comput Sci Col 21(5):8–9
14. Sung J, Sanchez Lopez T, Kim D (2007) The EPC sensor network for RFID and WSN integration infrastructure. In: Proceedings of the fifth IEEE international conference on pervasive computing and communications, 618–621
15. Akyildiz IF, Su W, Sankarasubramaniam Y, Cayirci E (2002) A survey on sensor networks. IEEE Commun Mag
16. Mitsugi J, et al (2007) Architecture development for sensor integration in the EPC global network. Auto-ID labs white paper

17. Cheekiralla S, Engels DW (2005) A functional taxonomy of wireless sensor network devices. In: Proceedings of the 2nd international conference on broadband network, vol 2, no 3–7, pp 949–956
18. Heinzelman WB, Chandraskan AP, Blakrisshnan H (2002) An application-specific protocol architecture for wireless microsensor networks. IEEE Trans Wirel Commun 1(4):660–670
19. Lee CKL, Lin XH, Kwok YK (2003) A multipath ad hoc routing approach to combat wireless link insecurity, vol 1, pp 448–452
20. Al-Karaki JN, Kamal AE (2004) Routing techniques in wireless sensor networks: a survey. IEEE Wirel Commun 11:6–28
21. Krishnamachari B, Estrin D, Wicker S (2002) Modeling data centric routing in wireless sensor networks. IEEE INFOCOM, New York
22. Intanagonwiwat C, Govindan R, Estrin D (2000) Directed diffusion: a scalable and robust communication paradigm for sensor networks. In: Proceedings of the 6th annual ACM/IEEE international conference on mobile computing and networking (MobiCom_00), Boston, MA
23. Sadagopan N, Krishnamachari B, Helmy A (2005) Active query forwarding in sensor networks. Ad Hoc Netw 3(1):91–113
24. Zhao J, Erdogan AT, Arslan T (2005) A novel application specific network protocol for wireless sensor networks. IEEE Reference number 0-7803-8834-8/05
25. Yao Y, Gehrke J (2002) The cougar approach to in-network query processing in sensor networks. In: SIGMOD
26. Stemm M, Katz RH (1997) Measuring and reducing energy consumption of network interfaces in hand-held devices. IEICE Trans Commun (Special Issue on Mobile Computing)
27. Ullah M (2009) Evaluation of routing protocols in wireless sensor networks. Master's thesis

Chapter 3
Challenges and Issues in the WSN and RFID

Abstract In this chapter research based work encompassing present and future trends of RFID and WSN has been presented. Various research aspects in the area of security with respect to sensing network and identification have substantially been explored and pointed out for future investigation. Efforts have been made to incorporate the latest and updated research so as to provide the premises for the work that has been done in the subject area.

Keywords RFID · WSN · Routing

3.1 Contemporary Work

The integration of promising technologies, RFID and WSN is expected to deliver more acceptable and absorbing results. However, many studies related to the integration of RFID and WSN are considered inadequate. The authors, Zhang et al. [1] proposed three types of integration namely heterogeneous network architecture, RFID reader integrated with WSN node in one device and smart tags having limitations to be applied over large area, due to requirement of long cable between reader and antennas.

Sung et al. [2] investigated the current research and limitations in the field of RFID and Wireless sensor networks and suggested the idea of a large-scale infrastructure which will be based on the standard EPC network architecture. The proposed architecture was planned to be an accurate infrastructure of the future.

By considering integration of RFID and EPC global-based RFID technologies Mitsugi et al. [3] proposed four reference models for integration.

1. A mixed architecture of WSN and RFID whose integration is top layer i.e. application level
2. Integration at hardware level where tags and WSNs data are collected by a RFID reader
3. WSN and RFID mix with the filter and collection level integration
4. EPCIS level logical integration which allows for a mixture of WSNs and RFID

J. Ahmed et al., *RFID-WSN Integrated Architecture for Energy and Delay-Aware Routing*, SpringerBriefs in Applied Sciences and Technology,
DOI 10.1007/978-981-287-414-6_3

Without taking into account for integration, Cheekiralla et al. [4] sub classified communication, power, memory, sensors to propose taxonomy of wireless sensor networks including RFID devices.

Zhang et al. [5] discussed three general and one novel composite network architecture. The authors focused RFID sensor network systems that utilizes distributed network architecture of smart nodes and to read tags through smart nodes it consider the MAC protocol with the Basic Frame Slot Aloha (BFSA). They also analyzed that the energy consumption of the smart nodes is mainly due to receiving of data from tags and relaying. Firstly, for the smart nodes to query the tags within the reach, they used BFSA protocol as the MAC protocol, and computed the energy consumption. Then, the WSN energy model was utilized for the computation of the smart node power for data relaying. And at last the lifetime of the smart node was also assessed.

Liu et al. [6] discussed four types of integration of the RFID with sensor networks, these include, tags integration with the wireless sensor node, readers integration with wireless devices and wireless sensor node, and mix of RFID and sensors. According to their investigation the powered RFID tags like the active or semi-passive RFID combined to the WSN has a wide field implementation because it expands its applications in terms of broad reading range. Further, the new RFID chipsets from other manufacturing parties like Intel are trying to decrease the implementation and manufacturing cost of RFID readers that lies in the UHF range as well. Thus, due to use of powered RFID technology and low price, readers that allow stationary deployment of readers in the same manner like WSNs can be organized. Similarly, the ability of multiple hop to hop communications of sensor networks can be utilized to take out information from the readers.

Catarinucci et al. [7] presented a costeffective multiple IDs tag, which is called S-Tag, which enables the generic sensors to be connected together; it is also enables the transmission of a properly combined ID codes that depends on the input of real value. They connected S tags to various types of the sensor and validate its suitability. The authors present and demonstrate the outcomes related to the S-Tag application are under much stressed condition. The very easy and simple interfacing to generic sensors, a compatible to the typical RFID technology and low price solution stands the S-Tag a competent nominee for the enhancement of the RFID-based telemedicine systems capabilities.

Ferrer-Vidal et al. [8] presented an overview of a performance wise optimized radio frequency (UHF) for tags identification and implanted the sensors and power banks. They investigated that substrates like an Organic one, e.g. a paper which has been very infrequently utilized in RF and UHF applications earlier were able to take advantage of the inkjet printing process that is necessary to realize the ultra low cost Sensor tags/RFID. It is applicable to the frequencies ranging reaching up-to 950 MHz in history. Initial frequency for this range would be 13.56 MHz Additionally, they usage of implanted re-chargeable film batteries (thin) will also enlarge the nodes' lifetime. They also investigated that compact dual-polarization antenna usage operating in the RF/UHF bands make high reading range possible with high rate transfer of data.

Catarinucci et al. [9] proposed a general purpose costeffective multiple ID tags that enable the generic sensors connection; it is regardless of the real value measured, that has the capability of transmission, on standard RFID reader interrogation, combined form of the ID codes that univocally converts codified values read from the measured sensor. They presented the results related to the application of the S-tag in very demanding situation, representing the high-quality concurrence between transmitted and received data. Moreover, examples of the use of the S-tag in very simple wireless sensor networks for temperature and humidity remote control have been reported and its good performance highlighted.

Li et al. [10] presented the advantage of the integration of WSN and RFID technologies. Animal and patient health monitoring and precious care is also discussed; in the field the real time monitoring system is very important. System structure of WSN and RFID is introduced and the results from the simulation of this integrated technology environment stated that it outperforms the existing RFID system with updating delay, cost of deployment and tag capacity requirement. The system of Hybrid WSN System and RFID (HRW) is also narrated in this research to enable the integration of the two technologies, which overcomes their disadvantages and puts their advantages to a good cause. The advantage of this technique over traditional architecture is avoidance of long wait queues, which is possible through the information in HSNs can be replicated among its neighbor nodes based on a special reduced functional RFID readers in HSNs.

Hussain et al. [11] presented the RFID and WSN integration in applications and smart homes of the system such as identifying a caregiver who entered the home. Then, they proposed an architecture consisting of RFID and a WSN to identify motion within a moving environment. As a result several useful applications can take advantage of this information. When WSN and RFID are combined, they can be used to create a system that is ideal for deployment in a smart home. Their proposed architecture has many application possibilities that can be implemented through a software system with little or almost no changes in the hardware.

Torres et al. [12] used the Synapse Network Evaluation kit and simulated the results. This scenario included both sensor nodes. To control the microprocessor a Python based control software is designed; to manage the tag IDs of people wearing them a database application is available. The RFID and WSN sensors were connected using I2C protocol interfacing. Also, the work of sending commands to the RFID node, to make it read a tag and send it back to the computer, was accomplished by the Python code developed which also controls the data signals. Their research had the potential of being adapted for use with secure real-time access control applications involving WSN and RFID technologies.

Pathak et al. [13] proposed a framework for software for integrated RFID mobile phones with adequate changes in the system is exemplified. They also discussed Operating System installation with a driver is also discussed in this research to run RFID reader and involvement of Java Platform ME package for support in programming for RFID reader. Proposed Scheme system service continuously watched over the RFID hardware for events. Standalone application made in Java can also use and control the RFID reader. Exemplary future applications and systems based on RFID and other technologies integrated with RFID mobile phones were also proposed.

Pereira et al. [14] presented a proposal of two heterogeneous architectures for integration between technologies RFID and WSN. A study was presented for this purpose with a model of non-deterministic pushdown automata (NPDA), which made their validations, through its states and transitions. This model represented the set of states sensor node and add on RFID transceiver. Then were defined: the language, alphabets and their evidence of recognition of words through its configuration instantly. The next step of their work included the proposal of models that can better represent the behaviors of the system. They also aimed to develop extension to the current model, wanted to add communicating among automata, so that they can better represent the system as a whole and from the scene to check the properties of the system: liveness, boundness, reach-ability among others.

López and Kim [15] proposed a framework for RFID and WSN integration in order to offer context aware services to users and objects. Their contribution in the integration of RFID and WSN constitutes a significant work in which sensor and RFID data merge to build dynamic context, and in which the designed architecture around the context proved to be compatible with the current EPC Network infrastructure. They proposed a practical implementation scenario for both real and simulated Wireless Sensor Network, Web services and EPCIS-like repositories. They also mentioned some software tools for monitoring and evaluating their algorithms.

Liu et al. [16] investigated recent research work and applications that integrate RFID with sensor networks, four classes of integration were discussed: “integrating tags with sensors, integrating tags with WSN nodes and wireless devices, integrating readers with WSN nodes and wireless devices, and a mix of RFID and WSNs”.

There are several researches done on the integration of WSN and RFID. Application of such research includes health care, supply chain management, managing cattle, condition of weapons in battle field, fire detection etc. researchers also performed protocol analyzing with various options. Some are mentioned below:

Heusse et al. [17] analyzed the performance anomaly of 802.11b networks theoretically. They derived the probability expressions of collision, throughput and time in contention period. Their research showed that, among all the nodes in the network there are few nodes which are responsible for degradation of network’s performance, by communicating at a lesser bit rate.

Saleem et al. [18] studied simulation research for evaluating ad hoc routing protocols and proposed a prime method for their evaluation. The protocols they studied included DSR, DSDV, AODV-LL, and Gossiping. Two key performance parameters, route optimality and routing overhead were modeled using proposed framework for performance evaluation.

Guo and Zhang [19] carried out survey on intelligent routing protocols in order to make contribution to network lifetime optimization in wireless sensor networks.

Protocol they discussed in their work mainly based on some intelligent algorithms as Ant Colony Optimization (ACO), Reinforcement Learning (RL), genetic algorithm (GA), fuzzy logic (FL), and neural networks (NNs). They made contributions by providing the guidance in WSN and computational intelligence (CI).

Hammoudeh and Newman [20] presented their research on routing that guarantees application specific services. They presented a novel routing protocol based on clustering for load balancing and optimization of route. The protocol was named as ROL, capable of meeting the application requirements by using various Quality of Service (QoS) parameters. They showed with the help of simulation results that proposed protocol increases the lifetime of a network more than any other similar scheme.

Sadagopan et al. [21] proposed a novel WSN routing protocol for query based applications. They gave a mathematical model for calculating the cost of energy related with a WSN query protocol called ACQUIRE. When proposed algorithm ACQUIRE was compared with other substitute schemes it was found that if parameters of ACQUIRE are set to their optimum values, it beats all the other similar schemes. The proposed protocol was declared best for single-shot, complexed-nature queries by extensive results. In particular, the performance of ACQUIRE was better than other schemes based on flooding for such queries in large networks.

Jacquet and Laouiti [22] carried out a performance analysis of proactive routing protocols with flooding-based routing mechanisms. The simulations and the modeling of control overheads were provided. Although extensive simulations were provided but the final analytical result was missing. A performance evaluation framework was provided for Bio-inspired ad hoc routing algorithms. However, the framework is only applicable to small-sized networks and does not cater for packet loss due to channel errors and collisions at the MAC layer.

In this [19], authors compared three multipath routing techniques (Multipath DSR, Multipath ZDR, and Multipath AODV) to ZigBee’s standard single path AODV routing protocol. Different sizes of WSNs were used to conduct simulations using the IEEE 802.15.4/ZigBee stack present in OPNET, and statistics were collected such as packet delivery ration, end-to-end delay, and battery consumption. The findings in results indicated that the packet delivery ratio in Multipath ZDR was highest, trading of with greater energy cost and shortest end-to-end delay. The performance of standard ZigBee AODV was inferior with respect to parameters mostly used when a network is udder the stressed load; however, being a single-path routing protocol, the consumption of energy is naturally lower than other examined protocols. Multipath AODV and DSR performed considerably poorer than Multipath ZDR with regard to all considered metrics as a result of increased inter-path and intra-path interference

Dahlstrom et al. [23] presented a performance comparison of AODV and DYMO based on the Zigbee standard. The performance comparison was based on the following metrics: packet delivery ratio, end-to-end delay, and energy consumption. Work was implemented for realistic node densities and simulation times in non-beacon enabled Zigbee networks. Their simulations showed that the performance of the protocols depend on the network topology. The results obtained indicate that DYMO outperforms AODV with respect to packet delivery ratio (PDR) and energy consumption. The Ad hoc On-demand Distance Vector (AODV) routing protocol enables the routing of data between a source and destination in mesh networks. The Dynamic MANET On-demand routing protocol (DYMO) was concluded to be a successor to AODV that was designed for wireless ad hoc networks.

Senoucia et al. [24] carried out research while selecting path in WSN in such a way that network's lifetime is increased. The definition of this parameter was determined in this work on the basis of provided type of service. By using common framework for evaluation and network's lifetime definitions, some representative routing protocols for sensor networks including hierarchical and flat routing protocols had been analysed. Analysed protocols included: DIRECT, GOSSIPING, FLOODING, LEACH, and HEED. Spatial-temporal distribution of network's lifetimes was analysed in detail through extensive simulations. Through this study, a new technique EHEED was proposed aiming at providing a good spatial-temporal distribution of lifetime. A comparatively analysis was also provided, comparing EHEED protocol to others. Experimental results showed that EHEED can extend the network lifetime remarkably and can be very effective for long-lived sensor network.

3.2 Issues and WSN and RFID

The study about integration of RFID and WSN will pose some significant challenges for the researchers when the integrated environment is composed of a large number of RFID and WSN components. To deal with such problems, simulation environment is needed to provide detailed analysis of application scenarios, to speed up the developing and testing process and minimize the cost and interference.

A large number of applications have emerged in recent studies, such as health and structural monitoring using RFID and WSNs. The main requirement that needs to be fulfilled by these applications is Quality of Service (QoS) including real-time delivery of data. Quality of service gets easily affected when network suffers from delay. Measuring delay accurately depends on synchronization of network. Additional overhead is introduced in traditionally synchronize the network, making network un-reliable causing packet loss. Thus, it is not always affordable to apply network synchronization to an operational network due to the limited node resource and large network scale. One of the challenges is analyzing the collected information.

3.3 Solution to the Challenges

There are number of parameters to evaluate the performance of a network including delay as an important one, such that a simple change in delay might result in variation of multiple metrics. On the contrary, when all the information is collected from the network, it results in large network overhead. Thus, the proper information is not received usually due to packet loss and limited resources. Moreover, an intrinsic randomness has been observed in distribution of delay depending on design of protocol. Efficiently and automatically extracting useful information from collected data becomes difficult.

In this regard, both the RFID and WSN technology have got a number of simulators like (QualNet, OPNet and SENSE, TOSSIM) RFID ("BL Ident Configurator", "RFIDSIM", "NS-2", "Matlab", "Labview", etc.) but there is still a room for the development of such a simulator that combines WSN and RFID technology.

In the field of RFID and senor networks' integration and protocol analysis, the simulator design needs the attention of researchers from many aspects but in this book, the research work has been focused on the achievement of following aspects:

1. RFID Tag Generation.
2. Tag energizing/de-energizing.
3. Data collection.
4. Tag/sensor data collection and providing to base station.
5. To analyse, implement and evaluate ACQUIRE protocol.
6. To analyse, implement and evaluate the directed diffusion protocol.
7. A comparative analysis is being performed between the two protocols.
8. A comprehensive model for delay performance measurement and analysis in a wireless sensor network is being discussed.
9. The provided delay model is light-weight and a robust method to calculate per-packet delay.

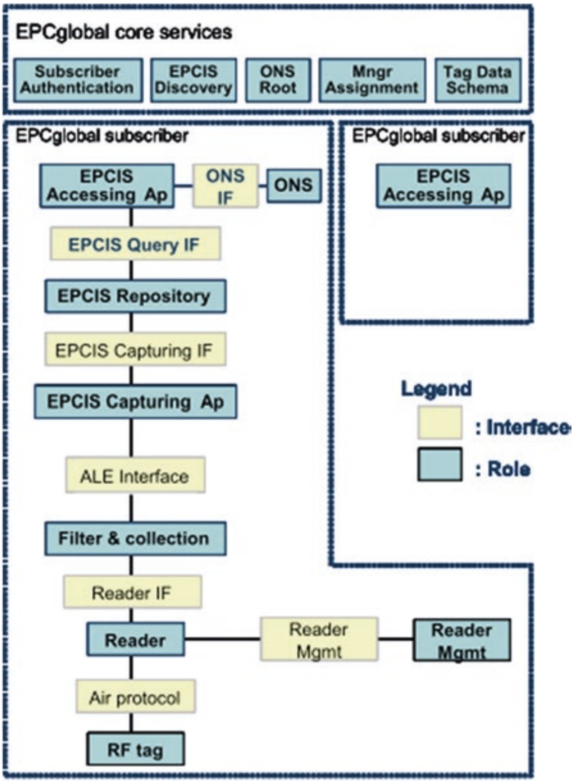
Integration part of the RFID and WSN can be divided into two major segments:

- Software integration
- Hardware integration

Mainly software controls the hardware or prototype that will enable the integration of technologies and let us perform the data conversion from one device/technology to another. More specifically it can be stated that at this point the term "software" refers to protocol. RFID and WSN have their own protocols through which the data or sensor information is collected, represented and then sent to other node and (or) base station. Data or information is exchanged in terms of packets in both protocols but one of the challenge is first to decode the information in the packets and then combine them in order to achieve a single packet formatted for smart node.

Both packets have their own heads and tails and also the bit pattern and lengths are varied. Diving in the detailed packet information we can see that sensor information is somehow matches in both, as RFID is using information without destination field and physical properties while in WSN the sensor informs the physical properties and the destination or next ad hoc node address. In addition, we need to relocate the next smart node with physical information at the same time the smart

Fig. 3.1 EPCglobal architecture reference model



node acts as base station for several RFID sensors. The block diagram of the architecture of the integration model is given in Fig. 3.1.

References

1. Zhang L, Wang Z (2006) Integration of RFID into wireless sensor networks: architectures, opportunities and challenging problems. In: Proceedings of the 5th international conference on grid and cooperative computing workshops (GCCW’06)
2. Sung J, Sanchez Lopez T, Kim D (2007) The EPC sensor network for RFID and WSN integration infrastructure. In: Proceedings of the 5th IEEE international conference on pervasive computing and communications workshops, pp 618–621
3. Mitsugi J, Inaba T, Patkai B, Theodorou L (2007) Architecture development for sensor integration in the EPCglobal network, Auto-ID labs white paper
4. Cheekiralla S, Engels DW (2005) A functional taxonomy of wireless sensor network devices. In: Proceedings of the 2nd international conference on broadband network, vol 2(3–7). pp 949–956
5. Zhang B, Hu K, Zhu Y (2010) Network architecture and energy analysis of the integration of RFID and wireless sensor network. In: Paper presented at the conference on Chinese control and decision, p 1381
6. Liu H, Bolic M, Nayak A, Stojmenovi I (2008) Integration of RFID and wireless sensor networks. School of Information Technology and Engineering, University of Ottawa, Ottawa, K1N 6N5

7. Catarinucci L, Colella R, Tarricone L (2010) Integration of RFID and sensors for remote healthcare. Innovation Engineering Department University of Salento, Lecce, ISBN 978-1-4244-8132-3
8. Ferrer-Vidal A, Rida A, Basat S, Yang L, Tentzeris MM (2006) Integration of sensors and RFID's on ultra-low-cost paper-based substrates for wireless sensor networks applications, 1-4244-073 2
9. Catarinucci L, Colella R, Tarricone L (2009) A cost-effective UHF RFID tag for transmission of generic sensor data in wireless sensor networks. IEEE Trans Microw Theory Tech 57(5):1291–1296
10. Li Z, Shen H, Alsaify B (2008) Integrating RFID with wireless sensor networks for inhabitant, environment and health monitoring. In: Proceedings of the 14th IEEE international conference on parallel and distributed systems
11. Hussain S, Schaffner S, Moseychuck D (2009) Applications of wireless sensor networks and RFID in a smart home environment. In: Proceedings of the 7th annual conference on communications networks and services research
12. Torres B, Pang Q, Skelton GW, Bridges S, Meghanathan N (2010) Integration of an RFID reader to a wireless sensor network and its use to identify an individual carrying RFID tags. Int J Ad hoc, Sens Ubiquitous Comput (IJASUC) 1(4):1–15
13. Pathak R, Joshi S, Parandkar P, Katiyal S, Ludhiyani A (2010) Applications of RFID and a software framework for facilitating its integration in mobile phones. Int J Acad Res 2(5):18
14. Pereira DP, Dias WRA, Braga M, da Silva Barreto R, Figueiredo CMS, Brilhante V (2008) Model to integration of RFID into wireless sensor network for tracking and monitoring animals. In: Proceedings of the 11th IEEE international conference on computational science and engineering
15. López TS, Kim D (2008) Wireless sensor networks and RFID integration for context aware services. Information and Communications University119 Yuseong-gu, 305-714, Daejeon, South Korea
16. Liu H, Bolic M, Nayakand A, Stojmenovi I (2008) Taxonomy and challenges of the integration of RFID and wireless sensor networks. IEEE Network 22(6):26–32
17. Heusse M, Rousseau F, Sabbatel GB, Duda A (2003) Performance anomaly of 802.11b. In: Proceedings of the 22nd annual joint conference on the IEEE computer and communications societies (INFOCOM '03). San Francisco, pp 836–843
18. Saleem M, Khayam SA, Farooq M (2008) A formal performance modeling framework for bio-inspired ad hoc routing protocols. In: Proceedings of the 10th annual conference on genetic and evolutionary computation (GECCO'08). Atlanta, pp 103–110
19. Guo W, Zhang W (2014) A survey on intelligent routing protocols in wireless sensor networks. In: Proceedings of the journal of network and computer applications. vol 38, Elsevier, pp 185–201
20. Hammoudeh M, Newman R (2013) Routing in wireless sensor networks: QoS optimisation for enhanced application performance. In: Proceedings of the journal of information fusion, i0
21. Sadagopan N, Krishnamachari B, Helmy A (2005) Active query forwarding in sensor networks. J Ad Hoc Netw 3(1):91–113
22. Jacquet P, Laouiti A (2000) Overhead in mobile ad-hoc network protocols. Research report 3965, INRIA
23. Dahlstrom A, Rajagopalan R (2013) Performance analysis of routing protocols in Zigbee non-beacon enabled WSNs. In: Proceedings of the conference on IEEE consumer communications and networking (CCNC)
24. Senoucia MR, Melloukb A, Senoucid H, Aissanic A (2012) Performance evaluation of network lifetime spatial-temporal distribution for WSN routing protocols. In: Proceedings of the journal of network and computer applications. vol 35, Issue 4, Elsevier, pp 1317–1328

Chapter 4
The Delay Model for *ACQUIRE*

Abstract In this chapter, after analysing the identified challenges from literature survey, a mathematical model is presented to calculate the end-to-end delays of a routing protocol ACQUIRE. Route selection is also discussed which is very important part in communication patterns. Its impact can be observed in the form of network delay when comparatively longer routes, involving greater number of sensor node are selected and also when network lifetime is degraded when choosing short routes, causing batteries depletion.

Keywords Routing · Delay · Timeliness · Probability

4.1 Basic Concept

Wireless Sensor Network (WSN) is formed of resource-constrained nodes group together in order to communicate messages between hops and also comprise groups of data containers. WSN features are highly application-dependent and can vary as domain of application varies. For example, two of the WSN parameters that can vary with higher probability are size of network and density of the nodes; others include mobility of a node mobility and disclosure to ambient phenomenon. There are, however, few limitations while implementing wireless sensor networks, which include the lack of reliability of node-to-node links and no stationary relays. These, along with strong energy constraints represent a key challenge for providing timeliness assurances.

The current advancements in the area of wireless sensor networks have made possible its usage in various applications including structural health monitoring. Main requirement for these applications is the guarantee of WSN quality of service (QoS), for example, the delivery of real-time data. Of the major factors affecting the quality of service of the system, is a significant delay.

J. Ahmed et al., *RFID-WSN Integrated Architecture for Energy and Delay-Aware Routing*, SpringerBriefs in Applied Sciences and Technology,
DOI 10.1007/978-981-287-414-6_4

4.2 Main Challenges

There are several existing methods to deploy the merged architectural support in WSN, however there are limitations for these methods. For example, exceeding maximum limit in a chain supply system [1] the network could possibly be broken during high bandwidth communication.

The real-time drive based classical methods claim the behaviour in each network layer should be deterministic. Attainable deterministic behaviour is only possible under ideal environments [2]. Considering the MAC layer, constrained delay is possible to accomplish through interrupted continuous detection and synchronization means neighborhood, which is often not economical in terms of energy. Correspondingly, hop-to-hop routing level, knowing about the network topology globally, does not conform to node's memory size which is limited. In addition, the common and vital factor which is also base for many approaches is, assuming the channel to be in perfect condition. High Bit-Error-Rate (BER) is found to be a problem in sensor networks. The application of sensor network with high packet demand like Wi-Fi, the BER increases with the number of nodes. In methods based on queuing theory, it depends on the service and/or inter-arrival times that a message may or may not fulfil their duration. Nevertheless, the estimates of these parameters when off-line, are not precise, so at run-time we need them to be tuned. To prevent delays there are several constraints, which also restricts end-to-end communication in real WSN deployments.

4.3 Network Delay Changes Caused by Routing Events

Two essential network parameters directly affecting several wide-area network applications are delay variations and network delays. These applications include real-time voice over IP and multicast streaming, time-critical financial transactions, locality-aware systems for redirection and server selection, positioning systems, proximity-aware DHTs and overlay routing systems [3]. The network delay sensitive applications feature the significance of understanding when, why and by how much network delays vary. Given source host and destination host on the internet when analyse, network delays can transform over time significantly.

The primary factors contributing to the network performance changes significantly are traffic fluctuations and network topology changes. Other factor that results in change of routes, affecting destination paths, includes network topology modifications such as traffic engineering or link failures. Whenever traffic sources are modified behaviourally, traffic fluctuations occur, e.g., flash crowd events. Thus network performance changes because of two fundamental causes discussed above, understanding the effect of these two causes on application performance is critical and in order to guarantee better network performance, their effect can be predicted. There is a significant contribution of user behaviour in traffic fluctuations and modelling the behaviour that is highly unexpected like DDoS attacks, is challenging. On the contrary, at inter-domain level, routing changes can be observed inactively and can be used directly for network performance prediction. The ability of network to perform such prediction is useful in applications like network-based performance-sensitive route selections and host-based proactive mitigation against performance degradation.

4.4 Delay Techniques

Delay has been measured at different levels through different techniques.

1. At the MAC level: In [4] static nodes are arranged under hexagonal topology to achieve hard real-time guarantee. Some changes in the requirements are seen in [5] this requirement is later relaxed in [5] however, the condition of static nodes still remained. Besides, the assumptions made in both works are the network communication conditions are optimum it follows bounded density. Neighbouring nodes are usually scheduled using TDMA technique in most of recent research work (e.g. [6]). Although the results obtained in these cases are valid underorganized environments, the assumption made in these cases is no network errors causing damage in transmission schedule and retransmission.
2. At the routing level: In [7] and [6] timeliness requirements are fulfilled by assigning velocities to messages. However, the assumption made in both works is localization capabilities equipped sensor nodes arranged in static networks. In [8] lengths routing paths are limited in order to guarantee less delay.
3. In [9] transmission latency of messages on each hop is determined by utilizing queuing models. The models explain that a traffic ruling mechanism is used which drops the messages with no expectation of meeting their time deadlines. Moreover, when the conditions are not stable, the pdf of delay distribution is estimated with a Gaussian distribution.
4. In [10] a schedule-ability condition has been approached in order to guarantee end-to-end delay estimation in multi-hop WSN. The assumption of specific path lengths, networks density, transmission speed of a channel and their transmission times, are also made.
5. In [11] end-to-end delays in WSN are estimated by probabilistic approach, message end-to-end delays are also estimated through this approach. This probabilistic estimation approach benefits adaptive QoS because of its instant updating ability at run-time.

The proposed delay model is based on Oliver's probabilistic model [11]. The purpose of presenting this work is evaluating the accuracy of the probabilistic approach and its validation. This metric gives the fair idea about the network status in terms of timeliness performance. Routing protocols that are already been proposed can be modified to make use of this metric and probability of end-to-end delays can be important parameter while taking routing decisions.

4.5 The Delay Model

4.5.1 *Timeliness Monitoring*

The main purpose of monitoring a network timely is building a parameter which is capable of evaluating end-to-end path in the form of timeliness performance at run-time. However, the feasibility of accurate estimation of end-to-end delay is quite low because of WSN working principles. Therefore, end-to-end delay of selected routing path has been estimated probabilistically in next section. Timeliness monitoring can be useful while designing the network strategy, on different levels. This information can be propagated towards the destination node or back to the source node by routing protocol, for adjusting the decisions made for routing to the stability of the selected path.

4.5.2 *Definitions and Notations*

A WSN can be represented as a graph $G(N,L)$ where N represents set of nodes and L represents a set of single-hop links. Two directly connected nodes $n_i, n_j \in N$ at a given time and between them there exist a link $l \in L, l = (n_i, n_j)$ such that n_i and n_j can communicate with each other by sending and receiving messages. $\hat{s} \subset N$ is the subset of sinks. Sinks is supposedly overtaking other nodes, when it comes to energy availability and resources. A sequence of links $(n_1, n_2) \ldots (n_{M-1}, n_M)$ form a routing path S_M such that every node including the intermediate node is connected to next node directly in the routing path, thus providing a multi-hop link between the source node and the destination node. A path, $\hat{s}$, is subset of another path g. In that case, we define $\hat{s}$ as a segment of g. Hence, $|S_{n1,nM}| = M$, which is the length of path.

4.5.3 *End-to-End Delay Estimation Using Probability Technique*

It is significant for time sensitive applications to have monitoring information related to end-to-end timeliness performance. The target in this research is wide-area WSN especially those related to monitoring of environment (e.g. detection and indication of fire, structural monitoring of buildings, etc.) with data acquisition that is timeliness sensitive (e.g. intrusion or fire alarms, damage in building's structure, etc.).

Routing protocol is responsible for making routing paths in order to provide a link between the sink and the sensor node. In such cases, some date would be continuously produced by nodes at a definite frequency, and for that period of time the paths are expected to be re-used. During the procedure while paths are building, the decision is needed to be made by routing protocol among multiple paths based on predefined criteria. Under these circumstances the probabilistic timeliness parameter works well because it selects the paths that have higher probability of performing well.

Case 1:
Considering the simplest case where $|S_M| = 1$ i.e. the path comprises only one link, $l = (n, \hat{\mathrm{s}})$ between any two hops n and $\hat{s}$. Thus, the transmission latency between these two hops can be represented by a random number D_M and its *pdf* is given by $p_M(\gamma) = P(D_M \leq \gamma)$ which is the probability that γ'' is the delay introduced by M in forwarding the message to the next hop. For message transmission latency β, this is the time between entrance of message and reception of acknowledgement. Thus,

$$\beta = t_{ack} - t_{in} - T \tag{4.1}$$

where,

t_{in} = Time when a message enters a node
t_{ack} = Time when acknowledgement is received
T = Additional time to receive the acknowledgement

At this point, the distribution of RV, d_M is not known which is required for measuring the expected value and probability of delay. To characterise the distribution, sample mean $\bar{x}$ is calculated to estimate the latency and sample variance s^2 to indicate the link quality [10]. A parameter $\rho(0 \leq \rho \leq 1)$ is set to weigh the actual measurements with respect to the past using exponential weighted moving average (EWMA) [8]. Hence,

$$\bar{x}'_M = \rho\beta + (1-\rho)\bar{x}'_{M-1} \tag{4.2}$$

$$s^{2'} = \frac{\rho}{2-\rho} s^2 \tag{4.3}$$

Equation 4.4 provides the sample variance s^2 with low memory requirements.

$$s^2_M = \frac{M-1}{M} s^2_{M-1} + \frac{M-1}{M} (x - \bar{x}_M)^2 \tag{4.4}$$

Equations 4.4 and 4.3 are updated at each node every time a message is forwarded through it.

4.5.4 End-to-End Latency Estimation

For RV, d_M the end-to-end latency for path M is also a Random Variable. By combining delays at the intermediate links,

$$d_M = \sum_{\forall (i,j) \in M} d(i,j) \tag{4.5}$$

Also,

$$p(d_M(t)) = P(d_M \leq t) \tag{4.6}$$

To characterize the distribution further, the *pdf* of the path is assumed as a normally distributed RV, $d_M = N(\mu d_M, s^2 d_M)$ it is uniform, non-negative and independent. Thus, Central Limit Theorem can be applied for convergence of the distribution [2].

Now,

$$\begin{aligned} \mu &= \bar{x} d_M \\ &= \sum_{\forall l \in M} \bar{x}' d_l \\ \sigma^2 &= s^2 d_M \\ &= \sum_{\forall l \in M} s^{2'} d_l \end{aligned} \tag{4.7}$$

Therefore, considering circumstances described above Eq. 4.6 becomes:

$$p(d_M(t)) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{D} e^{\frac{y^2}{2}} dy$$

where,

$$t = \frac{d_M - \mu d_M}{\sigma d_M} \tag{4.8}$$

Considering the above case, expected value of end-to-end latency M is μd_M.

When applying Central Limit Theorem, the RV is assumed to be independent. This assumption of independent might come with certain risk, since it is a strong assumption. Network might go through message latencies, under such circumstances events are forced to happen in dependent ways. However, initially this assumption was taken as an idea and it is expected that the possibility will be minimized since WSN has generally low throughput and nodes are distributed in spatial manner.

4.6 Energy Estimation

In [12] authors proposed a mathematical model in order to analyse the performance of ACQUIRE in terms of its expected time to complete the query and energy costs associated with it. The focus of their research was analysis of optimal look-ahead parameter *d*. The parameter they chose for analysis in terms of energy cost is number of transmission.

The assumptions for placement and characteristics of sensors are:

1. Sensors are laid uniformly in the region
2. Transmission range of all the sensors are same
3. Stationary nodes

4.7 Energy Analysis

4.7.1 Basic Definition and Notation

In [12] the considered scenario is such that a sensor network is comprised of X sensors. This considered network has the ability to track certain variables values including temperature, air pressure, humidity etc. Assume this network tracked N variables which are $V = \{V_1, V_2, \ldots, V_N\}$. The probability of each sensor to track these N variables is equally likely. Let the query to be $Q = \{Q_1, Q_2, \ldots, Q_M\}$ that comprises of sub-queries that are M in number such that $1 < M \leq N$, Let S_M be the average number of steps taken to resolve a query consisting of M sub-queries.

Now, consider the querier generating the query is sensor x^* and Q is the issued query consisting of M subqueries. Let d be the look-ahead parameter i.e. each sensor can send request to get information (updated or not) from sensors that are d hops away from it. Generally, on getting a query following functions are performed by sensor x:

1. *Local Update*: If the information a particular sensor has is not fresh, sensor x sends an update information request to all sensors within distance d hops away. This request is travels hop by hop, eventually reaches the destination. The sensors that are receivers of update requests then forward their information back to sensor x. Let the energy consumed during this procedure is E_{update}.
2. *Forward*: when requested query has been answered on the basis of received information, the querier then forwards the remaining unsolved query to some other node randomly chosen within d hop distance.

The important factor is quantifying the triggering of updates, since the update is only triggered when the information is not fresh. This quantification is termed as update frequency, which is an average amortization factor c. Thus at a given node, update triggered only once every c query. When the forwarded query is completely resolved, the destination node containing the query returns the whole reverse path and the completed response to the querier x^*. The expected number of hops from the node where the query is completely resolved to x^*, is denoted by α. Let S_M be the average number of steps to answer a query of size M. Thus, the average energy consumed to answer a query of size M with look-ahead d can be expressed as follows:

$$E_{avg} = \left(cE_{update} + d\right)S_M + \alpha \tag{4.9}$$

Generally, S_M is reduced when look-ahead parameter d is increased, while update energy cost E_{update} increases when d is increased.

4.7.2 Cost of Energy in Updating

Updating the information might be an expensive operation in terms of energy consumption, if E_{update} is the energy cost for updating in order to process the active query, it can be calculated as follows.

According to [12] assume that the active node x is generating a query Q. For a given a look-ahead value d, the active node x sends update request to get fresh information from sensors that lie within the range of d hops. This update request will be subsequently forwarded by all sensors within d hops except those nodes which are exactly d hops distant from querier x.

Thus the number of transmissions needed to forward this request is $f(d-1)$ when the number of nodes is within $d-1$ hops. The required information will then be forwarded to x by the requested nodes. Now, the information contained by sensors that are one hop away will be transmitted once, similarly for two hops distance the information will be transmitted twice and same is the case for sensors that are d hops away, transmission is going to be d times. Thus,

$$E_{update} = \left(f(d-1) + \sum_{i=1}^{d} iN(i) \right) \tag{4.10}$$

where

$N(i)$ is the number of nodes at hop i.

4.7.3 Total Energy Cost

In [12] it is assumed that the resolved query is forwarded by each active node to another node whose distance from that active node is exactly d hops, requiring d transmissions. Thus while answering a query of M size, the average energy spent is:

$$E_{average} = \left(cE_{update} + d\right)S_M + \alpha \tag{4.11}$$

where

α: The expected number of hops from node
c: The amortization factor
d: Look-ahead parameter
S_M = Average number of steps in answering a query

α, is the cost of returning the completed response back to the querier node. This response can be returned along the reverse path in which case α can be at most dS_M. Thus,

$$E_{average} = \left(cE_{update} + d\right)S_M \tag{4.12}$$

Special Case:
Considering $d = 0$:

If the look-ahead $d = 0$, the querier node x will not be requiring to send requests for updates from other nodes. The query will be resolved by x with the information it has, and remaining unresolved query will be forwarded to a randomly chosen neighbour.

The expressions show that look-ahead parameter d varies only with the amortization factor c and it is not dependent on the parameters M or N. Generally, the lower the value of c is, the higher will be the look-ahead parameter d.

References

1. Rizvanovic L, Fohler G (2007) The MATRIX a framework for real-time resource management for video streaming in networks of heterogenous devices. In: The international conference on consumer electronics, Las Vegas, USA
2. Decotignie JD, Keynote (2008) Real-time and wireless sensor networks: Why do we need another view at it? In: 7th international workshop on realtime networks (RTN) (abstract), pp 6–7
3. Pucha H, Zhang Y, Mao ZM, Hu YC (2007) Understanding network delay changes caused by routing events. In: Proceedings of ACM SIGMETRICS
4. Caccamo M, Zhang L, Sha L, Buttazzo G (2002) An implicit prioritized access protocol for wireless sensor networks. In: 23rd IEEE real-time systems symposium (RTSS)
5. Watteyne T, Auge-Blum I (2005) Proposition of a hard real-time MAC protocol for wireless sensor networks. In: 13th IEEE international symposium on modelling, analysis, simulation of computer and telecommunication systems (MASCOTS)
6. Gobriel S, Cleric R, Mosse D (2008) Adaptations of TDMA scheduling for wireless sensor networks. In: 7th international workshop on real-time networks (RTN)
7. Chipara O, He Z, Xing G, Chen Q, Wang X, Lu C, Stankovic J, Abdelzaher T (2006) Real-time power-aware routing in sensor networks. In: 14th IEEE international workshop on quality of service (IWQoS)
8. Sahoo A, Baronia P (2007) An energy efficient MAC in wireless sensor networks to provide delay guarantee. In: 15th IEEE workshop on local and metropolitan area networks (LANMAN)
9. Abdelzaher T, Prabh S, Kiran R (2004) On real-time capacity limits of multihop wireless sensor networks. In: Proceedings of the IEEE international real-time symposium (RTSS)
10. Karenos K, Kalogeraki V (2006) Real-time traffic management in sensor networks. In: Proceedings of the 27th IEEE real-time systems symposium (RTSS)
11. Oliver RS (2009) Technical report on estimation of the probability density function of end-to-end delays in WSNs
12. Sadagopan N, Krishnamachari B, Helmy A (2005) Active query forwarding in sensor networks. Ad Hoc Netw 3(1):91–113

Chapter 5
Simulator for Smart Node

Abstract This chapter covers simulator design of RFID and WSN based smart node in a delay factor environment. Simulator architecture with reference to standard software architecture model is also discussed. Smart node parameters calculation in a larger network is not an easy task to implement. Therefore, this scenario-based environment simulator can easily handle complex mathematical equations with larger data. The discussed simulation environment is an event based triggered system.

Keywords Simulator design · Smart node simulation · Environment simulation · Smart node simulator architecture

5.1 Proposed Solution to the Challenges

The primary goal for implementing a simulator is to accomplish the main tasks which have been presented in previous chapter, i.e. RFID Tag Generation, Tag energizing/de-energizing, Data collection, Tag/sensor data collection and finally providing the data to base station. Complexity of the scenario is another major problem in the design simulation which is minimized through the use of basic GUI interfaces and preset command sets. The developed simulator has the capability of creating an RFID tag and senor nodes by applying simple commands provided by the user using the GUI interface. User can also select a command from the provided commands sample space. Generated RFID tag is used for integration with a sensor node which acts like a reader for this tag. Thereafter, the integrated RFID tag and senor node communicates with the base-station to exchange the retrieved data. The data is obtained from the environment using generated RFID tags.

In addition, the simulator provides different interfaces for checking the tag state, commands implementation (Energizing and De-energizing) and random data obtaining. Rest of the chapter describes the issues and implementation of the simulator architecture design.

J. Ahmed et al., *RFID-WSN Integrated Architecture for Energy and Delay-Aware Routing*, SpringerBriefs in Applied Sciences and Technology,
DOI 10.1007/978-981-287-414-6_5

5.2 Simulator Architecture Design Issues

Implementing the simulator is a complex task and requires some basic architecture to better define its fundamental functionalities. One of the possibilities is to do so by exploiting the software reference architecture, Fig. 5.1. Security is not an issue in neighboring sensor nodes so keeping in view, the security layer is omitted from the

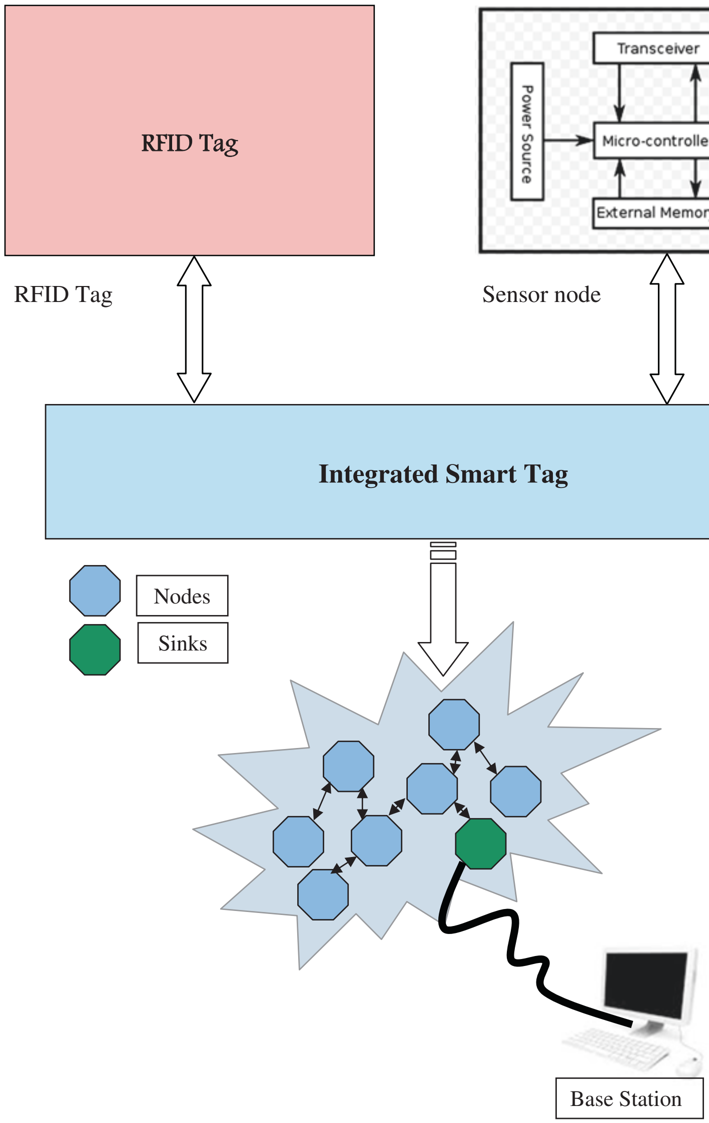


Fig. 5.1 Schematic diagram of smart node architecture

standard model; the whole communication and information exchange is taking place within a single node that requires no encryption therein. On the other hand, packet formation and fields concatenation is also important. As a result of this scenario, there is a prototype built by ELIMA partners for unit testing of info (IDU) which is a collection of technical (in field) operated data from various sensors. Another implementation of this logic is proposed by Emery [1] in form of integrated sensors in RFID architecture. Inside the system, data sensor combines the physical units and quantities of any given time with transmission timing of activity, at the same time transfer of data is done from the memory of sensor and processes stream operation. The "*Intro query*" represents the relation and combined form of tables and the role such as process data heuristics. The project of ELIMA [2] has been funded by EC research and the work aims to demonstrate the management and acquisition of life cycle of product data. This product data collect and exploit data from products in various passes of its life cycle including maintenance manufacture, end-of-life product recovery, and design. Tag sensor integration in embedded RFID is promoted by Deng et al. [3] (SE-RFID) which is also a possible implementation in an embedded system. Data sensed routes through the analog to digital converter and then pass RFID reader and collects user data. Collected items are referred (health monitor in time domain), and additional logic integration, provided for the extension of SE RFID. Ranasinghe et al. [4] collaborated vision of network of EPC global technology and the extension to the sensor proposed hardware level integration, specifically for the hardware architecture. It shows the development and importance of the energy harvesting and site for semi-active sensors. Zhang and Wang [5] represents the issue for the importance of combination of WSN and RFID reader, labeled as "intelligent integrated decade". Node intelligence deliberates the possibility to communicate with other node/labels and the forms WSN intelligent. There are many possibilities out of which most relevant are discussed in following part of this chapter.

There are two key components in the simulation design. One is the node itself with smart architecture and second is the simulation environment. There are several models available as an implementation of smart node. However, there are some constraints regarding the simulator architecture design. To fulfill these considerations in the simulator design, first there is a need to review existing embedded hardware implementation in detail.

5.2.1 Embedded Architecture Design

Logic packets programmable architecture is available for programmable (FPGA, CPLD, Microcontrollers) devices to implement the tag architecture. Because of the ability to re-program (Run-time) and easy modification, these types are used in the test field to add new functions, such as a set of sensors and logging data and data processing. Due to vast field of FPGAs that varies from power to performance is the main reason to deploy the design. Prototyping is easier on FPGA and this makes it more useful specifically in a timely strict manner. Power-specific design is an important issue in the design of any system and in RFID less power means more battery life or more power for other components in case of passive tags. Another major advantage is to deliver the design to an open source platform. It can be used either for design purpose or for research purpose. Most of the designs currently available are not in a deliverable format. Many research industries also focusing on the open source to make devices more useful and as open as possible.

Application Specific IC or ASIC on the other hand is also a good solution, however, the TTM (Time to Market) and cost per die is way different compared to FPGAs. ASIC are more compatible with the field and all parameters are specifically designed to work in extreme conditions. All the specifications of this implementation meet the criteria and it is easy to separate each module for simulator design.

In FPGA based architecture each component of the system serves as a module of the system. Each module has its own ports and each has its own implementation. As all modules are interconnected and working as a single system that is why each tag needs to be analyzed in its separate space. Following is a brief discussion for design of each component of smart tag.

5.2.1.1 Tag Components

Figure 5.2 represents general hardware architecture of an Adhoc enabled node with sensor interfaced within the system. There may be one or more sensors present in the system depending on the application. In the system implementation each tag has several components that need to be considered in the design architecture. Some of them are major ones.

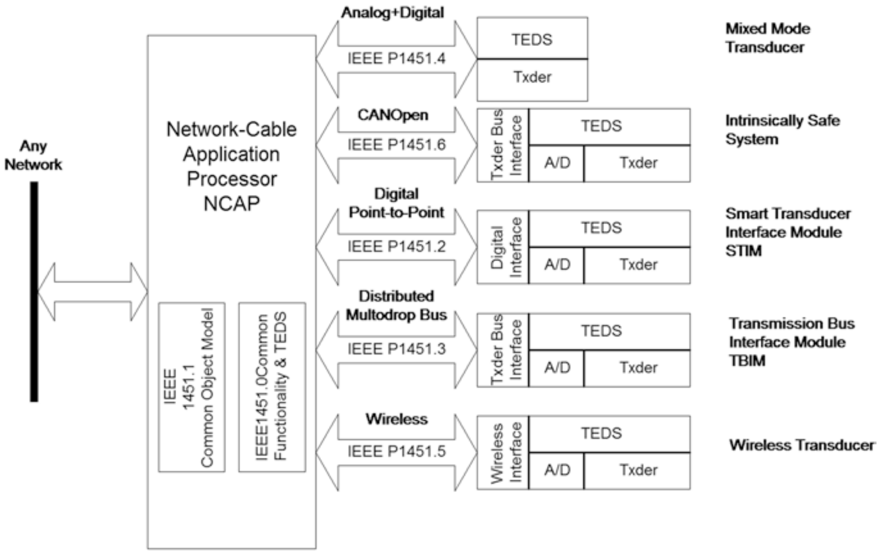

Any Network
Network-Cable Application Processor NCAP
IEEE 1451.1 Common Object Model
IEEE1451.0Common Functionality & TEDS
Analog+Digital
IEEE P1451.4
CANOpen
IEEE P1451.6
Digital Point-to-Point
IEEE P1451.2
Distributed Multodrop Bus
IEEE P1451.3
Wireless
IEEE P1451.5
TEDS
Txder
Txder Bus Interface
A/D
Digital Interface
Wireless Interface
Mixed Mode Transducer
Intrinsically Safe System
Smart Transducer Interface Module STIM
Transmission Bus Interface Module TBIM
Wireless Transducer

Fig. 5.2 Structure of IEEE 1451 standard

5.2.1.2 Sensor Component

There are two categories of sensor technology that simplifies its architecture. One is fixed with sensor functionality and the second is variable sensor function. Fixed jobs are important to the sensors where the functionality is limited or fixed tasks are executed on it. While on the other hand, sensors can do variable function by changing jobs e.g. protocol changing and form a new state within data fields. Both are runtime configurable systems and need simple protocol to work in a single way. In simulator architecture the reconfiguration represents a small memory of flags which acts like conditional parameters.

5.2.1.3 Battery

Tags reside between data and sensors, so data transfer means that tag writer works as a gateway to feed data from the A/D to a server for this particular job. The fixed job tag might sense a battery powered (active) and BAP. Alien [6] has the products that are mostly BAPs to improve writing performance and features e.g. embedded sensor and large memory. External analog/digital I/O is useful and various functions are optional in this case. Redemske [7] proposed the MAB (marking battery powered assistant) that is equipped with framework to detect the field strength of RFID testing and transmission of data. Most of the extreme forms of a tag (fixed), includes the passive words sensor.

5.2.1.4 Operating Frequency of Tag

Opasjumruskit et al. [8] proposed 125 kHz RFID package for embedded temperature sensor of animal health care. This work is done by on-chip digital (balanced) calibration. Cho et al. [9] suggests semi-passive GEN2 tag compatible with integrated temperature sensor. Temperature is measured by applying a time intensive readings. Calibration is needed to be done on the network and server side. Meiners et al. [10] proposed a smart sensor with pressure capability. It works on 125 kHz RFID technology for detecting incorrect placement of bandages, which is a treatment for venous ulcers.

5.2.1.5 UCODE

A mercury switch connects to tags to choose the brand that is working actively according to a mile. To do so, accelerometer data ID is represented per bit of data packet. UCODE ID Center represents all bits to e-bit No. 128 (UCODE), which represents geographical locations [11, 12]. These words are classified as static and negative sensor function. Active low-cost cards can be fixed by using the functions [13–15]. The function of the variables in context of BAP and on UCODE is discussed by Mitsugi [16] and Mori [17].

5.2.1.6 Power Consumption

Kobayashi [18] represents a low-power, active and small variable tags function. It is characterized by the adoption of the PT-engine, which is an embedded computer with a single-chip (SOC) wireless communication platform. It is also an illustrative example of sensor and tags variable function (sensor node). Ultra low power solutions are still in research phase and only prototypes are available [19]. One of the most challenging parts in low power or ultra-low power design is the security algorithm implementation while carrying all components in place. This is because of the number of cycles used for encryption and the power used by the light weight random number generator. In normal conditions more cycles are used to make the encrypted text more secure however in low power design it's a great challenge to design the simple polynomial combination to achieve same level of security.

Hardware architecture is briefly described as a reference for simulation architecture design. Before going into the details of architectural design of the simulator, we need a concrete list of process steps that defines the functionality of the simulator and later it will be easy to implement the list in simulator architecture.

5.3 Simulation Tasks and Requirements

The following important tasks are considered for the simulator:

1. RFID Tag Generation
2. Memory organization of created RFID tags
3. Tag energizing/de-energizing
4. Scenario designing in a WSN network
5. Capture data in each sensor node
6. Tag/sensor data collection and providing to base station

Table 5.1 is a small reference description of above stated tasks.

All the stated tasks are the major functions of the simulation environment. Sensor integration, data flow path and protocol integration on a node combines to make a smart node. Implementation of such a smart node in simulator design is discussed in Table 5.2.

5.3.1 Requirements and Issues for Integrated Sensor

There are several requirements and the issues that need to be included to create an integrated sensor structure of EPC global. Requirements for an integrated EPC global network sensor are as follows:

- Accommodation of a different type of functional node.
- Increasing importance of semantic modeling.
- Multiservice reader.

Table 5.1 Simulation tasks and their description

S. No	Task	Description
1	RFID Tag Generation	Sample RFID tags are generated by the user on which various simulator commands are executed later
2	Memory organization of created RFID tags	EPC and user memory contents are filled at this point. E.g. Serial number, EPC Object code, EPC Manager code
3	Tag energizing/de-energizing	Energize flag is set. It is not a part of actual architecture but only used in simulator to define the active tags
4	Scenario designing in a WSN network	A WSN scenario is created and passed to Matlab. It is simple sensor network consist of smart nodes. It covers the possibility of both IS and AdHoc communications
5	Capture data in each sensor node	Each smart node captures its neighboring RFID tags and stores in its memory
6	Tag/sensor data collection and providing to base station	Data stored in memory is transferred to the base station. The packets can follow either Adhoc mode path or Infrastructure mode path.

Table 5.2 Properties of RFID memory fields

S. No	Property	Default value
1	Slot counter	Random
2	Select flag	S0 inventory
3	Truncate	0 (No truncation)
4	State	−1, Non-energized
5	Password	(empty)
6	Kill password	(empty)
7	Access password	(empty)
8	CRC	Calculated CRC for EPC
9	Len	Length of data
10	RFU	00
11	EPC header	Provided by user
12	EPC manager number	Provided by user
13	EPC object code	Provided by user
14	EPC serial number	Serialized code for each tag
15	Zero fill	Zero padding

The needs and issues are classified specifically designed to EPC global network sensor integrated into two parts. One is how to identify and provide sensor data service or application information (data delivery). The work can be done in the delivery of required data without the knowledge of semantic data. The other is how to deliver data processing (data management) requesting the service entity to understand the implications of the data.

5.3.2 Requirements and Issues for Data Delivery

5.3.2.1 Sensors Plug and Play

EPC global network integrated sensor is a heterogeneous network. There may be a false identity cards, tags or labels Active sensor equipped with sensors. Data may come from the wireless sensor network through sensor BS (Base Station). In the case of the mark sensor, for example, the reader/writer needs to determine what type of sensor cards and read the sensor data requested, which his usually set in the memory of the user of the mark. Sensor data types can be categorized into static and dynamic. Sensor data "static" represents the length of the data and data types consistent throughout the operations. The signs include a fixed sensor function normally static data. Bio-sensing data, on the other hand, has the length of the variable data. Sensor function variable usually required to deal with the dynamic sensor data. It is therefore important to resolve the sensor tag memory scheme depending on the types of sensors.

5.3.2.2 Functional Decade of Public and Private

It is up to the industry's decision to delete or add software applications. Access of this system is limited or private. With the power estimation of RFID industry, it may expand the request to upstream and downstream channel affiliations. The data written needs to be shared among a group of nodes affiliated to strengthen its physical presence in the entire supply chain. That is, even in this system, which is part of the smart dust or cloud, data must be shared by affiliation, in the sense that it is publicly available to everyone. However, there are some systems (sub-systems) where it is applied openly. A good example can be found where the words represent geographic/physical locations. Software applications can be produced some specific geographic location and hold the request that needs to be added easily as an information/data service. Similarly, there is some Inspector General that performs same function over and over (Fig. 5.3).

5.3.2.3 Data Filtering

EPC and sensor data filtering (smoothing) may be required in order to transfer only the changes to the existing state and reduce communications overhead. The idea is powerful in that it may condense currents while transferring the same amount of information. The role of this "filter data" is an extension of the filter and the role of collecting network of EPC global. Since the sensor data source can be anywhere, and flexibility of data filtering site needs to be included.

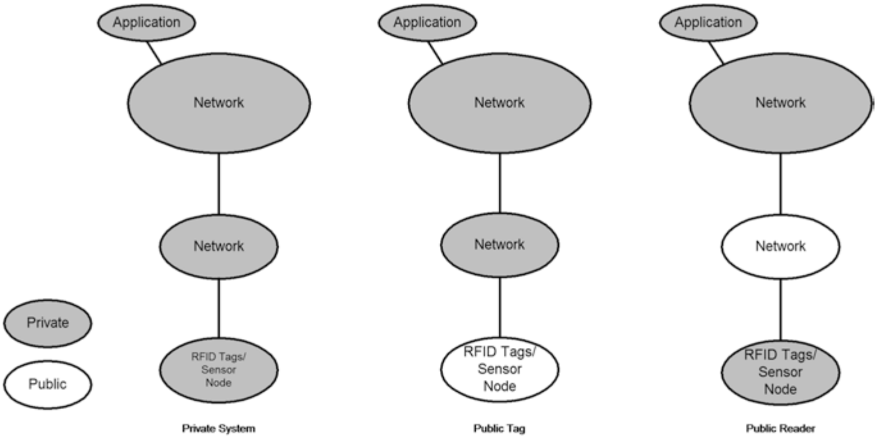


Fig. 5.3 Service providing domains

5.3.2.4 Enhanced Air Protocol

It has the ability to deal with short-sensing data exit sign read/write command. On other hand when it comes to big data, specifically on the BAP, sensor nodes and active tags are required to achieve the high-speed writing/reading. It is caused by extending the air from the protocol and increasing the bitrate of the mark, which is the simple increase in the hours or several bits forming technology. To read data quickly, consider FEC coding benefits from the gain [20].

5.3.2.5 Enhance the Standard Tag Data

Bio-sensing data is accommodated with user data that may be present in the sensor or sensor node. This practice is done where user data needs to be structured such that the corresponding investigator can investigate the beginning and the end of the address. Performance is achieved by identifying the basic scheme of memory for user data. In [21] brand memory is proposed to build the integrated form of indicators to sensor data written in the memory of the house. In [22], and provides a data structure tag, which can accommodate sensing data, featuring the smooth process with identity cards and mark sensor.

5.4 Implementation of Simulator

Protocol combinations are done at architectural level and all tag logics are simple binary patterned data (RZ or NRZ). Smart nodes are collaborated through smart dust with the base satiation and server selects its integrity in the security service layer provided by the tag. A detail of the implementation is presented in following parts.

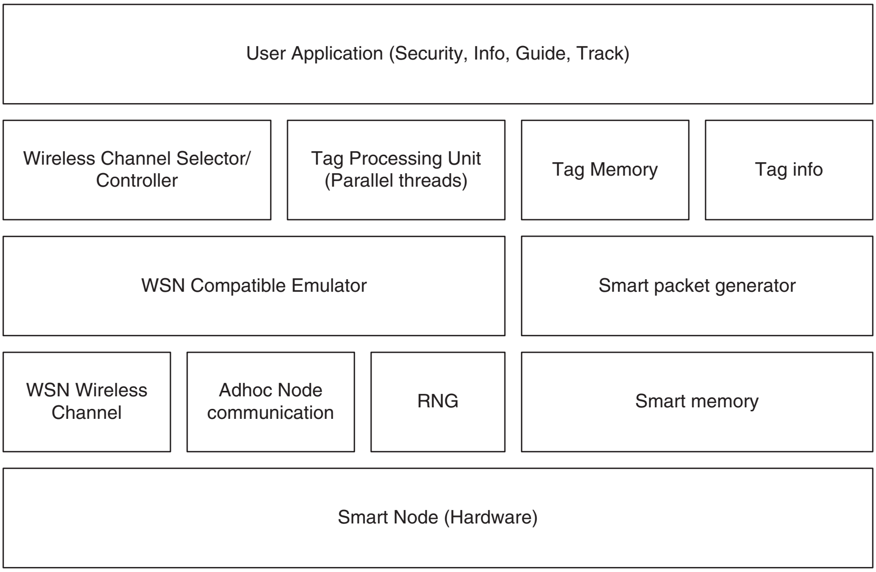


Fig. 5.4 General smart node architecture

5.4.1 Design

The WSN and RFID simulator are able to handle various smart tags at a time. Multiple jobs create multiple threads on services layer, which are then forwarded to multi-threaded queuing system. Figure 5.4 illustrates the basic block diagram of implemented architecture.

5.4.2 Data Flow

Sensed data is passed through several layers before it is able to be converted into smart memory architecture. This includes the separation of header and tag data. These fields are then processed from the Tag identity to alter the check sum. A RNG (Random Number Generator) serializes the packet to form a data stream. This binary data stream is then stored in the Smart memory. The data is now ready to be shared either publically or locally with private nodes. Some functional components and DFD is represented in Fig. 5.5.

DFD clearly explains the architecture and flow of data from the initial point to the end transmission of data towards base station. Simulation environment establishes global objects and defines the interfaces for basic components including the tags and the reader. Protocols like handshaking and data transfer is also introduced at this point. Followed by environment instantiation, tags are created.

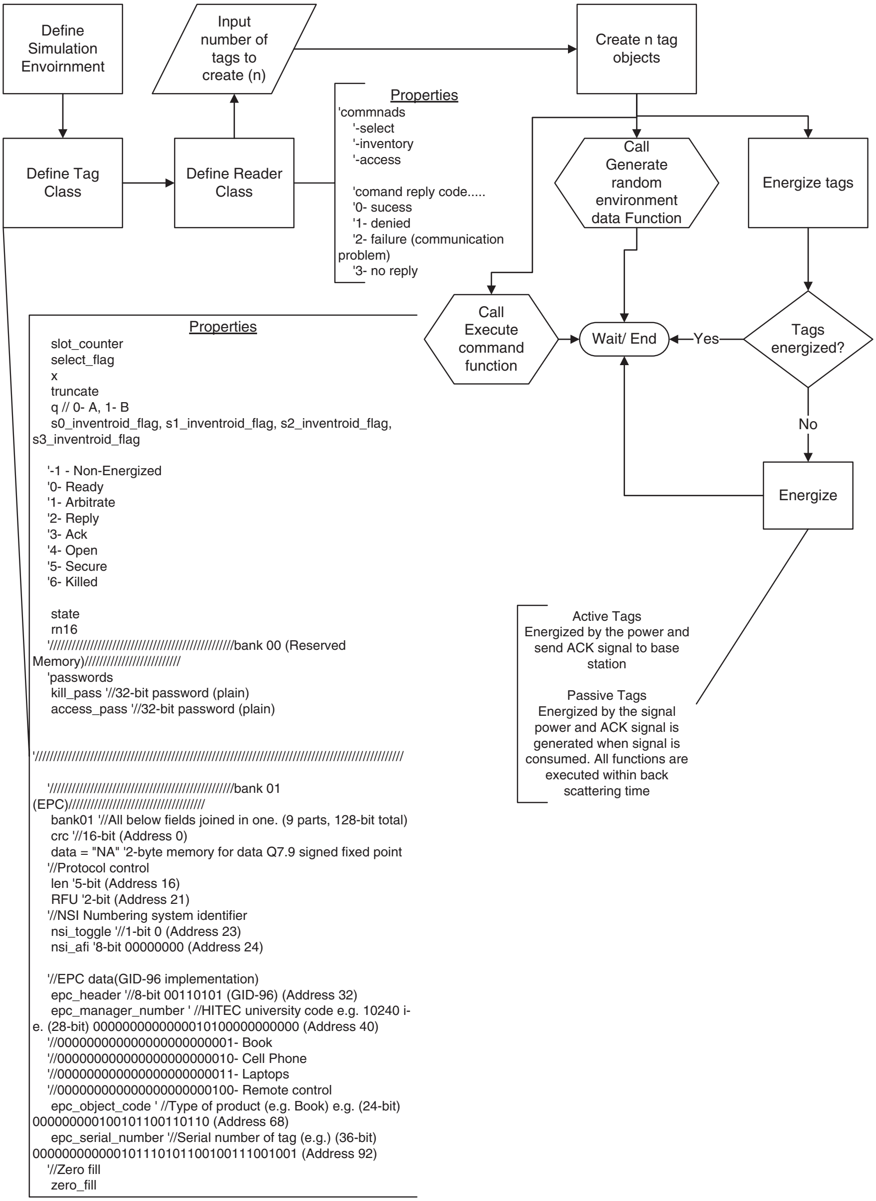
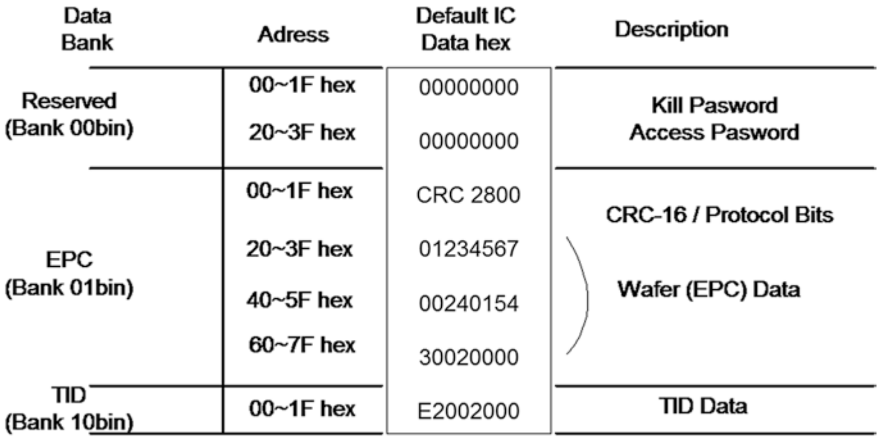

Define Simulation Envoirnment
Input number of tags to create (n)
Create n tag objects
Define Tag Class
Define Reader Class
Properties
'commnads
'-select
'-inventory
'-access
'comand reply code.....
'0- sucess
'1- denied
'2- failure (communication problem)
'3- no reply
Call Generate random environment data Function
Energize tags
Call Execute command function
Wait/ End
Yes
Tags energized?
No
Energize
Properties
slot_counter
select_flag
x
truncate
q // 0- A, 1- B
s0_inventroid_flag, s1_inventroid_flag, s2_inventroid_flag, s3_inventroid_flag
'-1 - Non-Energized
'0- Ready
'1- Arbitrate
'2- Reply
'3- Ack
'4- Open
'5- Secure
'6- Killed
state
rn16
'///bank 00 (Reserved Memory)////////////////////////
'passwords
kill_pass '//32-bit password (plain)
access_pass '//32-bit password (plain)
'//
'///bank 01 (EPC)///////////////////////////////////
bank01 '//All below fields joined in one. (9 parts, 128-bit total)
crc '//16-bit (Address 0)
data = "NA" '2-byte memory for data Q7.9 signed fixed point
'//Protocol control
len '5-bit (Address 16)
RFU '2-bit (Address 21)
'//NSI Numbering system identifier
nsi_toggle '//1-bit 0 (Address 23)
nsi_afi '8-bit 00000000 (Address 24)
'//EPC data(GID-96 implementation)
epc_header '//8-bit 00110101 (GID-96) (Address 32)
epc_manager_number ' //HITEC university code e.g. 10240 i-e. (28-bit) 0000000000000010100000000000 (Address 40)
'//000000000000000000000001- Book
'//000000000000000000000010- Cell Phone
'//000000000000000000000011- Laptops
'//000000000000000000000100- Remote control
epc_object_code ' //Type of product (e.g. Book) e.g. (24-bit) 000000000100101100110110 (Address 68)
epc_serial_number '//Serial number of tag (e.g.) (36-bit) 000000000000101110101100100111001001 (Address 92)
'//Zero fill
zero_fill
Active Tags
Energized by the power and send ACK signal to base station
Passive Tags
Energized by the signal power and ACK signal is generated when signal is consumed. All functions are executed within back scattering time

Fig. 5.5 DFD of smart tag (sensor)



Data Bank	Adress	Default IC Data hex	Description
Reserved (Bank 00bin)	00~1F hex	00000000	Kill Pasword
	20~3F hex	00000000	Access Pasword
EPC (Bank 01bin)	00~1F hex	CRC 2800	CRC-16 / Protocol Bits
	20~3F hex	01234567	Wafer (EPC) Data
	40~5F hex	00240154	
	60~7F hex	30020000	
TID (Bank 10bin)	00~1F hex	E2002000	TID Data

Fig. 5.6 Address distribution of RFID tag memory

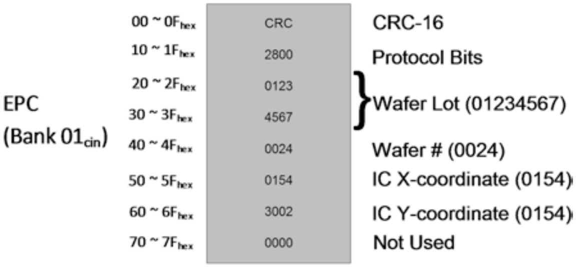
Simulator user can create N number of tags where N is the argument of the function called to Matlab server. Each of the tag has several properties which are also fairly explained in the DFD. Tags are created as following parameters by default [23].

The Random Number Generator provides the pseudo random numbers to the tag. This generator is based on a simple polynomial which is fixed for the simulator. For the real time implementation of the tag, simple polynomial RNG is used to save the power; as the power is an important factor in the design of a tag. The simulation architecture does not support active tags in this version. Zero padding is done at the end to make the tag compatible with generation 2 family architecture.

5.4.3 Memory Architecture

Class 1 Gen 2 tag sensor is an open standard currently available for reference. EPC supports several banks model for memory [24]. Most widely used is the bank 01 of memory which is also reflected in the tag architectural model. Tag data is also stored in this part. A tag can store basic information like object code, managerial code, access code and passwords. The complete memory architecture is stated below (Fig. 5.6).

Passwords are stored at the initial address of memory. This area is reserved in some cases, however, it is once or twice modifiable and in some cases it is only readable. Kill password is activated to flag the died tag. Access password let the managers secure reader entry in inventory state of tag. Between the four memory banks, EPC (Bank 01) is the place where data is stored in the tag. Details of EPC is as follow (Fig. 5.7).



00 ~ 0F$_{hex}$	CRC	CRC-16
10 ~ 1F$_{hex}$	2800	Protocol Bits
20 ~ 2F$_{hex}$	0123	Wafer Lot (01234567)
30 ~ 3F$_{hex}$	4567	
40 ~ 4F$_{hex}$	0024	Wafer # (0024)
50 ~ 5F$_{hex}$	0154	IC X-coordinate (0154)
60 ~ 6F$_{hex}$	3002	IC Y-coordinate (0154)
70 ~ 7F$_{hex}$	0000	Not Used

Fig. 5.7 EPC memory organization

CRC represents the integrity of the data. This is a 16 bit field out of total 96 bits. Protocol is defined initially which in this case is class 1 gen 2 tag communication protocol. A tag can represent several protocols however; it is easy to operate at a fixed one. Memory addresses from 0 × 20 to 0 × 6F are fixed for IC hardware properties. A reader can access this information to check the integrity of tag and the product general information.

5.4.4 Memory Organization

The memory architecture meets to the EPC Global™ Ultra High Frequency (UHF) Generation 2 specification. In manufacturing process, TI (Texas Instruments) probes every wafer and chip information which is programmed into the EPC. The info will be overwritten later when the chip is put into service.

5.4.4.1 Reserved Memory

This data is Bank 00 binary and contains the KILL password in locations 00 to 1F hex and the ACCESS password in locations 20 to 3F hex. These locations are shipped full of zeroes and unlocked. The passwords are only valid when programmed with non-zero values and (optionally) locked. When the passwords are locked they become unreadable and no-writeable. If the access password is zero; the IC will automatically transition to the Secured State rather than the normal Open state.

5.4.4.2 EPC Memory

This data is Bank 01 binary and contains a 16-Bit CRC which is calculated by the chip on the rest of the data in the EPC memory, 16-bits that are the Protocol Control bits and 96-bits that contain the EPC number. Texas Instruments programs unique wafer data, similar to the following, into the EPC field.

5.4.4.3 TID Memory

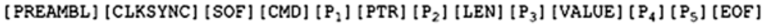
This data is Bank 10 binary and contains manufacture information. “E2” is prescribed in the EPC Gen 2 specification as a class identifier for EPCGlobal™, “002”, identifies the manufacturer as Texas Instruments and “000” is the IC revision number. TID memory is permanently locked.

5.4.4.4 User Memory

There is no “User Memory”.

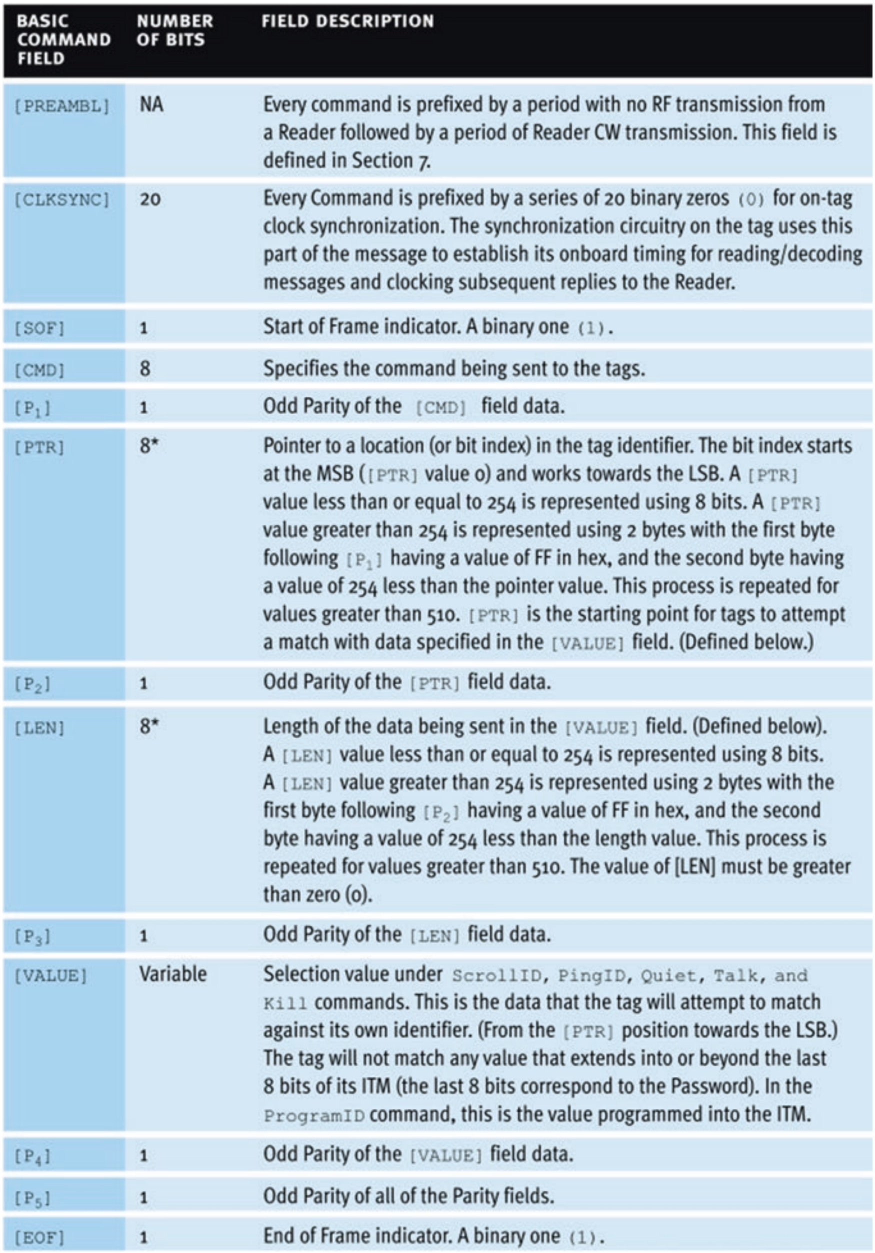
5.4.5 *Command Structure*

EPC Global class 1 Gen 2 sensors support following command architecture



[PREAMBL] [CLKSYNC] [SOF] [CMD] [P$_1$] [PTR] [P$_2$] [LEN] [P$_3$] [VALUE] [P$_4$] [P$_5$] [EOF]

Figure 5.8 presents a detail about each part of the command.



BASIC COMMAND FIELD	NUMBER OF BITS	FIELD DESCRIPTION
[PREAMBL]	NA	Every command is prefixed by a period with no RF transmission from a Reader followed by a period of Reader CW transmission. This field is defined in Section 7.
[CLKSYNC]	20	Every Command is prefixed by a series of 20 binary zeros (0) for on-tag clock synchronization. The synchronization circuitry on the tag uses this part of the message to establish its onboard timing for reading/decoding messages and clocking subsequent replies to the Reader.
[SOF]	1	Start of Frame indicator. A binary one (1).
[CMD]	8	Specifies the command being sent to the tags.
[P$_1$]	1	Odd Parity of the [CMD] field data.
[PTR]	8*	Pointer to a location (or bit index) in the tag identifier. The bit index starts at the MSB ([PTR] value 0) and works towards the LSB. A [PTR] value less than or equal to 254 is represented using 8 bits. A [PTR] value greater than 254 is represented using 2 bytes with the first byte following [P$_1$] having a value of FF in hex, and the second byte having a value of 254 less than the pointer value. This process is repeated for values greater than 510. [PTR] is the starting point for tags to attempt a match with data specified in the [VALUE] field. (Defined below.)
[P$_2$]	1	Odd Parity of the [PTR] field data.
[LEN]	8*	Length of the data being sent in the [VALUE] field. (Defined below). A [LEN] value less than or equal to 254 is represented using 8 bits. A [LEN] value greater than 254 is represented using 2 bytes with the first byte following [P$_2$] having a value of FF in hex, and the second byte having a value of 254 less than the length value. This process is repeated for values greater than 510. The value of [LEN] must be greater than zero (0).
[P$_3$]	1	Odd Parity of the [LEN] field data.
[VALUE]	Variable	Selection value under ScrollID, PingID, Quiet, Talk, and Kill commands. This is the data that the tag will attempt to match against its own identifier. (From the [PTR] position towards the LSB.) The tag will not match any value that extends into or beyond the last 8 bits of its ITM (the last 8 bits correspond to the Password). In the ProgramID command, this is the value programmed into the ITM.
[P$_4$]	1	Odd Parity of the [VALUE] field data.
[P$_5$]	1	Odd Parity of all of the Parity fields.
[EOF]	1	End of Frame indicator. A binary one (1).

Fig. 5.8 Command packet details

References

1. Kevin Emery thesis report on distributed eventing architecture: RFID and sensors in a supply chain. Viewed at MIT http://db.lcs.mit.edu/madden/html/theses/emery.pdf
2. University DM (2005) Final reference architecture and design guidelines for ELIMA-IMS
3. Deng H, Varanasi HD, Swigger M, Garcia K, Ogan O, Kougianos R (2006) Design of sensor-embedded radio frequency identification (SE-RFID) systems. In: Proceedings of the IEEE international conference on mechatronics and automation, pp 792–796
4. Ranasinghe DC, Leong KS, Ng ML, Engels DW, Cole PH (2005) A distributed architecture for a ubiquitous RFID sensing network. In: Proceedings of the international conference on intelligent sensors, sensor networks and information processing conference, pp 7–12
5. Zhang L, Wang Z (2006) Integration of RFID into wireless sensor networks: architectures, opportunities and challenging problems. In: Proceedings of the 5th international conference on grid and cooperative computing workshops, GCCW ‘06, pp 463–469
6. Alien viewed at: http://www.alientechnology.com/docs/AT_DS_BAP.pdf
7. Redemske R, Fletcher R (2005) Design of UHF RFID emulators with applications to RFID testing and data transport 4th IEEE Conference on Automatic Identification Advanced Technologies, pp 193–198
8. Opasjumruskit K, Thanthipwan T, Sathusen O, Sirinamarattana P, Gadmanee P, Pootarapan E, Wongkomet N, Thanachayanont A, Thamsirianunt M (2006) Self-powered wireless temperature sensors exploit RFID technology. IEEE Pervasive Comput 5(1):54–61
9. Cho N, Song S-J, Lee J-Y, Kim S, Kim S, Yoo H-J (2005) A 8-uW, 0.3-mm 2 RF-powered transponder with temperature sensor ISCAS, pp 4763–4766.
10. Meiners M, Sussenguth M, Missal W, Schary T, Benecke W, Lang W, Stucker M (2005) RF smart-sensor for venous ulcer treatment. In: Proceedings of the 13th international conference on solid-state sensors, actuators and microsystems TRANSDUCERS ‘05, vol 1, pp 453–456
11. Ubiquitous ID center home page viewed at: http://www.uidcenter.org/index-en.html
12. RFID Journal (2007) In Tokyo’s Shopping District, Auto-ID Tags Are the Latest Fad
13. RFcode Home page viewed at: http://www.rfcode.com/products.asp
14. Hokko sangyo home page viewed at: http://www.hokkosangyo.com/rfid.htm
15. NTT-AT NIRE home page viewed at: http://www.ntt-at.co.jp/product/nire/index.html
16. Mitsugi J (2006) Multipurpose sensor RFID tag. APMC 2006 workshop on emerging technologies and applications of RFID, WS04-4, pp 143–148
17. Mori (2006) UHF band RF circuits for RFID tag with battery supply. In: Proceedings of the IEICE 2006 general conference, CBS-1–6: pp S-11–12
18. Kobayashi S (2006) pT-engine project: the design challenge of ultra small and ultra low power node for sensor network. In: Proceedings of the international symposium on radio communications in ITRC Forum 2006, CARUT, INHA UWB-ITRC, HY-SDR ITRC, June 2006, pp 19–39
19. Philipose M, Smith JR, Jiang B, Mamishev A, Roy S, Sundara-Rajan K (2005) Battery-free wireless identification and sensing. IEEE Pervasive Comput 4(1):37–45
20. Intelleflex viewed at: http://www.intelleflex.com/pages/products.htm
21. Draft, ISO/IEC WD 24753 (2006) Information technology Radio frequency identification (RFID) for item management application protocol: encoding and processing rules for sensors and batteries
22. Osaka K (2007) Networked RFID middleware for sensor tag integration (acquisition of dynamic sensor data). In: Proceedings of the IEICE general conference
23. Accessed at http://ieee1451.nist.gov/
24. PowerID viewed at: http://www.power-id.com/Admin/fileserver.php?file=11

Chapter 6
Simulations Based Case Study and Analysis

Abstract This chapter explores the network-model, routing-model and the simulated results that have been designed for the analysis. Testing in real condition is not feasible so simulations have been used to virtually model the networks and do experiment on them. A worst case scenario has been simulated to analyse all probable outcomes.

Keywords Network · Simulation · Delay · Energy consumption · Query · Hop-count

6.1 Network Model

This section explores how network is analysed and implemented. The main goal is to implementing a random network which has some specific properties. A topology can be declared with the *net (num, a, range, nzone, seed*) command, where:

- net: Defines function for network model.
- num: Defines the number of nodes that are present in the network.
- a: Defines the area A×A of our network.
- range: Defines transmission range for the node.
- nzone: Is used for defining the number of zone.
- seed: Each seed gives us a specific topology.

All of this information will be chosen by user.

6.1.1 Node

The Network composes of different nodes. The number of nodes can be decided by user, but the location should be chosen randomly. This random place is produced by using a Matlab function which is called ‘*randi*’. By using this function we can generate random X and Y values between 0 and *a*, indicating the position of the node.

J. Ahmed et al., *RFID-WSN Integrated Architecture for Energy and Delay- Aware Routing*, SpringerBriefs in Applied Sciences and Technology,
DOI 10.1007/978-981-287-414-6_6

6.1.2 Zone

Whole network area can be divided to different regions which are called zones. Zones group a bunch of nodes and give them some common properties, for example rate at which energy is consumed. The number of zones will be defined by user and it is defined as a property of the node. How to divide the whole area into different zones such a way that no overlap occurs was challenging. Finally by using the '*voronoi*' function in Matlab, the problem has been solved.

To work with '*voronoi*' function first some random seeds should be produced for each zone. Each region seed has a random X and Y value that can be produced by function 'randi'. Then *voronoi* decomposes the space around each seed (i) into a region of influence R(i). The way that it divides the region is simple; it takes into account that all the points in one zone are more near to the seed of their zone than the seed of other zones.

6.1.3 Link

A link is used to connect two nodes. If the distance between two nodes is less than the transmission range then we can create a link between them. Each link in our model is a bidirectional link as shown in Fig. 6.1; this is implemented with two one-directional link. Each node has at least one in-Link and one out-Link. Each link has its own properties such as *Energy Cost*, *TXNode* and *RXNode*. For each link *TXNode* and *RXNode* shows which node is transmitter and which node is receiver of that specific link.

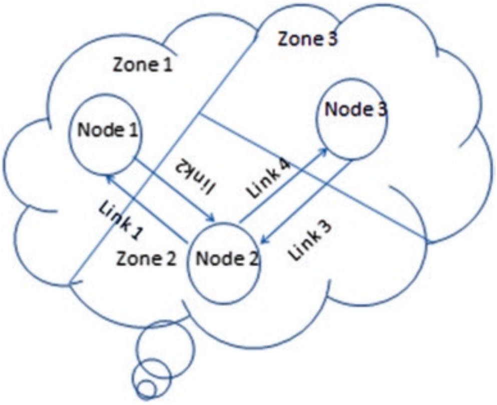


Fig. 6.1 Network model example

6.2 Routing Model

We intend to analyse how routing protocols behave in different working condition. For analysing our routing algorithms first we defined Cost to each link. Then by using some algorithms, found the shortest path cost from each node to the sink. Cost is different in two candidate routing algorithms.

6.2.1 Cost Function

For computing cost function in algorithms, should have some knowledge about packet energy, available energy and energy conservation rate.

6.2.2 Energy Consumption Rate

Energy Consumption rate of each node shows the rate of energy consumed for that node, this varies for different nodes in function of the zone to which they belong. Each zone in network has random value of consumption rate. So all the nodes are located in one zone has the same consumption rate.

6.2.3 Packet Energy

Packet energy defines the energy required for sending data to node *n*. Asynchronous MAC protocol which is suitable for WSN, a node with some data to transmit always wait for a beacon to start the transmission toward the beacon source. Node needs to wait for receiving beacon; this waiting time is called “Idle-Listening”. For sure in this period nodes need some energy to survive. Another factor which affects the packet energy is Transmission Cost which is the energy needed to send a packet. Prx and Ptx are properties of our transceiver.

6.2.4 Available Energy

This quantity shows the amount of energy that each node has before processing the packet.

6.2.5 Path Cost

After setting Cost to each link, the minimum path cost from each node to sink can be computed by using a recursive path cost function. In this function assumed that path cost of sink is zero, whenever a node receive beacon from other node recomputed the minimum path cost to the sink.

6.3 Simulation Model

In previous steps a way to model a network is described and defined how to compute path cost from each node to the sink. This section reports simulator architecture used in this research work and the transceiver features that have been used in simulation.

The simulator should be configurable in such a way to analyse some metrics which will be used later for experimental purpose, these metrics are:

- Energy consumed
- Delay

Simulator works base on the concept of event driven. For each simulation gives a set of random event. This event can be mainly three types:

- Data forwarding
- Beacon transmission
- Data transmission

In simulator function first we initialize the network and variables after that we initialize the simulation parameter like; condition when the simulation should stop and when the routing path should be recomputed.

Simulator stops to work in two cases: if it reaches to the maximum time of simulation or one of the nodes die. We configure that node die when the available energy of a node goes below of zero. To update the routing path, update threshold has been defined so whenever the available energy of battery goes below of this threshold the routing algorithm recomputed the path.

6.4 Case Study Scenario for Simulation

The traffic messages are simulated to a sink *s* from a sender node *n*, the traffic gets interfered with cross-traffic that is coming from other nodes as shown in Fig. 6.2. A simulation setup is chosen with variation in cross-traffic parameters and path length:

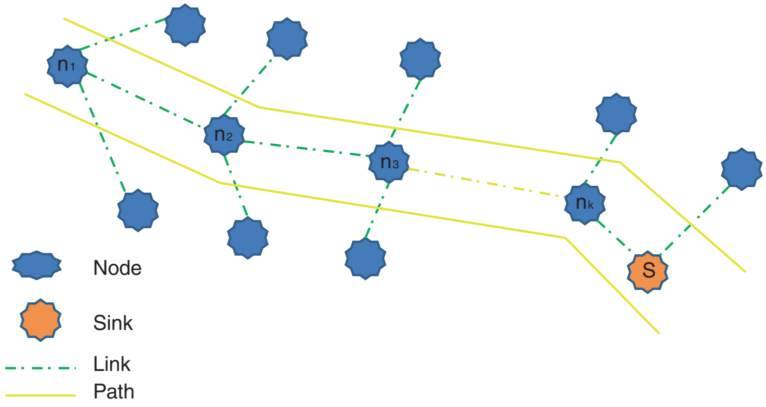


Fig. 6.2 Simulation scenario

- Experiments with varied path length: $|r_p| = \{5, 10\}$
- Sink node receives messages periodically by source node with $T = 30$ s,
- Estimated parameters are aggregated at each intermediate link while messages are communicating between n and s (Eq. 4.6).

Value for α is selected as: $\alpha = 0.9$

- The real end-to-end transmission latency experienced by each message is captured at seconds,
- There are two neighbours for every hop in the routing path, forwarded cross traffic follows a Poisson distribution with parameter $\lambda = \{60\text{ s}, 120\text{ s}, 480\text{ s}, 1200\text{ s}\}$
- $N = 100$ and $M = 20$
- Radio transceiver RFM TR1100 is used for radio interface.

Therefore, different combinations are made combining the parameters and simulations are done accordingly. Every iteration has a specific simulation time period. Notice that the route building process has not been considered yet. The purpose of the simulations is evaluating the estimation of end-to-end latency distribution, obtaining the pdf and validate the method.

6.5 Analysis of Case Study Simulation

Currently two routing algorithms are chosen for analysis. Since the idea of energy aware and delay aware routing in WSN is new, the experiments have been planned step by step. After each step, the data are analysed to plan the next step. Therefore each step gave a new idea on how to continue with experiment.

Table 6.1 Constant parameters in simulations

A (m)	1000
Range (m)	400
nZone	20
N	100
M	20
Ptx (W)	0.0538
Rrx (W)	0.0425
R (kbps)	500 × 1024
L (bits)	16 × 8
Battery capacity (J)	1.476

The simulator was prepared to show some analysis metrics. These metrics can be classified as:

- Average energy consumed in answering a query
- Delay

The analysis parameters are totally dependent to the configuration of the network, therefore by changing the network configurations it can be seen how these metrics will change in each of the two algorithms.

Main interest is to figure out how parameters like beacon rate, different data traffic and number of nodes affect the analysis metrics. The data traffic can be changed tuning the sense period. Sense period is the total period that each node waits for the beacon, receive the beacon and send data (Table 6.1).

6.5.1 Energy Model

The assumption has been made that each active node is capable of forwarding the queries that are resolved to another node that lie within the distance of d hops, making d transmissions in total. Hence whenever a query M is answered the average amount of energy spent according to Eq. 4.9 is given as:

$$E_{avg} = \left(cE_{update} + d\right)S_M + \alpha$$

Where the Energy parameters are given in Table 6.2.

6.5.2 Delay Model

The sample mean $\bar{x}$ and sample variance s^2 are calculated. The distribution is characterized on the basis of these two parameters. Sample mean is considered to

Table 6.2 Energy parameters

E_{avg}	Average energy consumed while answering the query
C	Amortization factor
E_{update}	Energy consumed while local update
D	Look-ahead parameter
S_M	Average number of steps to answer a query of size M
α	Expected number of hops

Table 6.3 Delay parameters

γ	0.57721
β	Transmission latency
D_M	Estimated delay
ρ	Parameter considering past values
μd_M	Expected value of end-to-end latency

be a really good delay estimator when it comes to shorter and sample variance roughly indicates quality of link. Hops with high variance may be experiencing a higher number of retransmissions. Some other parameters are given int Table 6.3.

6.6 Recommendation of Models on the Basis of Simulation

In this research an equally sophisticated mathematical model has been developed allowing us to characterize and evaluate the network performance analytically (in terms of delay and energy costs) of energy aware routing protocol ACQUIRE, as well as alternative technique Directed Diffusion, taken as standard. As per our knowledge, there is not much literature available similar to the proposed mathematical model.

Initially for simplicity of analysis, we have described, modeled and analyze a very basic version of the ACQUIRE mechanism in this research work. While our analysis is based on the assumption that a regular grid topology has been considered, the results obtained after simulating the proposed model can be modified for topologies other than grid type.

ACQUIRE, when comes to compare with other alternative routing protocols seems to beats all other strategies, keeping its parameters values optimum. The reason behind that performance would be that it is especially designed for complex, one-shot queries, even when the other schemes too are enhanced with cached updates.

Generally on getting a query, the first thing a sensor x does is update information locally. If the current information it has got is not updated one, x sends a request to all sensors that lie within the distance of d hops. This request is forwarded hop by hop. The information got by sensors is then forwarded to x. Let the energy consumed in this phase be E_{update}. Then after answering the query based on the information obtained, x then forwards the remaining query to a node that is chosen randomly from those d hops away. Subsequently the update is only triggered when the information sensor node as got is not up-to-date; now the question arises how to quantify how many times such up- dates will be triggered. This update frequency is modeled by an average amortization factor c, such that whenever every c query occurs at a given node, update is likely to occur. In other words the cost of the update at each node is amortized over c queries, where $0 < c \leq 1$. For example, if on average an up-date has to be done once every 10 queries, c = 0.1 and if the update occurs every 200 queries, c = 0.005. When the query forwarded by active node is completely resolved, the complete response is forwarded to source node by last. Where α is used to denote the expected number of hops from the node where the query is completely resolved to $x*$.

Figure 6.3 shows the plot between the average amount of energy consumed while answering a query verses look ahead parameter *d,* keeping amortization factor c constant, thus the update is not triggered in whole transmission showing that there is no change in the information the source is getting from the neighboring nodes lying at *d* hopes away.

Now, if d = D, where D is the diameter of the network, the entire query can be resolved in one step by $x*$ without being forwarded to any other node. Though, in

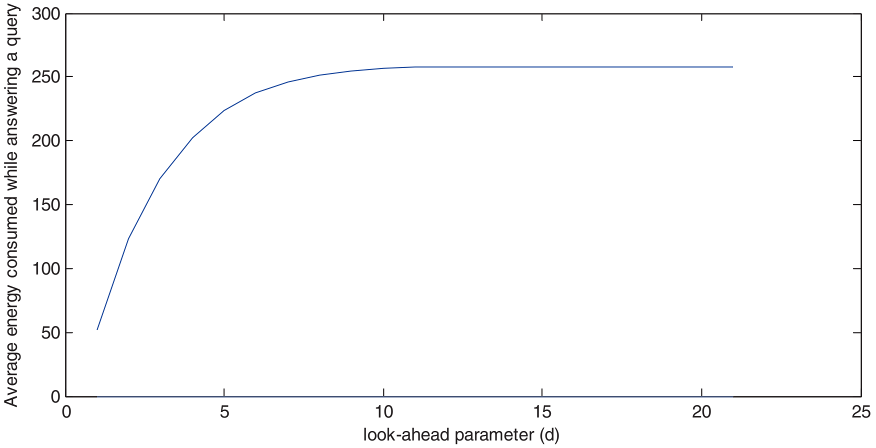


Fig. 6.3 Plot between average energy consumed while answering a query and look-ahead parameter d, keeping c = constant

this particular case, the value of E_{update} will be larger considerably. On the other hand, if d is made too small, a substantially larger number of steps S_M will be required. In general, S_M reduces with increasing d, while E_{update} increases with increasing d. It is therefore possible, depending on other parameters, that the optimal energy expenditure is incurred at some intermediate value of d.

In special case when look-ahead parameter $d = 0$, which is the case where network is performing random walk. Hence, if $d = 0$, there will be no requests for updating sent by querier node $x*$ and it will try to resolve the query with cached information. After doing its part, querier forwards rest of query to some other randomly chosen neighbouring node.

Figure 6.4 shows the results of average amount of energy consumed while answering a query verses the look-ahead parameter d at different values of amortization factor c. the uppermost plot in the figure shows the results when $c = 1$, means information is triggered every single query. In this particular case the amount of average energy consumed answering a query is quite large. Results are plotted by varying the values of c where $0 < c < 1$. In the give figure five different values are plotted by changing the value of c by the difference of 0.02, means the behavior is analyzed when information is triggered every 20 queries making $c = 0.05$. The results show that amount of energy consumed in answering the query is decreasing while c as amortization factor is decreasing. Energy has minimum value when $c = 0$ verifying the direct relation of energy consumed and amortization factor. This is quite true actually because change in information triggers the update, making all the sensor nodes participate to provide their

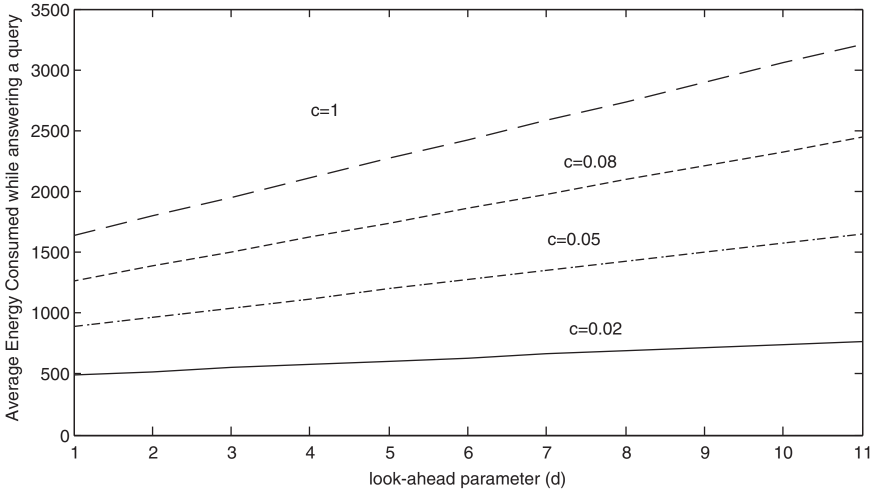


Fig. 6.4 Plot between average energy consumed while answering a query and look-ahead parameter d, at different values of c

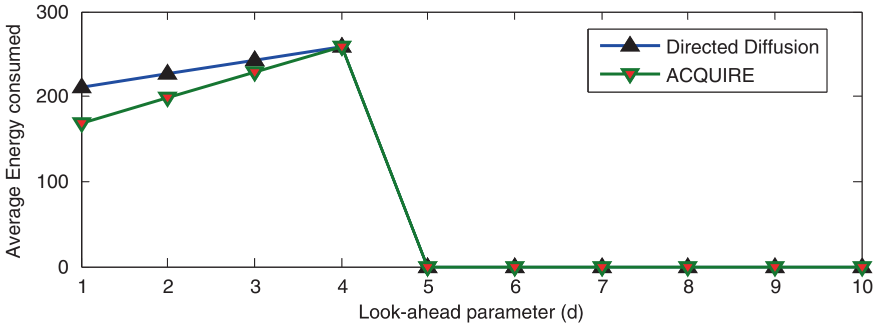


Fig. 6.5 Energy comparisons of ACQUIRE and directed diffusion

information. This requires larger amount of energy making less energy available for further transmission of data, making it highly un-recommended for larger networks and deviate from its basic idea of energy aware routing protocol.

Plot shown in Fig. 6.5 is the energy comparison for analysis between two routing algorithms ACQUIRE and Directed Diffusion verses look-ahead parameter *d.* The plot clearly shows that the average amount of energy consumed while answering the query in ACQUIRE is less than that of directed diffusion at lower values of *d.* At higher values of d like in above graph as the value of d reaches 5, the energies in both routing protocols overlaps. What happens actually is that as d increases, number of hops updating the source node increases making a lot of energy wasted in collecting the information. Thus a time comes where the energies of both the routing protocols overlaps. Thus showing ACQUIRE is much better option than directed diffusion for smaller values of *d*. even for the larger networks ACQUIRE outperforms directed diffusion in energy point of view (Fig. 6.6).

To further analyse the results obtained from simulations, when $|r_p| = 5$, Fig. 6.5 shows the comparison between pdf of standard Normal (N(0,1)) and pdf of simulated normalized distribution. Central point is shown at t = 1.5 s and it's at the higher tail on right side. The normalized data set is shown in blue at probability almost equal to 1. Both effects are related to each other and can be explained by the nature of the experiment measurements. In fact, the values represented come from measured end-to-end delays. This is important to know that there is a clear limit on the possible values from the left side (i.e. time delays cannot be negative). Looking at the range of absolute values, we observe that with a mean sample value of 4.5 ms very few messages achieved a delay less or equal than 2 ms and the distance between the minimum value and the mean is approximately of 5 ms. However, on the right side, this distance goes up to around 34 ms, with a maximum value close to 40 ms. Notice that the β

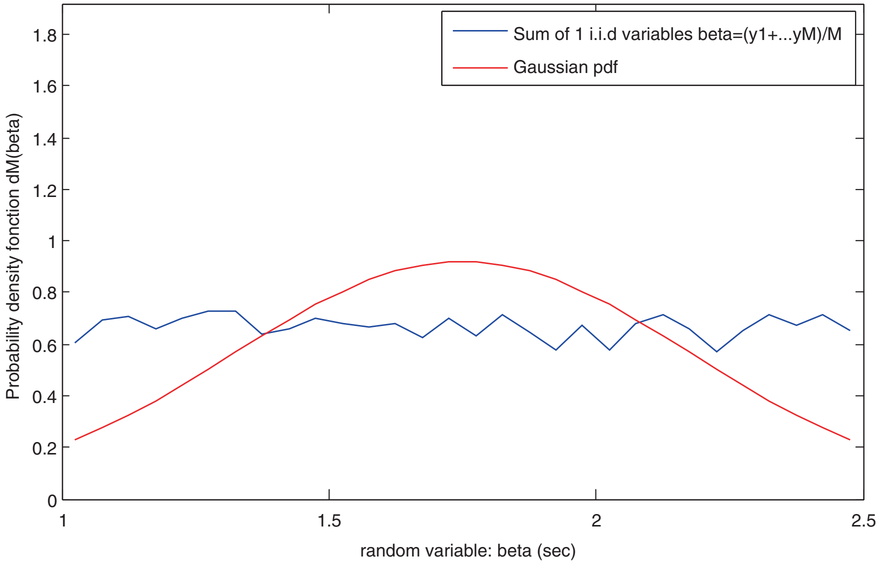


Fig. 6.6 Illustration of data samples converged through central limit theorem

value performing the Exponential Weighted Moving Average is responsible, in a certain way, of this effect. A lower β acts as a filter for higher sampled values and hence, reduces the tail on the right side. However, this would also affect the sample variance s^2 as values would get closer to each other. Thus, a side effect would be a distortion on the estimated distribution which would look thinner. On the other hand, higher values of β would reduce the smoothing effect of the Exponential Weighted Moving Average and estimate a better value for the sample variance. This would definitely reflect on the peak of the estimated distribution, although, at the same time, produce a thicker distribution shape.

Figure 6.7 shows the graph between delay and time in seconds. The graph shows a clear increase in delay as the factor t is increasing making β increase which makes sample mean increased eventually. According to delay model proposed in Eq. 4.5, this increase in sample mean value increase the delay. According to graph the delay is increased up to certain extent then there is a clear fall in the plot. This is based on the fact that ACQUIRE is query based routing protocol and query is resolved as the number of hops increases and a point comes where whole query is resolved that point is destination point. Thus as number of hops are increasing and required query is closer to get fully resolved, the delay decreases because t, the time between the nodes, decreases.

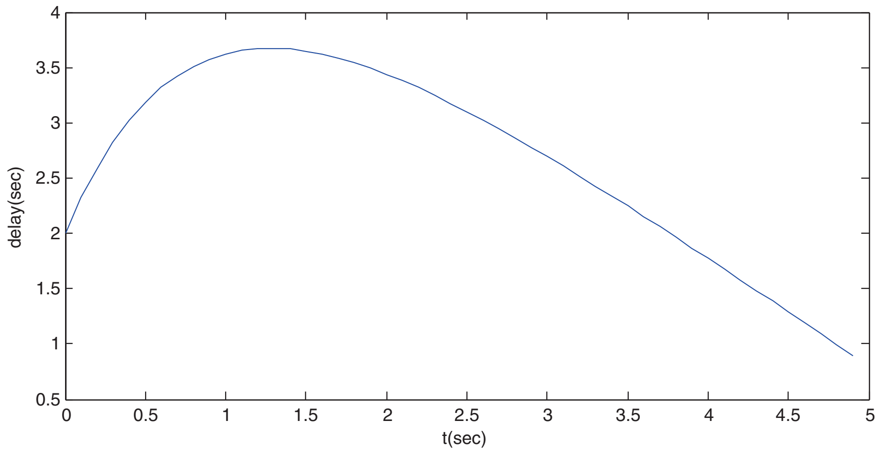


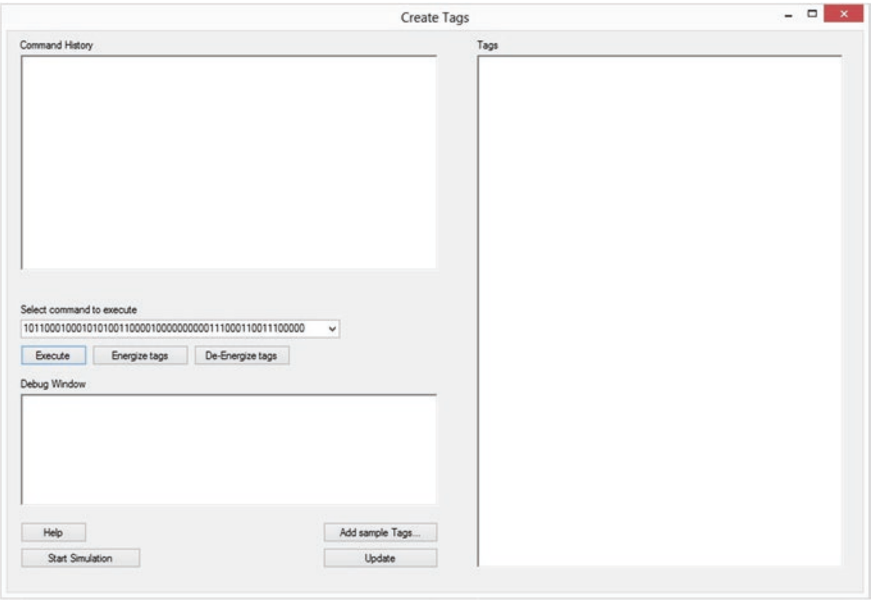
Fig. 6.7 Plot between delay and time

6.7 Analysis of Proposed Simulator

In this study a simulator is developed having the capability of simulating the environment that integrates a sensor node with a RFID tag, also called Smart Node. Figures 6.8 and 6.9 show the window for execution of commands for tags generation.

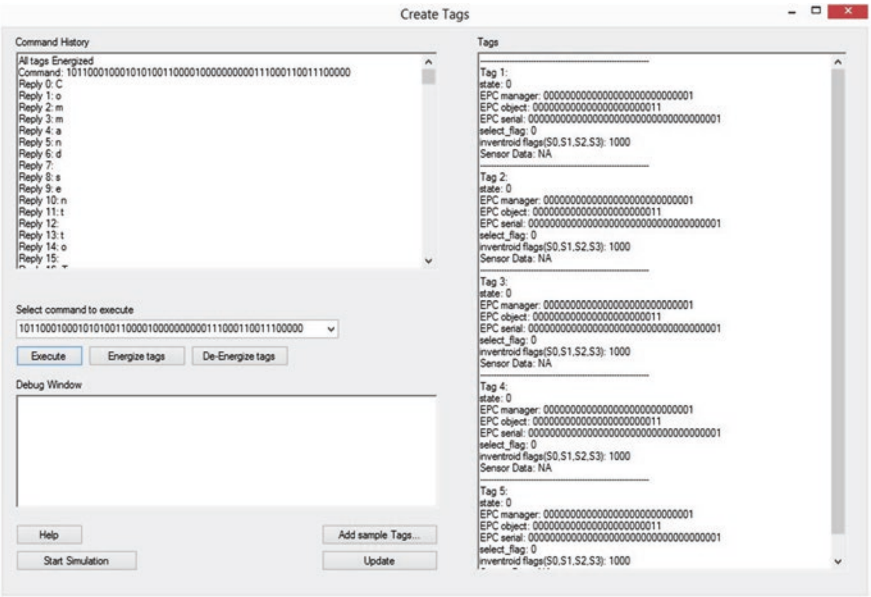
Scenario creation is a user defined function in this simulator. In the “Create Tags” window, user can select from various options to generate a specific simulation environment. Details of the options are as follows

S. no	Property	Description
1	Command History	I/O log
2	Tags	Status of each tag
3	Select Command	Custom/predefined command selection
4	Execute	Deploy command to all tags
5	Energize tags	Energize created tags, set status $= 0$
6	De-Energize tags	De-energize created tags, set status $= -1$
7	Debug window	Communication viewer
8	Start Simulation	Start simulating generated scenario
9	Add sample tags	Add tags to the scenario
10	Update	Sync to matlab server



Select command to execute: 101100010001010100110000100000000000111000110011100000

Fig. 6.8 Command window

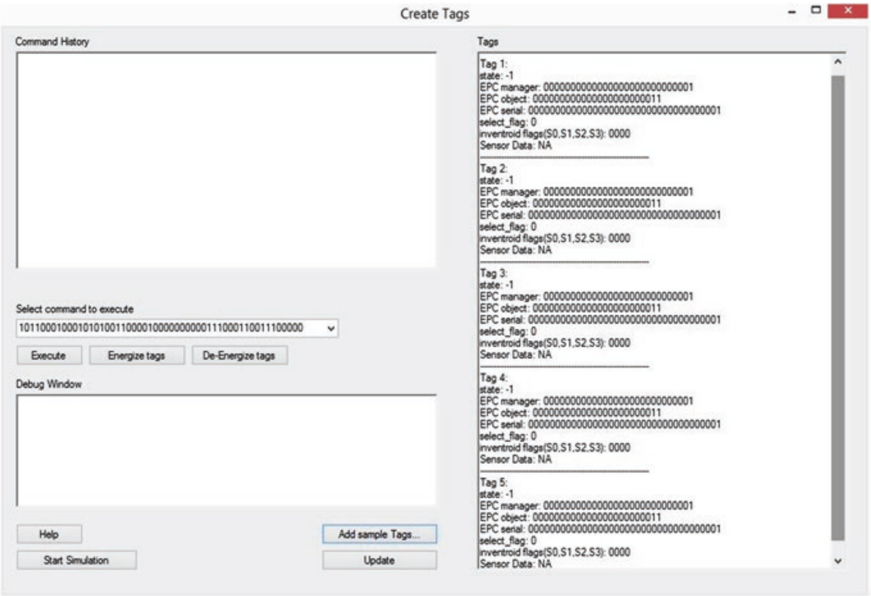


```
All tags Energized
Command: 101100010001010100110000100000000000111000110011100000
Reply 0: C
Reply 1: o
Reply 2: m
Reply 3: m
Reply 4: a
Reply 5: n
Reply 6: d
Reply 7:
Reply 8: s
Reply 9: e
Reply 10: n
Reply 11: t
Reply 12:
Reply 13: t
Reply 14: o
Reply 15:
```

Select command to execute: 101100010001010100110000100000000000111000110011100000

```
----------------------------------------
Tag 1:
state: 0
EPC manager: 0000000000000000000000000001
EPC object: 000000000000000000000011
EPC serial: 000000000000000000000000000000000001
select_flag: 0
inventroid flags(S0,S1,S2,S3): 1000
Sensor Data: NA
----------------------------------------
Tag 2:
state: 0
EPC manager: 0000000000000000000000000001
EPC object: 000000000000000000000011
EPC serial: 000000000000000000000000000000000001
select_flag: 0
inventroid flags(S0,S1,S2,S3): 1000
Sensor Data: NA
----------------------------------------
Tag 3:
state: 0
EPC manager: 0000000000000000000000000001
EPC object: 000000000000000000000011
EPC serial: 000000000000000000000000000000000001
select_flag: 0
inventroid flags(S0,S1,S2,S3): 1000
Sensor Data: NA
----------------------------------------
Tag 4:
state: 0
EPC manager: 0000000000000000000000000001
EPC object: 000000000000000000000011
EPC serial: 000000000000000000000000000000000001
select_flag: 0
inventroid flags(S0,S1,S2,S3): 1000
Sensor Data: NA
----------------------------------------
Tag 5:
state: 0
EPC manager: 0000000000000000000000000001
EPC object: 000000000000000000000011
EPC serial: 000000000000000000000000000000000001
select_flag: 0
inventroid flags(S0,S1,S2,S3): 1000
```

Fig. 6.9 Command execution in command window



Select command to execute: 1011000100010101001100001000000000001110001100111100000

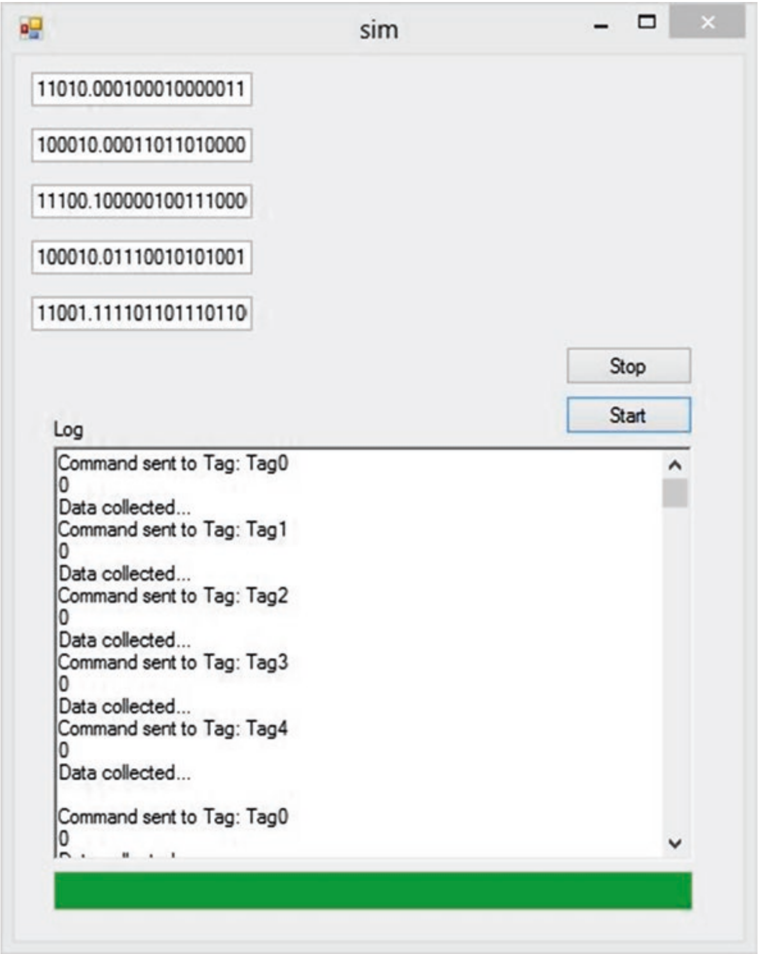
```
Tag 1:
state: -1
EPC manager: 0000000000000000000000000001
EPC object: 000000000000000000000011
EPC serial: 000000000000000000000000000000000001
select_flag: 0
inventroid flags(S0,S1,S2,S3): 0000
Sensor Data: NA
------------------------------------------------
Tag 2:
state: -1
EPC manager: 0000000000000000000000000001
EPC object: 000000000000000000000011
EPC serial: 000000000000000000000000000000000001
select_flag: 0
inventroid flags(S0,S1,S2,S3): 0000
Sensor Data: NA
------------------------------------------------
Tag 3:
state: -1
EPC manager: 0000000000000000000000000001
EPC object: 000000000000000000000011
EPC serial: 000000000000000000000000000000000001
select_flag: 0
inventroid flags(S0,S1,S2,S3): 0000
Sensor Data: NA
------------------------------------------------
Tag 4:
state: -1
EPC manager: 0000000000000000000000000001
EPC object: 000000000000000000000011
EPC serial: 000000000000000000000000000000000001
select_flag: 0
inventroid flags(S0,S1,S2,S3): 0000
Sensor Data: NA
------------------------------------------------
Tag 5:
state: -1
EPC manager: 0000000000000000000000000001
EPC object: 000000000000000000000011
EPC serial: 000000000000000000000000000000000001
select_flag: 0
inventroid flags(S0,S1,S2,S3): 0000
Sensor Data: NA
```

Fig. 6.10 Tags creation

Following is the sample window that shows 5 sample tags in a scenario. Each tag is representing a smart node object. Properties of each tag is stated next to it. By default the properties are set to their initial values. Status of each tag is −1 i.e. discharged by default (Fig. 6.9).

Once the tags are generated which are integrated RFID tag and Senor node, they will communicate with the base station to exchange the retrieved data obtained from the simulated environment. Figures 6.10 and 6.11 shows the creation of tags and data capturing respectively.

The simulator is able to handle various smart tags at a time. Multiple jobs create multiple threads on services layer, which are then forwarded to multi-threaded queuing system. The simulator results can be viewed using MATLAB.



```
11010.000100010000011
100010.00011011010000
11100.100000100111000
100010.01110010101001
11001.111101101110110
```

```
Command sent to Tag: Tag0
0
Data collected...
Command sent to Tag: Tag1
0
Data collected...
Command sent to Tag: Tag2
0
Data collected...
Command sent to Tag: Tag3
0
Data collected...
Command sent to Tag: Tag4
0
Data collected...

Command sent to Tag: Tag0
0
```

Fig. 6.11 Data capturing

Figure 6.12 shows the results/data collected by the base station of a sample scenario. Each colour represents one smart node. This is a temperature data collected from a wide space in a geographical area. Temperature value is randomly

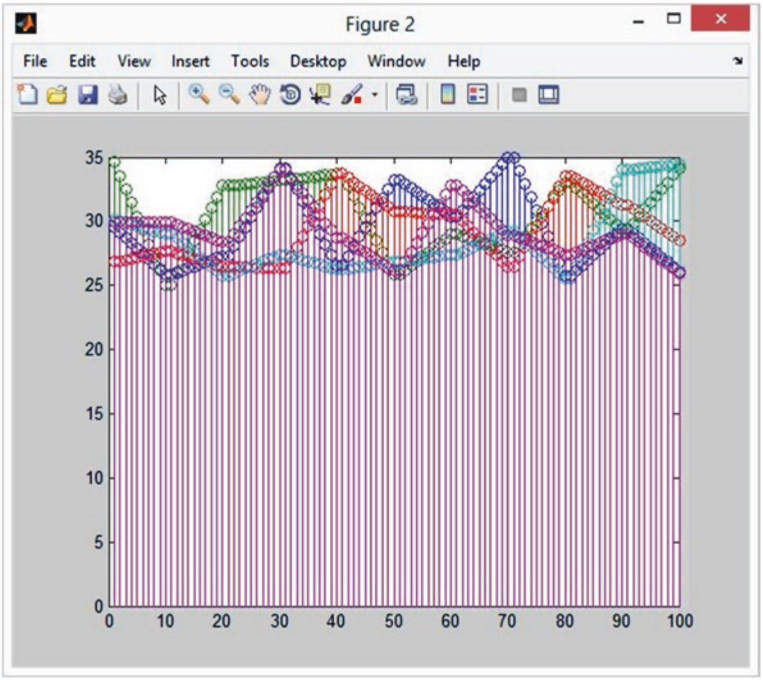
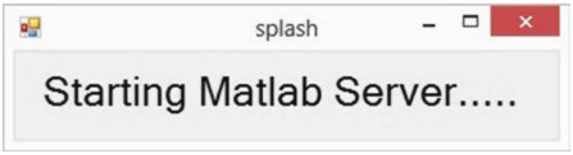


Fig. 6.12 Simulator results

varying from 25° to 35°. One discrete value of the stem is a captured packet of sensor data. In smart node this value is stored in EPC memory and transmitted to the base station. This is a collection of data over longer period hence there is variance present in the data (Fig. 6.13).

Fig. 6.13 Splash



Starting Matlab Server.....

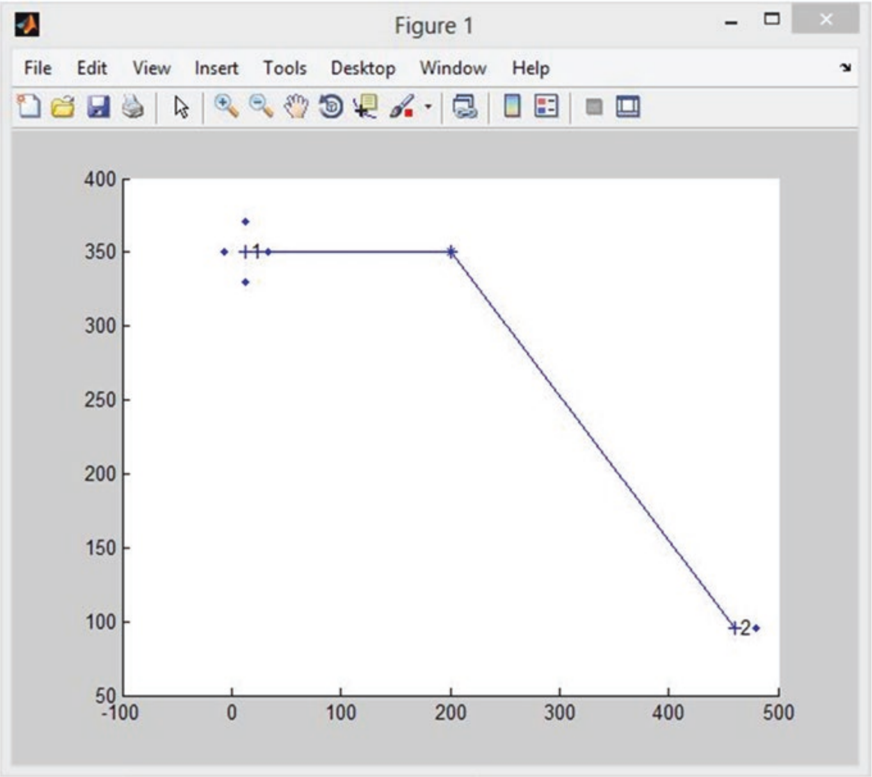


Fig. 6.14 Visualization of data transfer on MATLAB

Node to node communication is based on Ad Hoc network. Each communication signal changes the link color to red, which shows a busy channel. The stars in Fig. 6.14 shows the sensors and smart node enables it to communicate it to other smart nodes or directly to the base station. Some basic network properties like link bandwidth and maximum distance are set to average values.