Signals and Communication Technology

Shuang-Hua Yang

Wireless Sensor Networks

Principles, Design and Applications

Springer

Signals and Communication Technology

For further volumes:
http://www.springer.com/series/4748

Shuang-Hua Yang

Wireless Sensor Networks

Principles, Design and Applications

Springer

Shuang-Hua Yang
Department of Computer Science
University of Loughborough
Loughborough
UK

ISSN 1860-4862 ISSN 1860-4870 (electronic)
ISBN 978-1-4471-5504-1 ISBN 978-1-4471-5505-8 (eBook)
DOI 10.1007/978-1-4471-5505-8
Springer London Heidelberg New York Dordrecht

Library of Congress Control Number: 2013950085

Printed on acid-free paper

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

Dedicated to my family—Lili, my beautiful wife, Bob and James, my two brilliant sons

and

As a remembrance of my father, Mr. Xinsheng Yang, who passed away in August 2010

Preface

Wireless sensor networks (WSNs) are more and more frequently seen as a solution to large-scale tracking and monitoring applications, because of their low-data-rate, low-energy-consumption, and short-range link network which provides an opportunity to monitor and control the physical world to a previously unprecedented scale and resolution. The deployment of a large number of small, wireless sensors that can sample, process, and deliver information to external systems such as the satellite network or the Internet, opens many novel application domains. Potential WSN applications include industrial control and monitoring, home automation and consumer electronics, security and military sensing, asset tracking and supply chain management, intelligent agriculture and health monitoring. MIT classified WSNs as one of the ten emerging technologies that will change the world. Internet of Things (IoT), which is technically supported by WSN and other relevant technologies, has been classified as a national economic development strategy by the Chinese Government in 2009. Research in WSNs has mainly concentrated on energy consumption, routing, fault tolerance, data acquisition, and operating systems, particularly focusing on collecting and aggregating data from specific networks with an associated sink node, called a WSN gateway. Some work has been carried out on the connection of different disparate sensor networks for a single or multiple applications. Some of the most documented research challenges are attributed to issues relating to scalability, reliability, security, coverage, and massive deployment.

This book is concerned with the design and application challenges of ZigBee based WSNs, which we experienced firsthand in our research and development work over the past few years. A principle aim has been to include in the book a comprehensive coverage of topics suitable for use in university courses. This book is the result of nine Ph.D. theses and a number of public funded projects completed under my supervision. A significant aspect of this book is the presentation to the readers of enough technical details to enable them to actually repeat the work rather than merely understanding the principle involved. I hope that it will be a valuable reference book for industrial design as well as for university teaching and academic research. I believe that this broad targeted audience is an attractive feature of this book, as most of the very limited selection of WSN books currently available were written primarily for academic researchers or as a textbook, presenting the fundamental basic concepts while providing, little guidance on how to

carry out the actual design process. This book is unique in bringing together wireless communication principles with actual WSN design processes. It will enable readers to become increasingly capable in exploiting fully the new technologies described here in their research or industrial work.

This book consists of 15 chapters grouped into three parts. Part I (Chaps. 1 and 2) provides the principle of WSNs. Part II (Chaps. 3–9) focuses on providing solutions to various design issues. Chapters 10–12 in Part III explore the application technologies of WSNs in indoor location tracking, logistics management, and Internet of Things (IoT), followed by Chaps. 13 and 14, two real applications to home automation and building fire safety. Chapter 15 forms the conclusion.

Target Audience

The book can serve both as a textbook and a reference book. The primary target audience for this book is the university student community. The materials included in this book have been used several times as a handout for teaching Master of Science (M.Sc.) modules on WSNs. Resulting student feedback has been addressed in the book. The secondary audience for this book is the research and development community. This includes both academia from universities and research institutes together with industrial developers. It can also be used as a reference book for any readers, who are interested in getting insight into the WSN area but have been unable to find a any sources of real-life WSN designs.

Acknowledgments

Many people have directly or indirectly contribute to the work presented in this book. Chapters 3–12 were produced based on the theses of my former Ph.D. students. They are Dr. Fang Yao, Dr. Xin Lu, Dr. Hesham Abusaimeh, Dr. Yanning Yang, Dr. Khusvinder Gill, Dr. Tareq Alhmiedat, Dr. Huanjia Yang, and Dr. Ran Xu, and my nearly completed Ph.D. student Mr. Md Zaid Ahmad. Prof. Bokia Xia, Dr. Yuanqing Qin, and Mr. Guizheng Fu, my former academic visiting scholars, Mr. Donato Salvatore, my former research assistant, Ms. Weiwei He and Mr. Hakan Koyuncu, my on-going Ph.D. students, have also contributed to the work. I am extremely thankful to their hard work and cooperation. I would like to express my deep appreciation to my industrial collaborators from the consortiums SafetyNET (DCSI, Sure Technology, Jennic, Arqiva, and ASFP), IndeedNET (Advantica, Sure Technology, EMHA), and iNET (IDC), and my academic collaborators, Prof. Wan-Liang Wang at Zhejiang University of Technology, Prof. Chunjie Zhu at Huazhong University of Science and Technology, Prof. Xuemin Tian, Prof. Bokia Xia at Petroleum University, Prof. Ping Li at Liaoning Shihua University, Prof. Jie Chen at Beijing Institute of Technology, and

Prof. Hongyong Yuan at Tsinghua University, and Prof. Min-Hong Wu at Derby University. There are too many to name here. I would also like to thank the TSB project monitoring officers Mr. Guy Hirson (SafetyNET) and Mr. Mike Patterson (IndeedNET) for their constructive guidance in our research. My appreciation also goes to my colleagues in the Computer Science Department at Loughborough University for their enthusiasm and dedicated assistance they have provided me.

My gratitude goes to my colleague Dr. Roger Knott, and Ms. Charlotte Cross (Springer-Verlag) for their proof reading and to my formal Ph.D. student, Dr. Ran Xu, for his graphic expertise.

Finally, I gratefully acknowledge the financial supports from the Technology Strategy Board through Technology Program (TP/J3521A, TP/3/PIT/6/I/16993), Carbon Connection Trust, and European Regional Development Fund through Transport iNET program, EPSRC through Transforming Energy Demand through Digital Innovation (TEDDI) call in Energy Program (EP/I000267/1), Natural Science Foundation of China through Major International Joint Research Program (61120106010), and Santander Program for Mobility of Young Faculty and Researchers operated by Tsinghua University.

July 2013 Prof. Shuang-Hua Yang

Contents

Chapter 1
Introduction

Keywords Wireless communication · Wireless sensor networks

1.1 Wireless Communication Technologies

Computer networks have become an essential part of our world on which daily life, business, and education, rely heavily. Networks make information and services available to anyone on the network, regardless of the physical location of the resources or the users. Computer networks are divided into many types such as Personal Area Network (PAN), Local Area Network (LAN), Metropolitan Area Network (MAN), and Wide Area Network (WAN). As indicated by the names, a PAN is a computer network organized around an individual person. LANs are used to connect computers in a small area such as in one building or a number of buildings. While the network that connects computers inside a city or town is called a MAN. A WAN connects many numbers of computers over a large area such as a country or a continent. Conventionally, these network communication links are wired, i.e. they use physical cables connecting the different network devices. Wired computer networks allow for reliable data transmission, but the wiring required necessitates high installation cost, and in many cases is inconvenient. Wireless communication technologies provide the obvious solution to overcome these obstacles, although they have their own set of challenges such as interference, reliability and others.

Wireless Networks connect any devices or computer using radio waves, infrared, or any other wireless media. It can cover a large area, in which case it will be called a Wireless WAN, or it can cover a small area or a building, in which case it will be called a Wireless LAN (WLAN). Alternatively, it can provide an interconnection of information technology devices within the range of an individual person, in which case it will be called Wireless PAN (WPAN). A low-rate wireless personal area network (LR-WPAN) is a network designed for low-cost and very low-power short-range wireless communications.

S.-H. Yang, *Wireless Sensor Networks*, Signals and Communication Technology,
DOI: 10.1007/978-1-4471-5505-8_1,

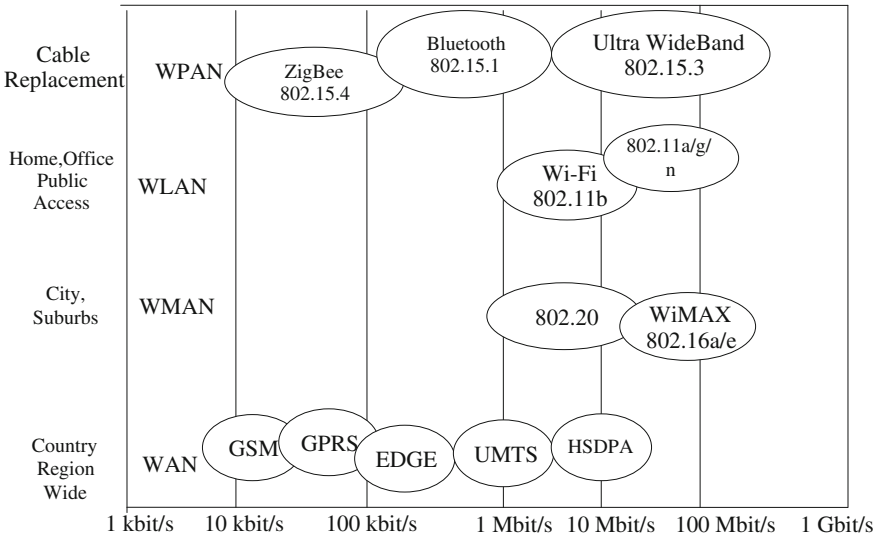


Fig. 1.1 Wireless communication standards and their characteristics (Benini et al. 2006)

There are various wireless communication standards existing, including ZigBee, Wi-Fi, WiMax, GSM (Global System for Mobile Communications) et al. Fig. 1.1 shows a brief comparison of operating characteristics of various wireless communication standards. These standards are categorized according to the supported throughputs, communication range and application areas. Standards such as Wi-Fi, WiMAX, UltraWideBand, and 802.11a/g/n are normally used for high data throughput applications, and generally require a main power supply. Systems constructed on the basis of GSM, General Packet Radio Service (GPRS), Enhanced Data Rate for GSM Evolution (EDGE), Universal Mobile Telecommunications System (UMTS) and High-Speed Downlink Packet Access (HSDPA) are designed to achieve full mobility. The Bluetooth standard was mainly developed to replace computer interconnection cables. The ZigBee standard was developed for wireless sensor networks.

1.2 Wireless Sensor Networks

Wireless sensor networks (WSNs) are a group of specialized autonomous sensors and actuators with a wireless communications infrastructure, intended to monitor and control physical or environmental conditions at diverse locations and to cooperatively pass their data to a main location and/or pass their control command to a desired actuator through the network (Yang and Cao 2008). We narrow the scope of the WSNs in this book by limiting the data communication to low data

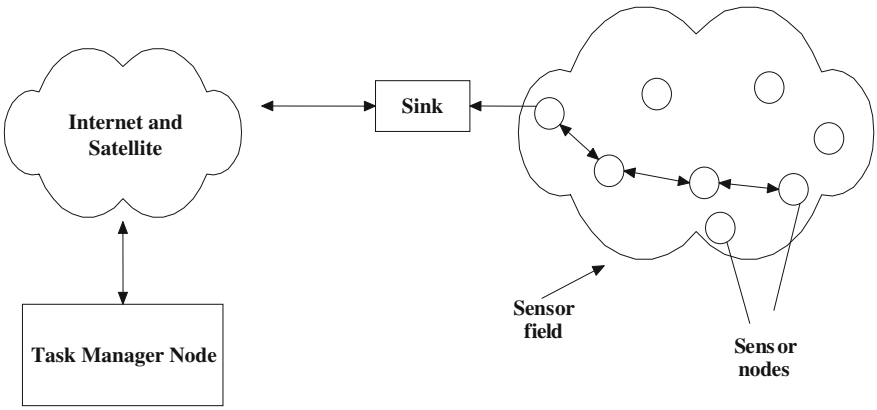


Fig. 1.2 Structure of a typical wireless sensor network (Akyildiz et al. 2002)

rates and short communication ranges, and the individual sensor node to physically small, low power, and low cost devices. A WSN consists of multiple nodes, ranging from a few to several hundreds or even thousands, where each node is connected to one or more other nodes. Nodes may be designed for carrying out one or more of the following functions—sensing, relaying data, or exchanging data with an outside network. A node for sensing is called a sensor node, one for relaying data a router, and one for exchanging data with other networks a base station, or sink node, which is similar to a gateway in a traditional network.

Every sensor node is equipped with a transducer, a microcontroller, a radio transceiver and a power supply, usually a battery. The transducer generates electrical signals based on sensed natural phenomena and environmental changes. The microcontroller processes and stores the sensing output. The radio transceiver with an internal antenna or connection to an external antenna receives commands from a central computer and transmits data to that computer. Figure 1.2 illustrates the concept of WSNs, where the data is collected from a sensor node and then transmitted to a sink node, which is connected to the Internet or a satellite network. Through the Internet and the satellite network the collected data is finally received by an application. Sensor nodes do not have to have a fixed location and most of them are randomly deployed to monitor a sensor field. Sensor nodes usually communicate with each other via an on-board radio transceiver.

1.3 Application Areas of WSNs

WSN applications can be classified into two categories: remote monitoring and mobile object location tracking. Both categories can be further divided into indoor and outdoor applications. Figure 1.3 gives a classification of possible applications of WSNs, which is similar in structure to that given by Yick et al. (2008). Military

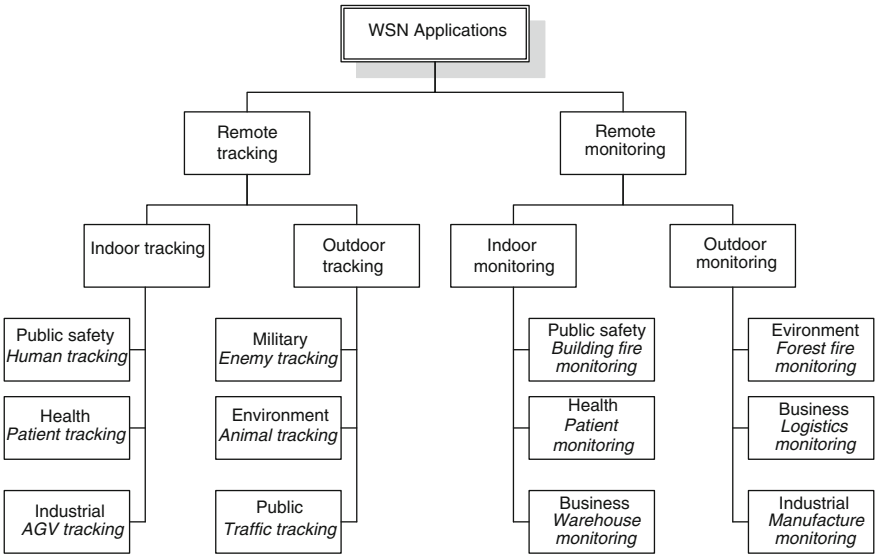


Fig. 1.3 Overview of WSN applications

applications include monitoring friendly forces, tracking enemy movement, checking the equipment status, or detecting any nuclear, biological or chemical attack. Environment applications include tracking the movement of animals, detecting forest or building fires, and sensing or detecting any chemical materials leakage. Commercial/logistic applications include vehicles and objects tracking, inventory monitoring et al.

Unlike mobile object location tracking applications, which need real-time updating of the tracking results, remote monitoring applications of WSNs measure the specific environment conditions periodically and send sampling data or warnings mainly in three modes:

- Periodically at a predefined time interval;
- As the result of a specific event, this often happens when the value of a specific measurement reaches a predefined threshold;
- In response to interrogation from a user.

1.4 Challenges in the Design and Implementation of WSNs

An important feature of the above WSN applications is the capability for the easy installation of a massive number of wireless sensor nodes. This feature triggers all of the design and implementation challenges normally surrounding wireless

communication, together with other challenges unique to the particular applications. The main ones are energy efficiency, interference, security, data management, and large-scale deployment. The design and implementation of WSNs must deal with all of these issues.

The problem of energy efficiency can be addressed in different ways. One approach is the optimization of both the hardware and the embedded software design, including routing algorithms, which minimizes the energy consumption and thus makes a WSN efficient. This book addresses the challenge in energy efficiency by optimizing power management at both the hardware component and the network levels.

Interference caused by other wireless systems working on a similar frequency band and co-existing in the same vicinity can greatly reduce the performance of WSNs. Ordinary interference avoidance mechanism might be not efficient for a large-scale WSN because of the constraints of WSNs such as its low computation capability. This challenge will be addressed in this book with a full discussion of such limitations. Some practical guidance for deploying wireless sensors networks will be provided.

Security risks are unavoidable for WSNs due to its wireless nature. Proper mechanism must be in place to protect healthy data distribution from any attack. Normally, the data transmitted over WSNs will have been encrypted and WSNs security management services will be in place. This book provides a solution for ensuring system level security, particularly focusing on remote Denial of Service (DoS) attacks.

When large amounts of data are generated over time, the cost of transferring all of such sensor data to a sink node is expensive. Data compression and aggregation techniques aid in reducing the amount of data transferred. The use of a robust strategy to manage distributed data flow, query and analysis is important to sensor networks. This book will address the challenge to data management resulting from both reducing the amount of data to be transferred, while improving the distributed capability of in-network data processing, i.e. rather than sending large amounts of raw data to the base station, a local sensor node's storage space is used as a distributed database to which queries can be sent to retrieve data.

A wireless sensor network often consists of a large number of sensor nodes in order to provide the effective sensor field required. They can easily cover a relatively wide geographical area. This characteristic makes it impossible for users to manually maintain the whole network. A comprehensive management architecture is required to monitor the WSNs, configure network parameters and implement system updating. Scalability issues can degrade system performance when the size of WSNs increases. The applications presented in this book detect serious problems with large-scale implementations. Such implementations only work effectively when the number of node is restricted to less than a hundred after which the congestion and extreme routing cost significantly slow down the data communication and eventually discontinue the system operation. This challenge is addressed through the application technologies section of the book.

1.5 Aims of the Book

This book is designed as a reference book or textbook for final year undergraduate and postgraduate students as well as researchers of wireless communication technologies. It is also useful for software and system engineers, company managers, and IT professionals who intend to implement WSNs. Thus, it sets out to explore and examine the principle, design and implementation issues of WSNs, looking at design processes and real applications in the area. This book differs from other books in the area where the IEEE 802.15.4 standard and ZigBee standard are further explained but which lack explanation and demonstration at the system level (Elahi and Gschwender 2009). This book is also distinct from those books where theoretical research results on limited independent topics are presented but lack the design and implementation focus (Misra et al. 2009). The aim of this book is to enable the readers, to design and implement WSNs for their own applications, after they have finishing reading this book.

References

Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., Cyirci, E.: A survey on sensor networks. IEEE Commun. Mag. **40**(8), 102–114 (2002)

Benini, L., Farrella, E., Guiducci, C.: Wireless sensor networks: Enabling technology for ambient intelligence. Microelectron. J. **37**(12), 1639–1649 (2006)

Elahim, A., Gschwender, A.: ZigBee Wireless Sensor and Control Network. Person Education, USA (2009)

Misra, S., Woungang, I., Misra, S.C.: Guide to Wireless Sensor Networks. Springer, Berlin (2009)

Yang, S.H., Cao, Y.: Networked control systems and wireless sensor networks: Theories and applications. Int. J. Syst. Sci. **39**(11), 1041–1044 (2008)

Yick, J., Mukherjee, B., Ghosal, D.: Wireless sensor network survey. Comput. Netw. **52**(12), 2292–2330 (2008)

Chapter 2
Principle of Wireless Sensor Networks

Keywords IEEE 802.15.4 · ZigBee · 6LowPAN · Wireless sensor networks

2.1 Introduction

Wireless sensor networks are a subset of wireless networking applications, which focus on enabling connectivity without, the need, generally, of wires to connect to the sensors and actuators (Gutierrez et al. 2004). Due to the length of the name "wireless sensor and actuator networks" or "wireless sensor and control networks", most people have adopted the shorter "wireless sensor networks" instead. In any case, it is important to remember that the design of this type of network is meant to collect information from wireless sensors and send control commands to actuators attached to the wireless network.

Sensor and actuator networks have existed for decades. Computer based control systems are a typical hardwired sensor and actuator network. As shown in Fig. 2.1, sensors and actuators are connected with a central computer or control terminal via a data bus system or other networks and implement control and monitoring functions. This type of hardwired sensor network is simple and reliable, and often seen in industrial control such as process control and manufacturing production control. Because of the involvement of large amount of cabling in the installation, wired sensor networks are hard to extend. The installation cost of hardwired sensor networks is high, which takes in the form of cabling, labor, material, testing, and verification. Furthermore, cables require connectors that can become loose, lost, misconnected, or even break. This problem is commonly known as the last meter connectivity problem and is called this due to the analogous problem in a wide area network.

The use of large number of hardwired sensors networked to a system brings considerable complexity to the system, including cabling deployment, power supply, and configuration, making it impossible in many cases such as forest monitoring and battlefield surveillance. Recent Integrated Circuit (IC) and Micro

S.-H. Yang, *Wireless Sensor Networks*, Signals and Communication Technology,
DOI: 10.1007/978-1-4471-5505-8_2,

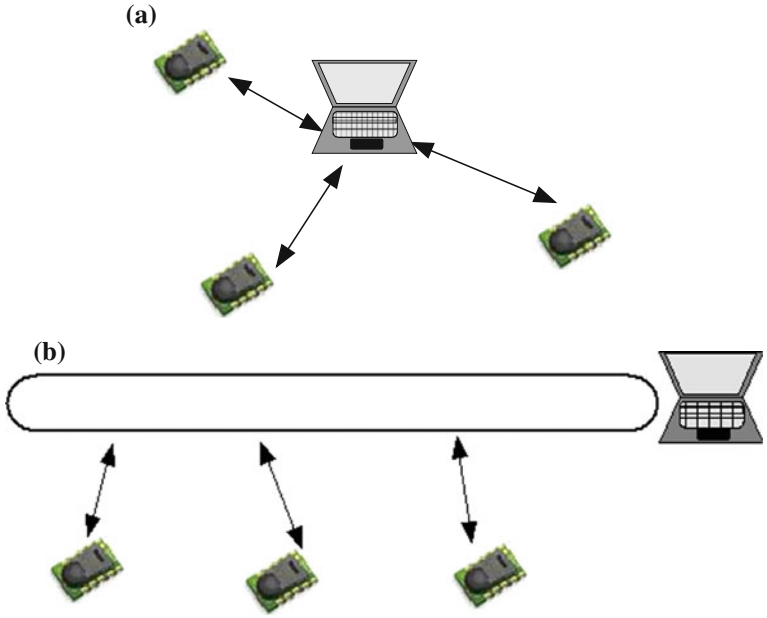


Fig. 2.1 Hardwired sensor and actuator network: **a** Star hardwired sensor and actuator network. **b** Data bus hardwired sensor and actuator network

Electro Mechanical System (MEMS) have matured to the point where they enable the integration of wireless communications, sensors and signal processing together in a single low-cost package, named as a sensor node (Schurgers and Srivastava 2001). Such a sensor node is equipped with data processing and communication capabilities. A set of such sensor nodes forms a wireless sensor network. It is now feasible to deploy ultra-small sensor nodes in many kinds of areas to collect information. The sensing circuitry measures ambient condition related to the environment around the sensor and transforms them into measurable signals. After necessary processing, the signals are sent to a pre-defined destination via a radio transmitter. All of these operations are powered by batteries to ease deployment, since a traditional power supply (i.e. mains power) may not be available.

This type of wireless solutions for sensor networks combines flexible connectivity with ease of installation. The scope of sensors determines the range of applications of wireless sensor networks. There are many types of wireless sensors depending upon the type of sensing required (Lewis 2004; Akyildiz et al. 2002):

- Temperature;
- Humidity;
- Acoustic waves;
- Vehicular movement;
- Lighting condition;
- Pressure;

- Soil makeup;
- Noise levels;
- The presence or absence of certain kinds of objects;
- Mechanical stress levels on attached objects;
- The current characteristics such as speed, direction, and size of an object.

Moreover, there are many applications for the wireless sensor networks, including the following:

- Continues sensing for environmental and condition monitoring;
- Event detection for disaster response;
- Location sensing for mobile target tracking and localization;
- Local control for home automation, industrial automation etc.

Because the reliability and security of hardwired networks can be higher than that of wireless communication systems, wireless sensor networks are not recommended to replace hardwired sensor networks. It is expected that hybrid networks, wired and wireless, will coexist. Wireless sensors will act as extension to wired networks whenever the wireless capability adds value to the applications (Gutierrez et al. 2004).

If we consider only wireless sensor networks with low cost, low energy consumption, low data rate, and short communication range, IEEE 802.15.4 will be the most commonly used communication standard in the design of such wireless sensor networks. ZigBee and 6LowPAN are two most widely adopted IEEE 802.15.4 based communication protocols. This chapter will introduce IEEE 802.15.4 as the foundation of wireless sensor networks and then describe ZigBee and 6LowPAN as two typical wireless sensor networks. A comparison of ZigBee and 6LowPAN will be given at the end of the chapter.

2.2 IEEE 802.15.4 Standard and Wireless Sensor Network

2.2.1 OSI and WSN Stacks

The Open Systems Interconnection (OSI) seven-layer model, proposed by the International Organisation for Standardisation (ISO), forms the basis for the design of the WSN protocol stack. However, unlike the seven-layer OSI model, that consists of the physical layer, the data link layer, the network layer, the transport layer, the session layer, the presentation layer and, the application layer, the WSN protocol stack does not adopt all the seven layers of the OSI model. In reality, the seven-layer OSI model has too many layers making it overly complex and difficult to implement (Aschenbernner 1986). Consequently, the protocol stack adopted by WSN consists of only five layers, as shown in Fig. 2.2.

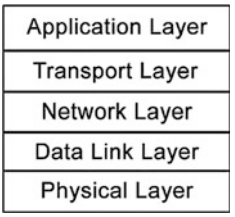


Fig. 2.2 WSN protocol stack

The five-layer WSN protocol stack consists of the physical layer, the data link layer, the network layer, the transport layer and the application layer. Each layer is designated a specific set of task to perform independently of the other layers in the protocol stack.

The first layer of the protocol stack, the physical layer, is responsible for defining and managing the connections between individual devices and their communication medium. The physical layer is responsible for frequency selection, carrier frequency generation, signal detection, and modulation and data encryption. Moreover, the physical layer defines the type of connectors and cables compatible with the communication medium.

The second layer of the protocol stack, the data link layer, is responsible for providing services that allow multiple nodes to successfully access and share a communications medium. These services include medium access control, reliable delivery, error detection and error correction.

The third layer of the protocol stack, the network layer, is responsible for establishing the communications paths between nodes in a network and successfully routing packets along these paths. The requirements of different routing protocols can vary and the choice will influence the communication paths set up. Some routing protocols will favour communication paths that help the WSN to deliver the best Quality of Service (QoS), other energy saving protocols may choose the path that enables the WSN to achieve the best lifetime while other will use a hybrid of both objectives.

The fourth layer, the transport layer, is responsible for providing a higher-level layer of the protocol stack and consequently providing the users with transparent and reliable communications between end-users. There are varying forms of transport layer protocols; two of the most popular and contrasting are the transmission control protocol (TCP) and the user datagram protocol (UDP). Connection oriented transport layer protocols, such as TCP, provide a reliable communication service, with extensive error handling, transmission control, and flow control. Whereas, connectionless transport layer protocols, such as UDP, provide an unreliable service but with minimum error handling, transmission, and flow control.

The fifth and final layer, adopted by most WSN, is the application layer. The application layer resides close to the users of the system. There are many potential applications implemented at the application layer including, Telnet, Hypertext Transfer Protocol (HTTP), File Transfer Protocol (FTP), or Simple Mail Transfer Protocol (SMTP). In terms of WSN, the application layer programming primarily

deals with the processing of sensed information, encryption, the formatting and storage of data. Moreover, the application layer scans the underlying layers to detect if sufficient network resources and services are available to meet the user's network requests.

2.2.2 Overview of IEEE 802.15.4 Standard

The embedded software design of the wireless sensor networks needs to rely on some standards to ensure that the network system is functional on different hardware platforms. Current standards can be simply divided into two categories public or private, according to the design purpose. Manufacturers of wireless sensor networks will complete the bottom layer development (wireless modulation/demodulation module, MAC layer and network layer protocols, etc.) using the selected standard. Then the developers will build their own applications on top, after purchasing the products from the manufacturers. It is not correct to say that a single standard can suffice for all features required for the wireless sensor networks. Actually, there is no unified standard existing for the concept of WSNs. The existing standards, especially for private standards, usually focus on the specified applications, which might reduce the support available elsewhere. For example, if a standard enables the product to provide a long system lifetime, the support provided for data throughput may be comprised.

The public standards have a much better balanced performance on the above issues than the private standards as their targets are to adopt as much supports from the manufacturers as possible. Any development of a public standard will take into consideration many possible aspects in order to ensure the maximum compatibility. Private standards have a faster development progress than public standards since they only need to improve the content of the standard for their own purpose. However, as indicated by their name, private standards may not be available for public access.

The IEEE 802.15.4 standard (2003) is explicitly designed as a new Low-Rate Wireless Personal Area Network (LR-WPAN) standard for applications that require low data throughout and have limited resource of power and computation capability. It aims to overcome the problems associated with the existing standards such as WiFi and Bluetooth. The standard specifies the physical (PHY) layer and medium access control (MAC) layer for the use of LR-WPANs (IEEE 2003). The first version of IEEE 802.15.4 was published in 2003. Unless we state otherwise, the IEEE 802.15.4 standard described in this chapter is this version.

The IEEE 802.15.4 standard defines the specification of the physical and MAC layers. A comprehensive network layer definition is not directly provided by this standard; instead the standard defines the simplest network topologies—star topology and peer-to-peer topology, which could form the infrastructure for networks based on this standard. Figure 2.3 shows the architecture of the IEEE 802.15.4 standard.

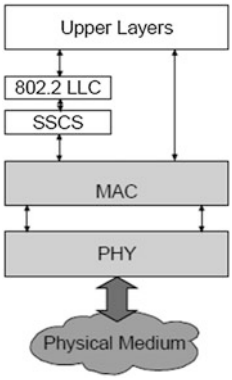


Fig. 2.3 Device architecture defined in IEEE 802.15.4 (IEEE 2003)

In Fig. 2.3, the architecture consists of a two-layer definition, the PHY and MAC layers. The PHY layer mainly includes the radio transceiver and the corresponding low-level control mechanism. The MAC layer provides the definitions for the data transfer by accessing the PHY layer. The service specific convergence (SSCS) and IEEE 802.2^{TM} Type 1 logical link control (LLC) defines a standard mechanism for the upper layers to access the service of the PHY and MAC layers. Because of the characteristic of limited resource, the wireless sensor network applications normally require the used protocol to be as simple as possible, which can reduce the system overhead. The IEEE 802.15.4 architecture is simple and allows the developers to design the application software at a low-level, which can directly interact with the data transfer. More traditional standards, which comply with the standard Open System Interconnection Reference Model (OSI), might be able to provide reliable and abundant service, but the model's 7-layer definition makes that kind of architecture too complicated to be applicable for WSNs' development.

2.2.3 Full Function Device and Reduced Function Device

According to the IEEE 802.15.4 standard, there are two types of devices participating in IEEE 802.15.4 system, a full-function device (FFD) and a reduced-function device (RFD). An FFD is given the capability to implement a full-function IEEE 802.15.4 stack, which makes it be able to become a personal area network (PAN) coordinator (which can initiate and manage the whole network. This includes the establishment of the network, and the acceptance of association requests from other devices, etc.). Alternatively, it can become a coordinator (which has the same functionality as the PAN coordinator, except for initiating a network), or a normal device. An RFD is a device, which can implement the basic functions of the stack, i.e. a minimal implementation of the IEEE 802.15.4 protocol. An RFD cannot be used to initiate and manage a network, but can be used to

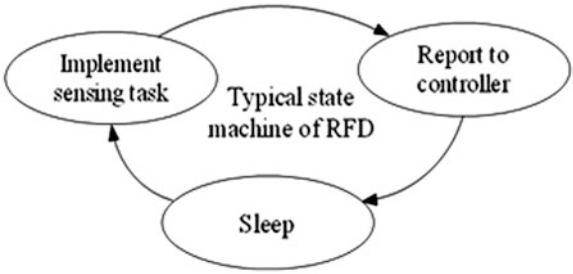


Fig. 2.4 Typical state machine of a RFD

execute extremely simple tasks. The common usage of the RFD is to connect to sensors and regularly send the sensor readings to the network. It is defined in the IEEE 802.15.4 standard that a FFD can talk to other FFDs and RFDs. Using this feature; the upper-layer can implement routing protocols to construct a multi-hop network. However, an RFD can only talk to a FFD since the lack of network management capability makes the RFDs unsuitable for participating in complicated network activities such as sending out beacon signals synchronizing network devices. Consequently, a RFD can last longer than a FFD under the same environment condition. Some wireless sensor network applications are for long term and independent monitoring, consequently, frequently changing the power supply for the distributed sensor nodes is not realistic. In order to save energy, RFDs are more suitable for implementing the functions of such sensor nodes.

Application code running on FFDs can run more complicated applications than those running on RFDs, e.g. application such as network formation, network maintenance, packet relay, network device management. Application code running on RFDs should be kept as simple as possible. Figure 2.4 illustrates a state machine model of a typical RFD. This RFD regularly implements a sensing task, reports the sensor reading to a controller, and then goes to sleep for a certain period before waking up for the next round of sensing.

2.2.4 IEEE 802.15.4 Topologies

IEEE 802.15.4 supports star, tree, cluster tree, and mesh networks. Figure 2.5 depicts the star and peer-to-peer topologies of IEEE 802.15.4. The star topology is used to form star and tree networks, and the peer-to-peer topology to form cluster tree and mesh networks.

In the star topology, a FFD serving as a coordinator is specified to be the central device, which is called the PAN coordinator, and starts and manage the whole network. Other coordinators and network devices must join the network by associating themselves with the PAN coordinator. The PAN coordinator controls all network communications. The peer-to-peer topology also requires a PAN

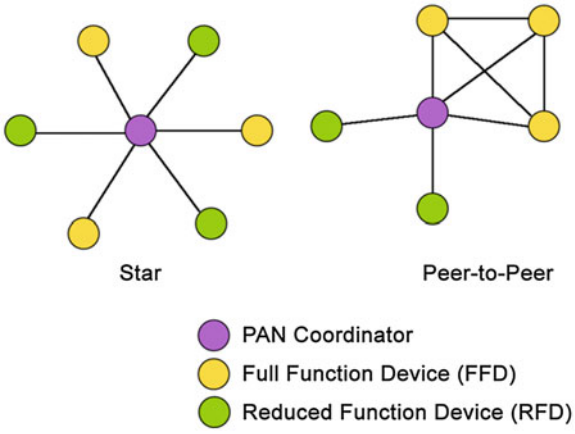


Fig. 2.5 IEEE 802.15.4 topologies

coordinator to initialize the network start-up procedure. However, the communications within a network are based on the peer-to-peer topology and are not limited by the PAN coordinator. Any FFD device can freely talk to any other FFD device so long as they are within effective communication range. Any RFD device can talk only to it's parent FFD device and cannot directly talk to any other RFD device. RFD devices and their parent FFD device form a tree topology.

Cluster tree topology can be a single cluster network or a multi-cluster network. A single cluster network contains only one cluster-head (CH). All the nodes are connected to the cluster-head with one hop, and the network topology becomes a star topology. A multi-cluster network contains more than one cluster-heads. Each node in a cluster can only talk to its cluster-head. All the cluster-heads form an upper level sub-network, which can directly talk to their head, which might be a sink node, connected to an external network or the head of the cluster heads. Nodes in different clusters do not directly talk to each other but communicate among themselves via their cluster heads. Figure 2.6 illustrates the cluster tree topology, which has hierarchy architecture with the clusters at the bottom level and the cluster-head network at the upper level.

A more complex cluster tree topology is shown in Fig. 2.7 where each cluster illustrated by a dotted cycle connects with another cluster via a border node. The border nodes can be a cluster-head or an ordinary node. A designated device (DD) is required to connect with the network via a border node. The DD device with its border node forms cluster 0 with cluster-head CH0. There are four other clusters with cluster-heads CH1 to CH4 in Fig. 2.7. Cluster-head CH1 is serving as a border node for clusters 0 1, cluster-head CH3 as a cluster-head for clusters 1 and 3. Both CH1 and CH3 have two logical addresses, one as a cluster-head and another one as a border node. The cluster tree topology shown in Fig. 2.6 is different from the one shown in Fig. 2.7, which is a flat network.

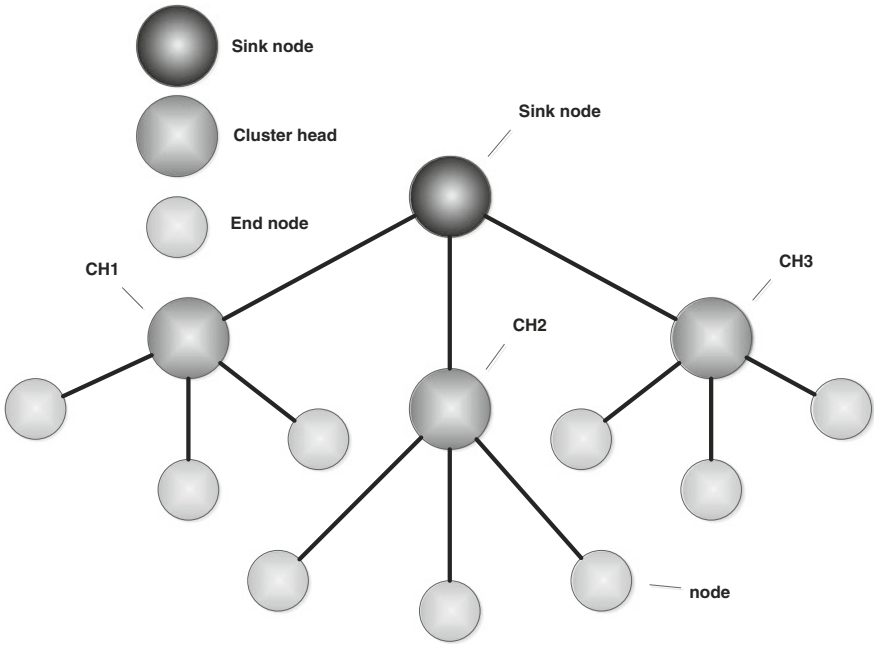


Fig. 2.6 Cluster tree topology

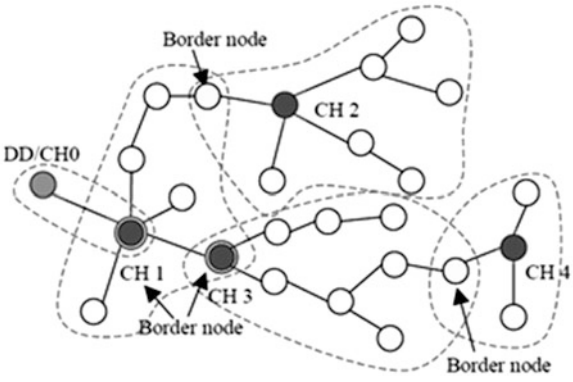


Fig. 2.7 Multi-cluster network connected via border nodes (IEEE 2003)

2.2.5 Multiple Access in IEEE 802.15.4 Wireless Systems

As in all kinds of networks, the wireless nodes in wireless systems have to share a common medium for signal transmission. Multiple Access Control (MAC) protocols in the IEEE 802.15.4 standard defines the manner in which the wireless medium is shared by the participating nodes. This is done in a way that maximizes overall system performance. MAC protocols for wireless networks can be roughly

divided into three categories: fixed assignment (TDMA an FDMA), random access assignment (CSMA/CA), and demand assignment protocols (e.g. polling). In this section, only the most basic concepts of multiple access for wireless networks are presented.

2.2.5.1 Frequency-Hopping/Direct-Sequence Spread Spectrum

Frequecy-hopping spread spectrum (FHSS) divides the scientific band in the ISM band into 79 channels of 1 MHz each. The transmitter divides the information and sends each part to a different channel. The process is known as frequency hopping. The order of the channels or hop sequence used by the transmitters is predefined and has already been communcated to the receiver. Bluetooth uses FHSS for its transmission.

Direct-sequence spread spectrum (DSSS) divides each bit into a pattern of bits called a chip. The chip is generated by performing an XOR (exclusive-OR) operation on each bit with a pseudo random code. The output of the XOR operation, i.e. the chip, is then transmitted. The receiver uses the same pseudo random code to decode the original data.

2.2.5.2 FDMA, TDMA, and CDMA

Frequency division multiple access (FDMA) divides the available spectrum into subbands (i.e. channels) each of which is used by one or more users. Using FDMA, each user is allocated a dedicated channel, different in frequency from the channels allcoated to other users. The user exchange information using the dedicated channel. The largest problem with FDMA is the fact that the channels cannot be very close to one another. A separation in frequency is required, in order to avoid inter-channel interference, as transmitters that transmit on a channel's main frequency band also output some energy on sidebands of the channel.

Time division multiple acces (TDMA) allow users to share the available bandwidth in the time domain, rather than in the frequency domain. TDMA divides a band into several time slots and each active node is assigned one or more time slots for the transmission of its data.

Code division multiple access (CDMA) follows a different approach. Instead of sharing the available bandwidth either in frequency or time, it places all nodes in the same bandwidth at the same time. The transmission of various users are separated by a unique code that has been assigned to each user. CDMA is often referred to as direct-sequence spread spectrum (DSSS). CDMA can be understood by considering the example of various conversations using different languages taking place in the same room. In such as case, people that understand a certain language listen to that conversation and reject everything else in the other language (Nicopolitidis et al. 2003, p. 59.)

2.2.5.3 CSMA/CA

Carrier-sense multiple access with collision avoidance (CSMA/CA) protocols are the basis of the IEEE 802.11MAC layer. A CSAM node that has a packet to transmit listens to see if another transmission is in progress. If this is true, the node waits for the current transmission to complete and then continues to wait for a span of time known as the short interframe space. Then, if there is still no traffic on the medium, the node will start transmission; otherwise, it has to wait again for the medium to become clear.

2.3 Constructing WSNs with IEEE 802.15.4

Figure 2.8 illustrates the general procedure in terms of which a wireless sensor network is established. The procedure starts with a radio channel assessment, then the network initialization, the network establishment announcement, then several further actions, which take place in parallel. This section introduces the procedures of setting up a wireless sensor network with the corresponding concepts defined in the IEEE 802.15.4 standard. The procedure is shown in Fig. 2.8.

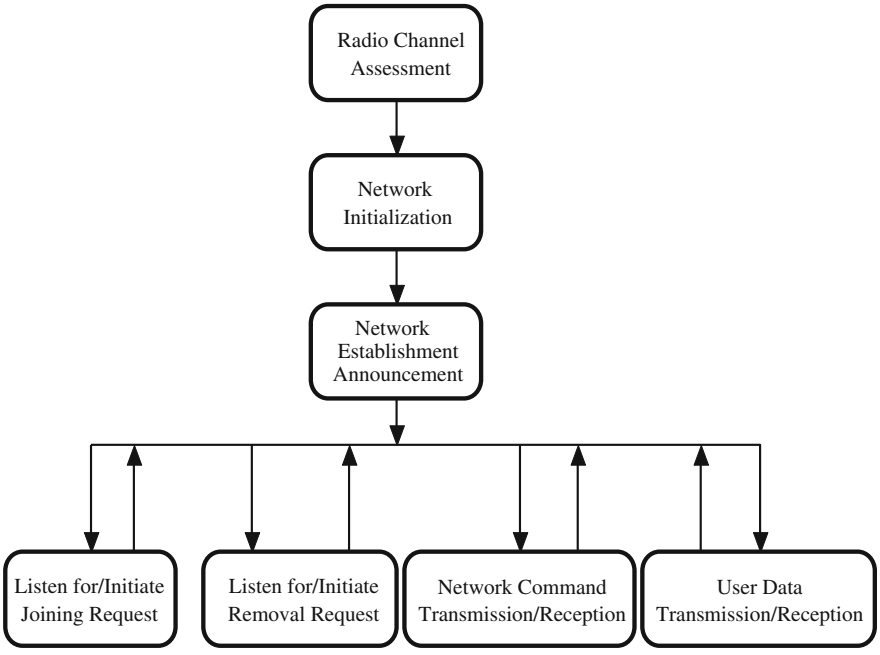


Fig. 2.8 Procedure of establishing a wireless sensor network

2.3.1 Radio Channel Assessment

The first essential task for the construction of a wireless system is always assessing that the desired transmission medium is available. The details of this assessment depend on the characteristics of the wireless network that is to be designed. For networks that utilize frequency hopping, the assessment might focus on the analysis of all available channels and then working out the scheme for hopping. The assessment carried out for networks that utilize frequency division multiple access focuses on searching for the most suitable channel for the network use, such as the cleanest, that which causes the least radio activities, etc. Another important issue in the channel assessment stage is to address how many other systems using the same wireless frequency bands exist in the vicinity. As wireless sensor networks are simple and easy to deploy, multiple networks are highly likely be operating close by. Trying to avoid conflict with other networks is quite crucial during the assessment stage. Chapter 7 in this book will cover the detail of interference avoidance.

The IEEE 802.15.4 standard specifies three functions related to channel assessment: energy detection, active scan, and passive scan. These terms are explained below:

Energy Detection: Energy detection is clearly defined to give the system the ability for determining the energy level on the specified channels.

Any wireless signal activity in the chosen channel increases its energy level. Consequently, using energy detection can locate any potential interfering sources.

Energy detection is the most effective method to assess the channel, particularly, if the unwanted wireless signals do not have the same characteristics of modulation and spreading as the IEEE 802.15.4 transceiver.

Active Scan and Passive Scan: The functions of active and passive scanning are designed to help the system detect how many similar wireless networks exist in the vicinity. Before a FFD coordinator starts an IEEE 802.15.4 network, it should implement at least one active scan. This function is implemented by sending out a beacon (a kind of synchronization signal used to synchronize the network device, normally generated by a network's PAN coordinator) request within the FFD's personal operating space (POS). Then the FFD coordinator will record the received responses, or named beacon frame, containing the network description from any other existing coordinators, as shown in Fig. 2.9. By comparing the output with the received descriptions, the current FFD coordinator is able to determine if it is possible to start the desired network in this area or on the specified channel.

A passive scan implementation is to enable the current FFD's receiver to listen for network beacons on the selected channel over a certain period, as shown in Fig. 2.9. If other coordinators transmit beacons containing their networks' description, the beacons will be recorded and processed using the same method as the active scan.

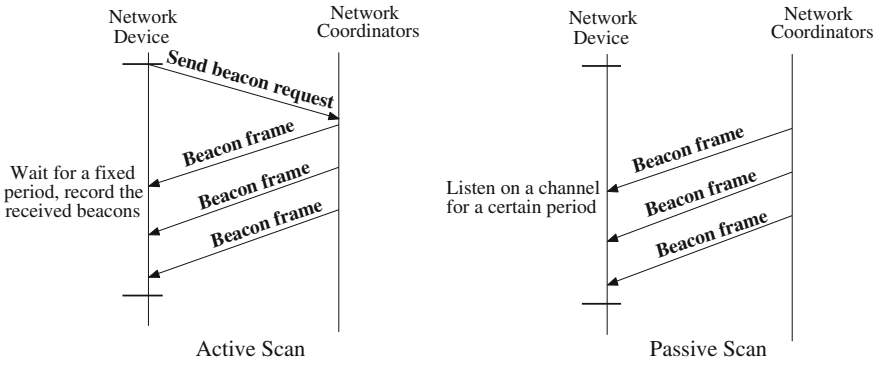


Fig. 2.9 Active/passive scan

The functions of energy detection and active scan are available only for FFD devices, while the passive scan can be applied to both FFD and RFD devices. A typical channel assessment procedure is illustrated in Fig. 2.10, in which energy detection, active scan and passive scan are integrated in a 16 channels assessment.

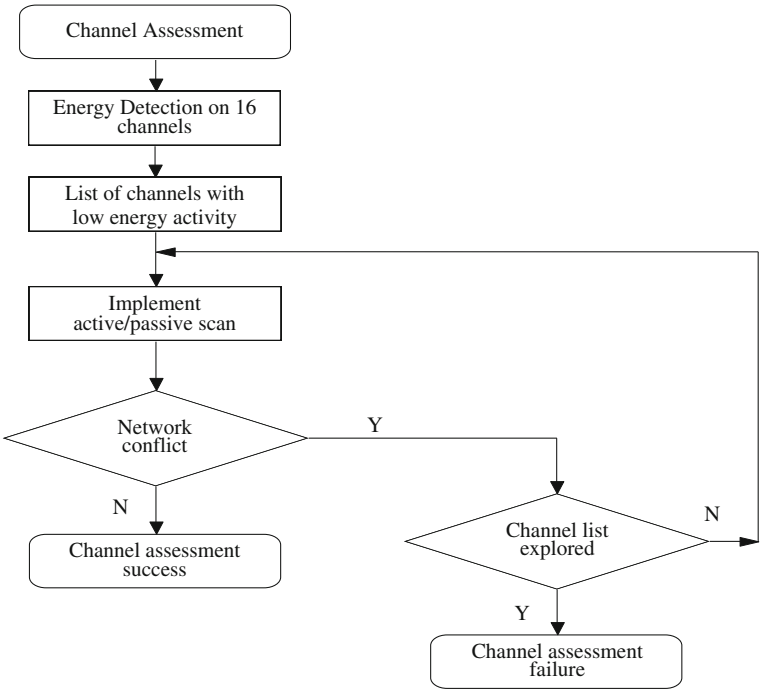


Fig. 2.10 Channel assessment procedure

2.3.2 Network Initialization

Network initialization is implemented by the PAN coordinator. The content of network initialization is to specify various network parameters before actually starting a network. The parameters include the working channel, the network identifier, the network address allocation and setting an IEEE 802.15.4 network beacon.

2.3.2.1 Network Parameter Setting

The working channel is specified according to the results of a channel assessment discussed previously. The IEEE 802.15.4 standard defines the use of the radio frequency and corresponding modulation schemes. The supported data rate is also specified according to the frequency and modulation usage. There are a total of 27 channels across the three frequency bands, which are defined in the standard. Table 2.1 summarizes the allocation of the frequency bands.

Because the IEEE 802.15.4 standard does not support dynamic data rate changing or frequency hopping, a plan for frequency use must be made in advance. Another issue at this stage is the frequency band selection. It needs to comply with the radio regulations, local to where the system is to be deployed.

Once the working channel is decided, the system should select a network identifier by which other devices can identify the network. As a network system, the IEEE 802.15.4 standard supports a 16-bit length network identifier (PAN ID) for labeling each network. The selected PAN ID must be unique and hence cannot be the same as any other network within the radio sphere of influence. Consequently, the active or passive scan can provide useful information for the specified network.

The IEEE 802.15.4 standard defines two basic communication address modes, extended address mode and short address mode. The extended address mode specifies the use of a 64-bit length number, which is fixed in the device's firmware when it was manufactured. The 64-bit address can ensure the device's uniqueness. The disadvantage is that the use of the extended address mode will reduce the effective payload size of any data packet. The short address mode specifies the use of a 16-bit length number. The generation of the 16-bit network address is the responsibility of the PAN coordinator when it starts the network. For example, a PAN coordinator can set its own network address as 0x0000. Then any devices joining the network subsequently can be allocated a 16-bit network addresses by

Table 2.1 Allocation of frequency band and data rate

Frequency band (MHz)	Channel	Bit rate (kb/s)	Modulation
868–868.6	0	20	BPSK
902–928	1–10	40	BPSK
2,400–2,483.5	11–26	250	O-QPSK

adding 1 to the PAN coordinator's address, 0x0001, 0x0002, etc. The length of short address mode decides the theoretical network capacity which cannot exceed 65,535 (i.e. 2^{16}). The use of the short address mode in an IEEE 802.15.4 network can increase the effective payload size in a data packet, but it must be correlated with the PAN ID. Otherwise, the short address's uniqueness cannot be ensured. The standard has no default short address allocation scheme, and network developers can design an appropriate scheme based on the applications requirement.

2.3.2.2 Superframe Structure

The feature of low power consumption in the IEEE 802.15.4 standard is achieved by a low duty-cycle setting. The component which consumes the most power in a wireless system is the transceiver. A typical working current for an IEEE 802.15.4 transceiver is about 20–30 mA. This is significant energy consumption if the transceiver is kept on for all the time, particularly when the module is powered by battery. The IEEE 802.15.4 standard defines the concept of "Superframe Structure" to allow the system to reduce the transceiver usage, while enabling the network to still function.

The superframe structure is a certain period bounded by the network beacons. Upon receipt of the beacons, the network devices' transceivers are synchronously functional and start to execute the designed tasks within the range of the superframe. The superframe structure specifies the period within which the transceivers can be active. If an active period is finished, the transceivers should stop working and remain quiet for the following inactive period until the arrival of the next beacon. The mechanism for synchronization means the system has a chance to save energy without losing communication. To ensure the devices synchronize with the same source, the network beacons are sent from the PAN coordinator, which is required to be power on for the whole network lifetime. Figure 2.11 illustrates the superframe structure, where the abbreviations have the meanings shown in Table 2.2.

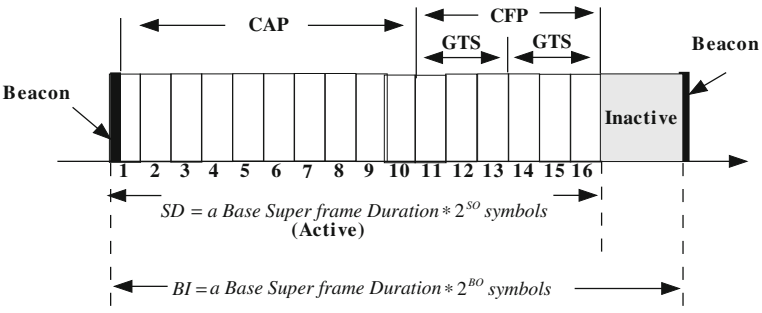


Fig. 2.11 Superframe structure

Table 2.2 Abbreviations in superframe structure

Abbreviation	Meaning
CAP	Contention access period
CFP	Contention free period
GTS	Guaranteed time slot
SD	Superframe duration
SO	Superframe order
BI	Beacon interval
BO	Beacon order

In Fig. 2.11, the superframe structure consists of two main portions: an active period and an inactive period. The length of the active period is denoted as the superframe duration (SD) and calculated by the equation $SD = aBaseSuperframeDuration * 2^{SO}$ where the range of SuperframeOrder (SO) is from 0 to 15, and aBaseSuperframeDuration is calculated as the product of the number of slots (16 in the most case) and the base slot duration (60 in the most case). The whole duration of the superframe structure is called the beacon interval (BI), which includes both the active and the inactive portion and is calculated by the above equation. The range of the beacon order (BO) is from 0 to 15 and the values of SO and BO are related as follows: $0 \leq SO \leq BO \leq 14$. This is because if $BO = SO = 15$, the value of SO should be ignored and the superframe will not exist, consequently the transceivers will be in a state of continuous working with no energy saving.

If $0 \leq SO = BO \leq 14$, the inactive portion will not exist as the length of the beacon interval is equal to the active portion. If $0 \leq SO < BO \leq 14$, the difference between the superframe duration and the beacon interval is the inactive portion, in which all network communications remains silent until the arrival of the next beacon frame.

On receipt of the beacons, network devices can start to implement the designed communications, which must stop before the end of the active portion if $SO < BO$ or before the end of the superframe period if $SO = BO$.

The active portion of the superframe structure is divided equally into 16 slots with the whole duration being $aBaseSuperframeDuration * 2^{SO}$, and further subdivided into two parts: the contention access period (CAP) and the contention free period (CFP). During the CAP, each network device can commence a network communications if required. However, during the CFP, only the devices that have been registered can commence communications. Registrations should be submitted previously to the PAN coordinator by the appropriate network devices. The processed registration information will be contained in the beacon signals. By examining the received beacons, all devices should be able to know if they are registered and hence are allowed to carry out communications in the CFP. Use of the CFP can be allocated to a number of devices, and the communication duration permitted for each registered device is controlled by the guaranteed time slot

(GTS). Further detail can be obtained by reference to the IEEE 802.15.4 standard (2003).

Once the beacon order BO and the superframe order (SO) are chosen, the duty cycle can be calculated. For example, on one of the 16 channels on the 2.4 GHz band, if the beacon order and superframe order are set at 3 and 2 respectively, the beacon interval and superframe duration can be calculated as follows:

$$\begin{aligned} BI &= aBaseSuperframeDuration * 2^{BO} symbols \\ &= numberOfSlots * baseSlotDuration * 2^{BO} symbols \\ &= 16 \times 60 \times 2^3 \times 16 = 122.88 \text{ ms} \end{aligned} \tag{2.1}$$

$$SD = aBaseSuperframeDuration * 2^{SO} * symbols = 960 \times 2^2 \times 16 = 61.44 \text{ ms} \tag{2.2}$$

The PAN coordinator will generate about eight beacons each second $(1{,}000/122.88 \approx 8)$. During each beacon interval, the transceivers of the network devices work for about 61.44 ms and keep quiet for the rest of time. Therefore, the duty cycle is about 50 % $(61.44/122.88 = 0.5)$, which briefly means about 50 % of the energy consumption is possibly saved.

Making the transceivers work in the "on or off" mode can save energy, but might cause two problems: firstly, the system may be not able to finish a complete transmission and reception in the time available or secondly, the system response may be delayed. In the first case, it is necessary to calculate the time required for transmitting a single data packet. A full size IEEE 802.15.4 data packet is 133 bytes (IEEE 2003). Using the given data rate, for example 250 kbps at 2.4 GHz, the time required to send an IEEE 802.15.4 data packet is up to 4.256 ms (i.e. $(133 \times 8)/(250 \times 10^3)$. The active portion section in the superframe shown in Fig. 2.11 should be long enough to handle such a transmission within a beacon interval.

Concerning the response delay caused by the mode "on and off" in the data transmission, a proper duty-cycle, i.e. a beacon interval and a superframe interval should be set since a low duty-cycle setting will slow the system response. Table 2.3 summarizes the beacon order and superframe order setting when the duty-cycle is set at 50 % with exception of $BO = SO = 0$.

In Table 2.3, the setting of the superframe order is less than beacon order by 1 (except $BO = SO = 0$). Therefore, the duty-cycle is fixed at 50 %. As defined by the standard, the network will remain quiet during the duration of the inactive period when there is no radio communications allowed. The beacon orders from 1 to 6 in Table 2.3 have the beacon interval set to less than 1 s, which is acceptable for most applications. Beacon orders from 7 to 10 have considerable delay, because the inactive period has duration of about a second (from 983.04 to 7864.32 ms). Increasing the superframe order can reduce the system response delay and decreasing the beacon order, i.e. increasing the duty-cycle. However, a high duty-cycle will consume more power, which is less energy efficient.

Table 2.3 Summary of beacon order and superframe order setting at 50 % duty-cycle

Beacon order (BO)	Superframe order (SO)	Beacon interval (ms)	Superframe interval (ms)	Inactive period (ms)
0	0	15.36	15.36	0
1	0	30.72	15.36	15.36
2	1	61.44	30.72	30.72
3	2	122.88	61.44	61.44
4	3	245.76	122.88	122.88
5	4	491.52	245.76	245.76
6	5	983.04	491.52	491.52
7	6	1,966.08	983.04	983.04
8	7	3,932.16	1,966.08	1,966.08
9	8	7,864.32	3,932.16	3,932.16
10	9	15,728.64	7,864.32	7,864.32

Achieving the balance between the system performance and power consumption is already a challenge to any power supply limited application, and application specific solutions can be achieved. If both BO and SO are set as 15, there will be no power saving issues, as the superframe structure does not exist.

2.3.3 Network Establishment Announcement

Once the network parameters have been initialized, the PAN coordinator can announce the successful establishment of the network. The actual procedure for announcing the establishment of the network is determined by the network protocols used. The purpose of the announcement is to indicate to other devices the existence of the current wireless system. There are two ways to achieve this purpose: announce actively or respond passively upon receiving a requested. Some wireless protocols use the regular beacon signals to synchronize the network operations. This type of network is called a beacon-enabled network. It also informs those newly starting devices with the characteristics of the current wireless systems such as the working channel, frequency band, physical location, etc. If the protocol does not support a regular beacon signal emission, this type of network is called a non-beacon-enabled network, and the PAN coordinator will keep listening on the working channel, and respond to any valid requests which are sent by those devices executing radio channel assessments.

For a beacon-enabled network, after the announcement of the establishment of the network, a beacon signal will be regularly sent out according to the setting of the SO and BO. During the whole working period of the network, the PAN coordinator should ensure the persistent beacon transmission in order to make it detectable by those devices implementing a passive scan, meanwhile any active scan initiated by other devices should also require a response.

2.3.4 Listen for/Initiate Joining Request

After successfully initializing an IEEE 802.15.4 network, the PAN coordinator now becomes the prime network manager. Unless the transceiver of the PAN coordinator is busy on data transmission, it should keep listening on the selected working channel all the time in order to perform the duty of network management.

Any devices wishing to join the network should implement three basic steps: initiate an active scan (FFD only) or a passive scan to locate the desired PAN coordinator, synchronize with the network beacons if applicable $(0 \leq SO \leq BO \leq 14)$, request to join the network by issuing an associate request to the located PAN coordinator. Upon receipt of the joining request, the PAN coordinator can implement the designed procedure to validate the request. If the request is granted, the PAN coordinator can decide how to allocate a network address to the device, and it then sends back a response containing the network information (i.e. the network address) and decision to the device. If the joining request is rejected, the PAN coordinator should send back corresponding feedback. Upon the receipt of the response from the PAN coordinator, the network device can use the allocated address to implement network communication, or call a predefined algorithm to deal with the response of "joining failure".

2.3.5 Listen for/Initiate Removal Request

The way to deal with the removal request is the reverse process to the joining request. The PAN coordinator can delete the device address from the accepted device list and notify the device the removal decision. Alternatively, the PAN coordinator can implement procedures when it receives the disassociate request from the network device. Upon receipt of the notification from the PAN coordinator, the device can ensure that the removal request is permitted.

2.3.6 Network Command Transmission/Reception

The transmission and reception of the network commands are mainly for network management purposes. They are normally invisible to the users without any user interventions. However, sometime a command requiring user interventions will not proceed until the user's instructions are obtained. Therefore, it is necessary to have a processing module for this kind of use in the system design. For example, when a network device notices that there is another IEEE 802.15.4 network in operation in the vicinity and which is using the same network ID, it should send a conflict notification command to the PAN coordinator. Then PAN coordinator should then start an active scan and determine a new PAN ID by broadcasting the coordinator-

realignment command. In this case, the new PAN ID selection is required to be completed by user intervention. Another example is when a network system starts to increase the security level for adopting new devices; the details of any such requesting device need to be reviewed by the upper layer management system. Subsequently network command transmission and reception will occur in this case.

2.3.7 Data Transmission and Reception

The data transmission and reception in the IEEE 802.15.4 standard is categorized according to the use of a beacon. Figure 2.12 illustrates the communication method. There are two communication directions: from a coordinator to a network device, and from a network device to a coordinator.

2.3.7.1 Coordinator → Network Device

In a beacon-enabled network as shown in Fig. 2.12a, when a coordinator has a packet to transmit to a network device, it stores the data in the local buffer, and

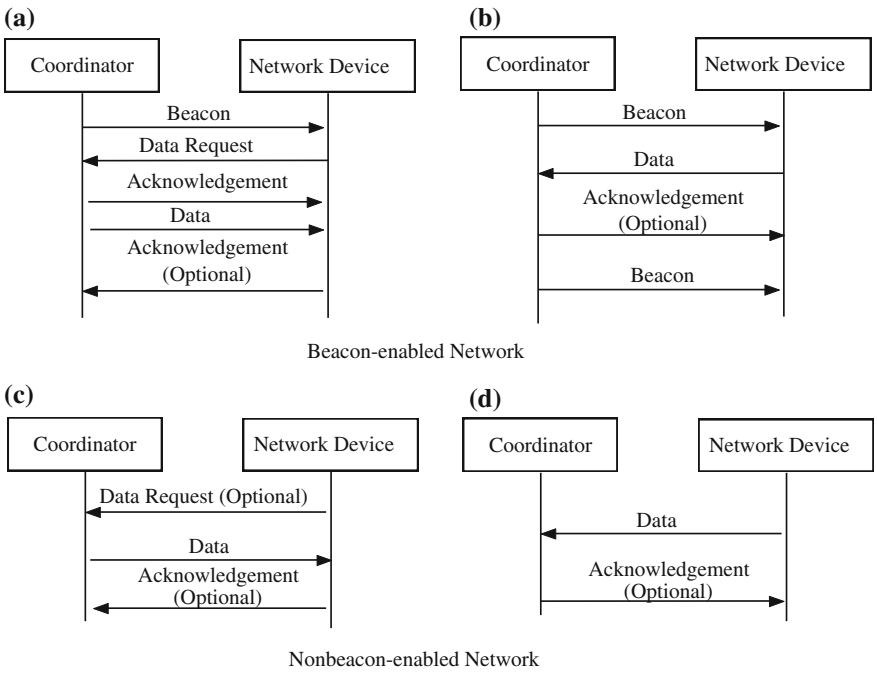


Fig. 2.12 Communication methods defined in IEEE 802.15.4 standard

puts the packet information (i.e. destination address) into the "address pending list" of the beacon frame. On receipt of the beacon frame, the network device will know if there is a packet pending on the coordinator. There are two options for the network device to proceed: if the network device's *macAutoRequest,* an indicator for the MAC response mode, is set as TRUE, it should automatically send the data request command to the coordinator using the slotted CSMA-CA (carrier sense multiple access with collision avoidance) to request the pending data. If the *macAutoRequest* is set as FALSE, the stack should present the application layer with a primitive of "Beacon Notify", and let the application decide if it is necessary to send a data request command. On receipt of the data request command, the coordinator will firstly decide the method to send an acknowledgement to the network device. If the coordinator is able to check the local buffer and determine that the pending packet for that network device exists, it then sends the acknowledgement within *macAckWaitDuration*, a predefined duration time. If it is not able to complete the acknowledgement sending within the required time, the coordinator should send the acknowledgement with the data pending field, an indicator of the pending state, set to 1. After sending out the acknowledgement, the coordinator should send the data packet to the network device if the data pending filed is set to 1 in the previous acknowledgement frame. The length of the data payload will be 0 if there is no data pending. On receipt of the acknowledgement, the network device will enable its receiver for the maximum duration of *aMaxFrameResponseTime,* if the data pending field in the acknowledgement is 1. The network device may be required to send back an acknowledgement to indicate the successful reception. The data frame transmission from the coordinator to the network device should use the mechanism of slotted CSMA-CA.

In a nonbeacon-enabled network (Fig. 2.12c), if a coordinator wishes to send a data packet to a network device, it has two options: sends the data packet to the network device directly using the unslotted CSMA-CA, or stores the data into the local buffer and waits for the data request command from the network device, if the network device is programmed to "poll" the coordinator within a certain interval. The process for polling data from the coordinator in the nonbeacon-enabled network is the same as in a beacon-enabled network. However, the CSMA-CA mechanism should use its unslotted version as there is no superframe structure existing.

2.3.7.2 Network Device → Coordinator

In a beacon-enabled network (Fig. 2.12b), the whole network is in an active period. Then on receipt of the regular beacon, if the network device has a packet to transmit to the coordinator, it can commence the communication using the slotted CSMA-CA. It must ensure that the transmission including the acknowledgement can be finished before the end of the active period. Otherwise, the procedure will be temporarily suspended and resume at the start of the next active period. In a nonbeacon-enabled network (Fig. 2.12d), if a network device has a data packet to transmit to the coordinator, it can simply start the communication with the use of unslotted CSMA-CA.

The method of storing the data on the coordinator and sending it out until receiving a request from the network device is called indirect transmission. The indirect transmission is designed for keeping power consumption low. The reason is that the network devices are normally in a sleep state and the radio receiver is off in order to save energy. Storing data on the coordinator can make it convenient for the network devices to obtain the information without keeping the receiver on all the time. The network device can send the data request command when a time slot is available.

2.3.8 Slotted and Unslotted CSMA-CA

As mentioned in the previous section, both beacon-enabled networks and non-beacon-enabled networks use CSMA-CA in their data transmission. Two versions of CSMA-CA are available: slotted CSMA-CA for beacon-enabled access and unslotted CSMA-CA for non-beacon-enabled access. The concept of "slot" is only available in the beacon-enabled network. Figure 2.13 illustrates the "slot" and corresponding "backoff period" in a superframe structure where $BO = SO = 0$.

In Fig. 2.13, the superframe structure is divided into 16 equal sections, each of which is called a "slot". The basic element used to locate the appropriate time point in each slot is called the "backoff period", which is represented by the *aUnitBackoffPeriod* symbol, and is the time that a device must wait before accessing the network again. Because the slot number in a superframe is fixed at 16, according to Eq. (2.2) the duration of a single slot is obtained as:

$$\begin{aligned} T_{SuperframeSlot} &= \frac{SD}{16} \\ &= \frac{60 * 16 * 2^{SO} symbols}{16} = 60 * 2^{SO} symbols \end{aligned} \tag{2.3}$$

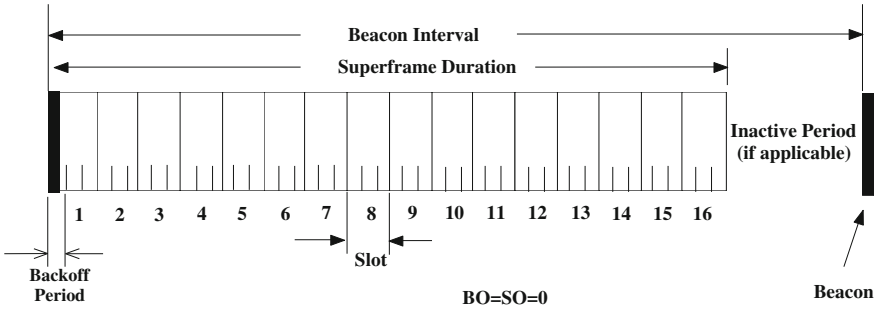


Fig. 2.13 Slot and backoff period in the superframe structure with $BO = SO = 0$

The duration of a single backoff period is defined as: $T_{Backoff\_Period} = aUnitBackoffPeriod = 20\ symbols$

Then the number of backoff period in a single slot $N_{Backoff\_Period}$ is defined as:

$$N_{Backoff\_Period} = \frac{T_{SuperframeSlot}}{T_{Backoff\_Period}} = \frac{60 * 2^{SO} symbols}{20\ symbols} = 3 * 2^{SO} \quad (2.4)$$

In Fig. 2.13, the number of backoff period in each slot is 3, where $SO = 0$.

There are a number of terms used in the operation of CSMA-CA summarized in Table 2.4. Figure 2.14 shows the flowchart of the slotted and unslotted CSMA-CA operation.

In Fig. 2.14, to perform CSMA-CA, the system must first check whether the current network is beacon-enabled. If it is, the slotted CSMA-CA shown in the left hand side is used. Otherwise, the unslotted CSMA-CA shown in the right hand side is used.

In the slotted CSMA-CA, three parameters should be initialized before proceeding, *NB*, *CW* and *BE*. *NB* is set to an initial value is 0. *CW's* initial value is set as 2, and will be reset to 2 each time the channel is assessed to be busy. *BE* is the backoff exponent, which defines how many backoff periods a device should implement before attempting to assess the channel. If the parameter of *macBattLifeExt* is set as FALSE, *BE* should be equal to the value of *macMinBE*. Otherwise, *BE* should be initialized to the lesser of 2 and the value of *macMinBE*. After completing the parameter initialization, the system should locate the boundary of the next available backoff period (*point* 1). Then it delays for a number of backoff periods, where the number is randomly selected between 0 and

Table 2.4 Terms used in CSMA-CA operations

Term	Meaning
Unit of backoff period	The time that a device must wait before accessing the network again
Backoff exponent (BE)	The time a device must wait before it may attempt to retransmit after previously attempting to transmit a packet over a busy channel
macMaxBE	Maximum backoff exponent, always 5
macMinBE	Minimum backoff exponent, always 3
Number of backoffs (NB)	The number of times a device has tried to access the network via CSMA-CA, the initial value is 0
Contention window (CW)	The length of CW is defined as the number of unit of backoff periods that a channel must be clear before transmission can proceed, the initial value is 2
MacBattery life extension (*macBattLIfeExt*)	When this value is set to true, a device will use the MAC battery life extension period to calculate the number of backoff periods
MacMaxCSMABackoffs	The maximum number of times a device can perform CSMA-CA to access a network, the default value is 4
Backoff period boundary	The start of the beacon in the superframe

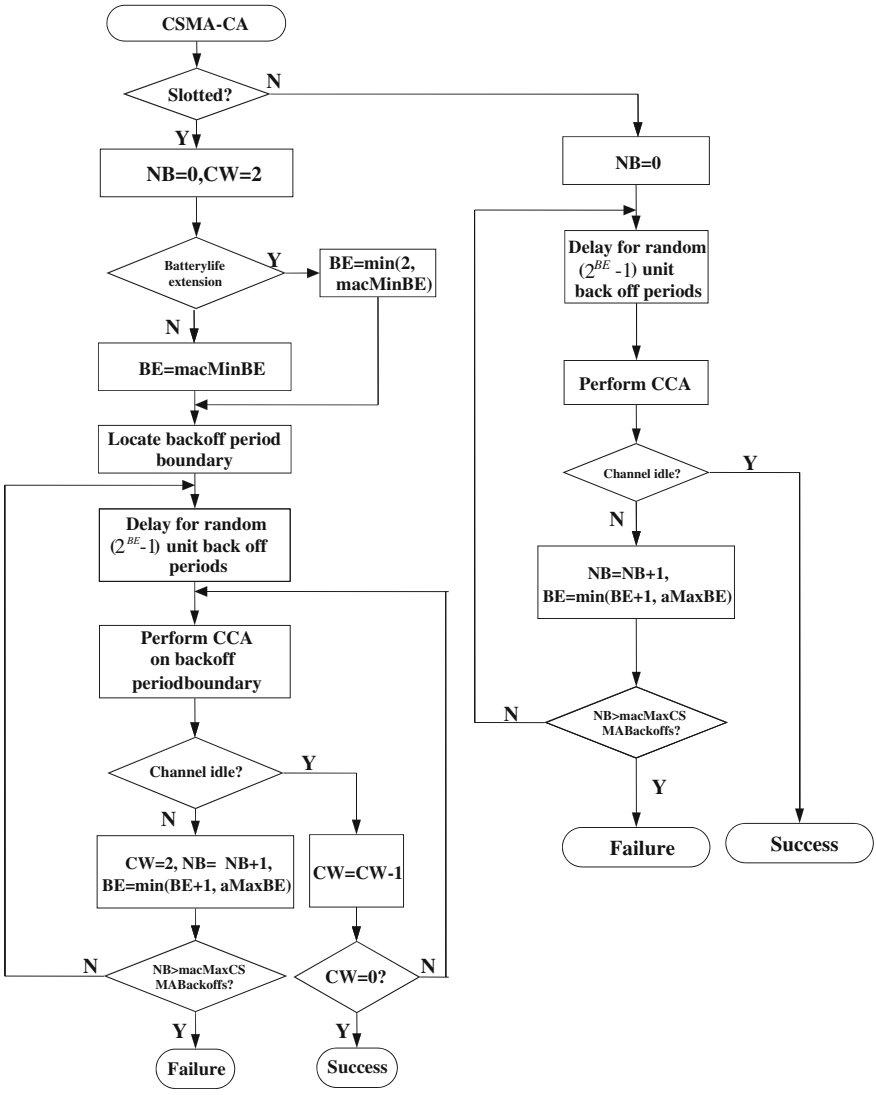


Fig. 2.14 Flow chart of CSMA-CA in IEEE 802.15.4 standard

$2^{BE} - 1$ (*point* 2). When the delay is finished, the system should perform the CCA on the boundary of the next available backoff period. An important rule defined by the IEEE 802.15.4 standard is that before the first time random delay of the current attempt the MAC layer should evaluate whether the delay can finish before the end of the CAP. If it cannot finish, the system should pause the counter at the end of the CAP, and resume it at the start of the next superframe. If it can finish, the system should apply the backoff delay. When the backoff delay finishes, the

system should evaluate again to determine if the rest of the operations, including two CCA analyses, data frame transmission and the possible acknowledgement reception can be completed before the end of the CAP. If the MAC layer can handle these, the system should start to perform CCA now (*point* 3). If not, the system should stop and wait for the next superframe and repeat the evaluation.

If the channel is assessed to be busy, *CW* is reset to be 2. The value of NB is increased by 1. *BE* is reselected from the lesser of $BE + 1$ and *macMaxBE*. If the value of *NB* is greater than *macMaxCSMABackoffs*, the current attempt is announced to have failed. If not, it should go to *point* 2. If the channel is assessed to be idle, the system should check *CW* by subtracting 1. If *CW* is not equal to 0, the system should go to *point* 3. Otherwise, the MAC layer can commence the data transmission on the boundary of the next available backoff period.

In the unslotted system (i.e. nonbeacon-enabled network), two parameters are required to be initialized, *NB* and *BE*. *NB* is set to be 0, and *BE* is set to be *macMinBE*. After initialization, the system should delay a number of backoff period where the number is randomly selected between 0 and $2^{BE} - 1$ (*point* 4). After the delay, the MAC layer can perform CCA (*point* 5). If the channel is assessed to be busy, increase *NB* by 1 and reselect *BE* from the lesser of $BE + 1$ and *macMaxBE*. If *NB* is greater than*macMaxCSMABackoffs*, the current attempt is announced to have failed. If not, it should go to point 4. If the channel is assessed to be idle, the MAC layer can immediately commence the data transmission.

2.3.9 Summary of Data Transmission in IEEE 802.15.4

Because most of the functions used in the data transmission are encapsulated into the stack, the developers may not have been able to access the actual implementation of the mechanisms defined or described in the standard. In spite of that, it is still necessary to know what the system does. Particularly since a lot of stacks will handle the encountered problems and return them to the applications to ask for the users' manual processing. For example, due to the hardware capacity, the coordinator cannot store the pending data permanently in the local buffer. After a certain period, the stack will return an event of "TRANSACTION_EXIRED" to the application. Or when the coordinator wants to store a new pending data, the stack will also return an event of "TRANSACTION_OVERFLOW" if there is no space available. For the CSMA-CA implementation, the failure of CCA, successful transmission, and missing acknowledgement are all designed to return a corresponding event to the application. Properly handling the event returning from the stack in the function block of "Network Command Transmission/Reception" is essential for the embedded software design.

2.4 ZigBee and Wireless Sensor Networks

2.4.1 ZigBee Stack Structure

The IEEE 802.15.4 standard defines a mechanism to achieve the wireless communication featured with low data rate and low power consumption. It only supports the star and peer-to-peer topologies. There is no definition for a comprehensive network system. Technically, as the IEEE 802.15.4 standard focuses on the development of the PHY and MAC layers, it is mainly suitable for wireless communication, rather than for use in large scale network applications. The ZigBee specification was created in 2004 by the ZigBee Alliance in order to establish large-scale wireless networks on top of the IEEE 802.15.4 standard, which only defines the PHY and MAC layers, for low-rate wireless personal area network (LR-WPAN). The name ZigBee comes from the honeybee, which uses a zigzag type of dance to communicate with other members. ZigBee developers want to emulate this action for solving complex communication tasks simply in LR-WPAN. The ZigBee standard offers a stack profile that defines the network (NWK), security, and application layers. Developers are responsible for creating their own application profiles or integrating with the public profiles provided by the ZigBee Alliance. The publically available ZigBee profiles cover smart energy, building automation, home automation, home and hospital care, telecom applications, consumer electronics control, and industrial process monitoring and control (Elahi and Gschwender 2009). The late version of the ZigBee standard was called ZigBee PRO and published in 2007. Figure 2.15 shows how ZigBee stack sits on top of IEEE 802.15.4.

The characteristics of the ZigBee standard focus on low data rate, low cost, low complexity, low power consumption, ease-to-implement. Table 2.5 shows a comparison of ZigBee characteristics with those of WiFi (IEEE 802.11) and

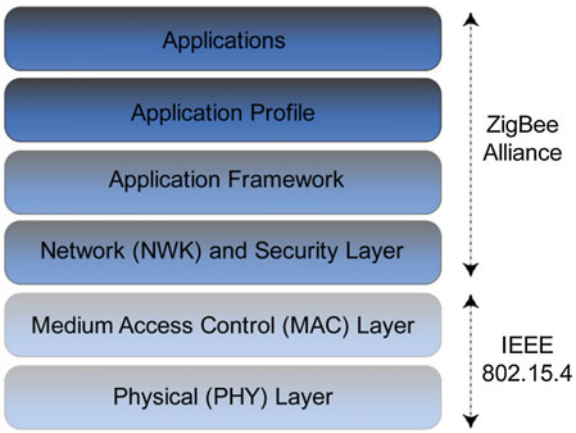


Fig. 2.15 IEEE 802.15.4 and ZigBee stack

Table 2.5 Comparison of ZigBee, WiFi, and bluetooth (Elahi and Gschwender 2009)

	WiFi (IEEE 802.11)	Bluetooth (IEEE 802.15.1)	ZigBee (IEEE 802.15.4)
Application	Wireless LAN	Cable replacement	Control & monitoring
Frequency bands	2.4 GHz	2.4 GHz	2.4 GHz, 868 and 915 MHz
Battery life (days)	0.1–5	1–7	100–700
Node per network	30	7	65,000
Bandwidth	2–100 Mbps	1 Mbps	20–250 kbps
Range (m)	1–100	1–10	1–75
Topology	Tree	Tree	Star, tree, cluster tree, mesh
Standby current (Amps)	20×10^{-3}	200×10^{-6}	3×10^{-6}
Memory (KB)	100	100	32–60

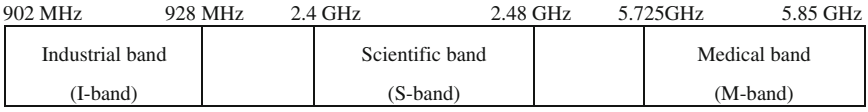


Fig. 2.16 ISM band frequency allocation

Bluetooth (IEEE 802.15.1). Table 2.5 shows that WiFi, Bluetooth, and ZigBee use the ISM (industrial, scientific research, and medical applications) band, which is license free and has a transmission power of less than 1 W. Figure 2.16 illustrates the frequency allocation of the ISM band made by the federal communication committee (FCC). ZigBee can be set at I-band, S-band, or M-band. WiFi and Bluetooth can only be set at S-band.

Figure 2.17 illustrates the architecture of the ZigBee stack, which is divided into three sections, as follows:

- IEEE 802.15.4, which consists of the MAC and PHY layers.
- The ZigBee section, which consists of the network (NWK) layer, the application support sublayer (APS), the security service management, and the ZigBee device object (ZDO). The endpoint is an application object, which can have up to 240 separate application objects. An endpoint defines input and output to the APS. The ZDO performs control and management of application objects.
- The ZigBee application section, in which developers can use the ZigBee application profiles or develop their own application profile.

In the ZigBee specification, the network devices are categorized into three types: ZigBee coordinator, ZigBee router, and ZigBee end device. A ZigBee coordinator is an IEEE 802.15.4 PAN coordinator, which is a fully function device. Any ZigBee network should have one and only one ZigBee coordinator. The ZigBee coordinator should be capable of selecting an available channel and

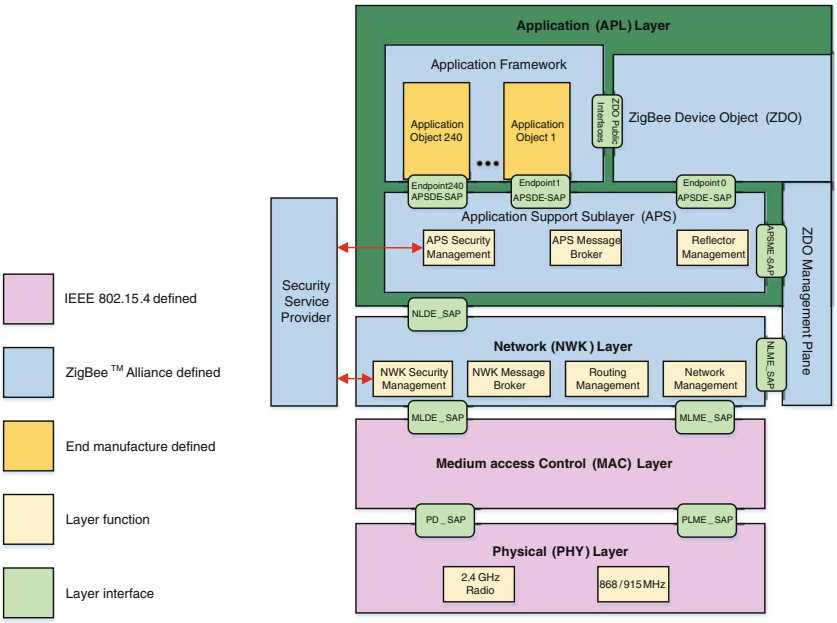


Fig. 2.17 Stack structure of ZigBee (2004)

appropriate network identifier (16-bit length) for the creation of a new network. As the first device that starts the ZigBee network, the ZigBee coordinator takes the responsibility of adopting the new devices and allocating network address.

A ZigBee router is an IEEE 802.15.4 full function device. The ZigBee router should provide the capability to select an existing ZigBee network to join and extend the ZigBee network by adopting new devices, which are out of the communication range of the ZigBee coordinator. Meanwhile, the ZigBee routers construct the backbone of the network by implementing the designed routing protocols.

A ZigBee end device is an IEEE 802.15.4 full function device or reduced function device. As the ZigBee end devices are often deployed at the end of the network to implement the sensing tasks, a reduced function device is more suitable for such use, as they are more energy efficient.

2.4.2 ZigBee Topologies

2.4.2.1 ZigBee Topologies

ZigBee is on top of IEEE 802.15.4 and uses the IEEE 802.15.4 specification for its MAC and PHY layers. ZigBee network supports star, tree and mesh topologies by

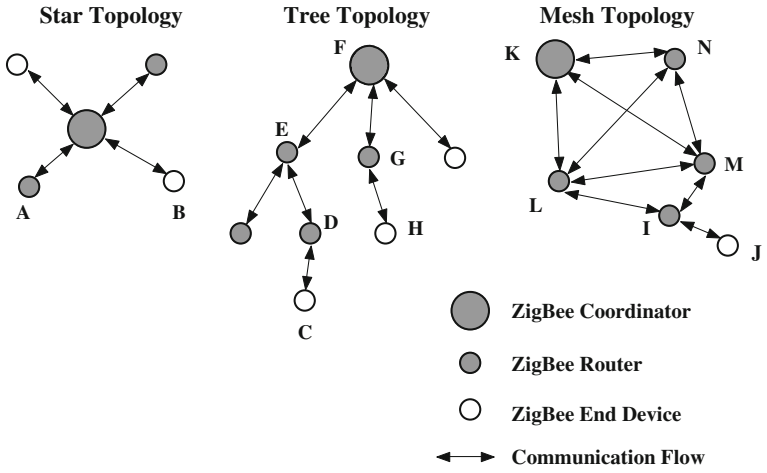


Fig. 2.18 Three topologies in ZigBee specification

extending the use of peer-to-peer topology. Figure 2.18 illustrates the three topologies.

In Fig. 2.18, the star topology is the easiest one to achieve. The ZigBee coordinator is the centre node of the network, other ZigBee devices including the ZigBee routers and ZigBee end devices are required to connect to the ZigBee coordinator to form the network. The star topology is not suitable for large-scale applications due to the limitation of the ZigBee coordinator. Because all the devices must join the network through the ZigBee coordinator, devices, which are out of the radio range of the coordinator, cannot be networked. The main shortcoming of the star topology is that the failure of the centre node (ZigBee coordinator) will affect the whole network. Devices in a star network cannot communicate with each other directly. For example, if device A is to send a message to device B in the start topology shown in Fig. 2.18, the message will be sent to the ZigBee coordinator first, and then relayed to the destination.

The tree topology is more flexible compared with the star topology. Its deployment is not limited by the coordinator, and can be extended by using ZigBee routers to adopt sub-devices. The criteria to form a tree network are: an end device joins the tree via a router device, and a router device joins the tree via another router device (the ZigBee coordinator can be used as a router device as well). The difference is that a router device can adopt end devices or other router devices as its sub-devices, which are also called children. An end device cannot have children. Therefore an end device cannot be a parent device. The network communications in a tree network must comply with these rules. For example, if device C is to send a message to device H, the message should be sent back to device F by passing through devices D and E. Then device F sends the message down to the device H through device G. The criterion is that the message must travel from the

source node up the tree to the nearest common ancestor and then down the tree to the destination node (ZigBee 2004). The disadvantage is that there is no alternative route available if any one of links on the route fails. However, the implementation of the routing protocol is fairly easy as each device just needs to maintain a tree table and simply pass the message to the parent node or to the descendent node which points to the destination.

The mesh topology has the same structure as the tree topology, but its network communications are more flexible. All routers are allowed to communicate with each other without needs to send message to the parent device first. For example, if device J is to send a message to device K, the possible routes can be J → I → L → K, J → I → M → K, J → I → M → N → K, J → I → L → N → K and J → I → L → M → N → K. The network routing algorithm would pick up an alternative route from the available options when some of them fail.

2.4.2.2 ZigBee Hybrid Network

By appropriately using star, tree, and mesh topologies, or a combination of them, ZigBee network can form various architectures. Among them the mesh topology is the most popular network topology with flexible network configuration and the capability of self-healing in network communication.

- *Flexible Network Configuration*

From the view of logic relationship, the mesh topology is the same as the tree topology. Because the mesh topology employs the same criteria as the tree topology to form the network, it can be thought of as a special version of the tree topology. Meanwhile a mesh topology is an amplified star topology. Since the router nodes can commence communications with each other, it is possible to program the communication flows to make them aggregate at a single point, if required by the application. Therefore, the mesh topology is flexible in its network configuration. However, a star or tree network cannot be extended to a mesh network.

- *Capability of self-healing*

As discussed above, the star network and tree network have the same critical defect: if any link on the route or the centre node fails, the whole network will breakdown. Using the dynamic routing protocol, the mesh network can solve the problem by walking around the failed links or nodes.

- *Hybrid structure*

A ZigBee network has normally a hybrid structure rather than one of the three topologies discussed above. It consists of two layers as shown in Fig. 2.19. Layer 1 consists of the ZigBee coordinator and ZigBee router devices. The router devices construct the backbone of the network within which the routing protocols can be implemented. The ZigBee end devices, which are categorized into layer 2, join

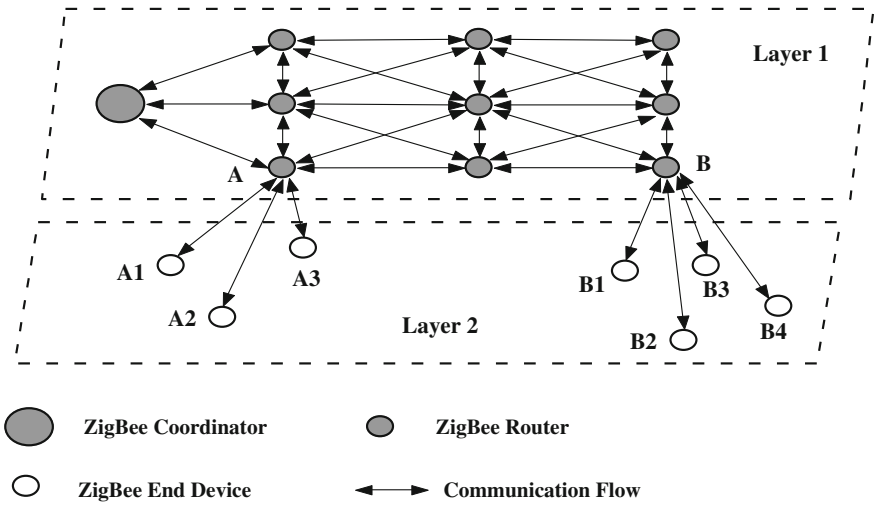


Fig. 2.19 ZigBee hybrid network

the network through their parent ZigBee router devices. According to the ZigBee specification, a ZigBee end device can only talk to its parent device. Therefore, the ZigBee end devices are not involved into the network communication relay. Any network communications between a ZigBee end device and other ZigBee network devices must be sent to the corresponding parent device, and then routed to the destination. The parent ZigBee router device and its connected ZigBee end devices form a star topology.

2.4.3 ZigBee Address Allocation Scheme

The ZigBee specification also defines a practical network address distributed allocation scheme for the network use, named as "Cskip", which is not specified in the IEEE 802.15.4 standard. Briefly, the network capacity i.e. the number of available 16-bit addresses in a tree topology is decided by four parameters:

- C_m—Total number of children that any parent device may have
- R_m—Total number of router children that any parent device may have
- L_m—Maximum depth of the network (i.e. the level at which parent devices may no longer have children)
- d—Actual depth of the device under consideration

The above four parameters are stored in the network information base of the coordinator. The ZigBee coordinator assigns a block of addresses to each router

based on the maximum number of children C_m. The allowed number of end devices accepted by a router device is calculated as:

$$MaxEndDevices = MaxChildren - MaxRouters = C_m - R_m \tag{2.5}$$

In the sequence, when a router device successfully joins a network, its parent device allocates a block of address for its use, which means the joined router device becomes a potential parent device. Each joined router device can accept a certain number of children devices whose number cannot exceed C_m. The joining of the new router device is considered to extend the depth of the network, and the depth should not be greter than L_m.

If an end device successfully joins a network, it will be allocated a network address by its parent device. The joined end device does not have the capability to accept new children devices.

Cskip is the method for calculating the total number of possible descendants that exist down any branch in the network. It is defined as follows:

$$Cskip(d) = \begin{cases} 1 + C_m \times (L_m - d - 1) & if \quad R_m = 1 \\ \frac{1 + C_m - R_m - C_m \times R_m^{L_m - d - 1}}{1 - R_m} & if \quad R_m > 1 \end{cases} \tag{2.6}$$

Figure 2.20 illustrates the address allocation with $C_m = 5$, $R_m = 2$, and $L_m = 2$. According to Eq. (2.6), $Cskip(0)$, $Cskip(1)$, and $Cskip(2)$ are obtained to be 6, 1, and 0 respectively. $Cskip(0)$ is the maximum possible number of descendants that lie down each router-branch from the coordinator. $Cskip(1) = 1$ means that each branch from the router maximally has one descendants. $Cskip(2) = 0$ means that any router at this level has no descendant.

The total number of potential nodes in the network can be calculated as follows:

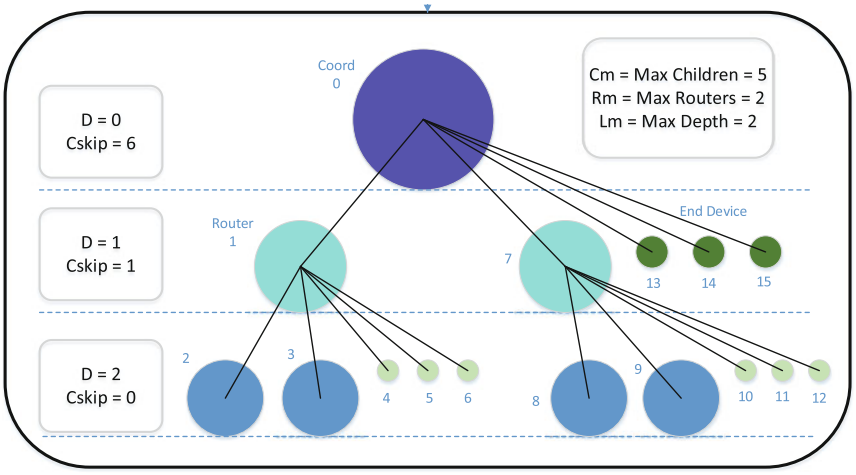


Fig. 2.20 Cskip address allocation scheme (ZigBee 2008)

Table 2.6 Network capacity evaluation under certain parameters (ZigBee 2008)

Increasing value	C_m	R_m	L_m	Total number of nodes
None	20	6	5	31,101
C_m	21	6	5	32,656
R_m	20	7	5	56,021
L_m	20	6	6	186,621

$$Node_{total} = Cskip(0) \times R_m + (C_m - R_m) + 1 \tag{2.7}$$

According to Eq. (2.7), the total number of nodes in Fig. 2.20 is obtained as $6 \times 2 + (5 - 2) + 1 = 16$, in which includes the coordinator. Changing the parameters C_m, R_m, and L_m can change the maximum number of nodes in a network. Table 2.3 shows a summary of network capacity according to the given parameters of C_m, R_m, and L_m. Each ZigBee device is assigned a logical 16-bit address by the Zigbee coordinator or router when joining a ZigBee network. Therefore, the maximum number of network nodes in any ZigBee network is $2^{16} = 65{,}535$. The last row in Table 2.6 is not realistic.

Cskip(*d*) is also used as the offset value to assign addresses to routers and its end devices. Assuming the address of the coordinator is fixed. The address of the first router R_1 is equal to the address of the coordinator +1.The following equation is used to assign an address to router R_n:

$$R_n = R_1 + (n - 1) \times Cskip(d) \tag{2.8}$$

Where d is the depth of the network at the upper level, i.e. the coordinator level. In the same way, each router device assigns an address to each of its descendants. For example, router R_1 initially assigns an address to one of its children, which is one greater than its own address. It then uses *Cskip(d + 1)* as the offset value to assign addresses to the other children connected to it. The rest of the routers use the same procedure.

ZigBee PRO offers stochastic address assignment. This means that each node, when joining the network, is assigned a random number for 0–65,536 as the address. If the new address of the node has been used for any existing device, a conflict notification will be announced and another address should be assigned to the new device.

2.4.4 ZigBee Management Mechanisms

The main feature of the ZigBee standard is its efficient and effective management mechanism designed for the application layer. The definition for the ZigBee management mechanism consists of the address management, profile management, device & service discovery and binding.

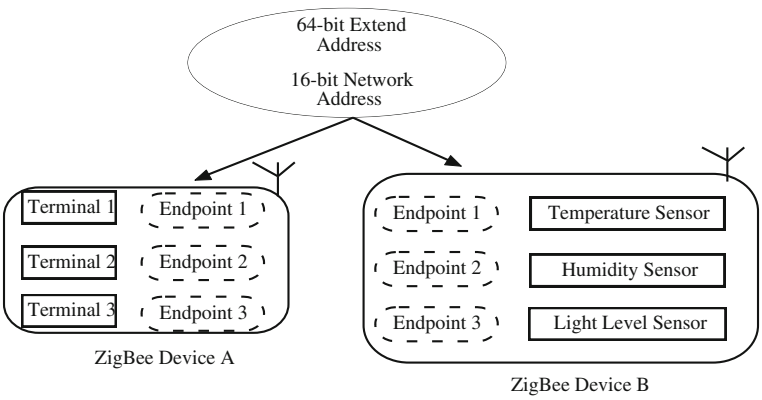


Fig. 2.21 Address management in ZigBee

2.4.4.1 ZigBee Address Management

As the IEEE 802.15.4 standard is used to construct the bottom layer (PHY and MAC) of the ZigBee stack, the 64-bit extended address and the 16-bit network address are both available for a ZigBee network's use. However, they are not sufficient to identify multiple objects sharing the same physical address. In the ZigBee specification, the concept of Endpoint addressing is specified to solve the problem.

Figure 2.21 shows the address usage in a ZigBee network, in which there are two ZigBee devices A and B need to communicate with other. Device A has three terminals, which correspond to the three sensors on devices B respectively. If terminal 1 on device A wishes to establish communication with the temperature sensor on device B, it can request device A to establish a wireless communication channel to the device B, using either its IEEE 802.15.4 64-bit extend address or its 16-bit network address. The problem is how to make device B recognizes that the communication is for the temperature sensor, rather than one of the other two sensors? The ZigBee specification defines a sub-level addressing mode—Endpoint, to help the system distinguish the multiple objects existing on one physical device. "Endpoint" is a kind of categorization, which virtually exists in the stack. Each ZigBee device can support up to 240 virtual objects (endpoint 0 is used for endpoint management). Each virtual object has its own property and can be independent from other objects. If the starter of the communication specifies which endpoint it is looking for, the ZigBee stack running on the destination ZigBee device can easily locate the target object. The concept of Endpoint in the ZigBee specification is useful, particular for wireless sensor networks. A sensor node is normally equipped with more than one sensor for executing multiple sensing tasks.

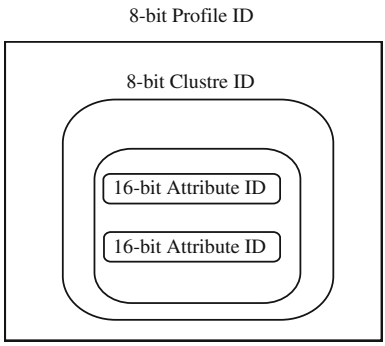


Fig. 2.22 ZigBee profile management

2.4.4.2 ZigBee Profile Management

Profile management is the communication fundament in the ZigBee specification. It consists of the agreements on messages, message formats and processing actions that have been well defined to enable the cooperation in the system. By following the same profile, the different components are able to create an interoperable, distributed application. And also, the products from different manufacturers can seamlessly communicate without worrying about compatibility. Figure 2.22 shows the concept of a profile.

In Fig. 2.22, the Profile ID is an 8-bit number, which specifies the property of the current profile. The ZigBee Alliance has specified a number of profiles (Home Control Stack Profile, Building Automation Stack Profile, Plant Control Stack Profile, etc.) which are called public profiles. The manufacturers can define their own profiles for the specified applications, which are called private profiles. The developers can obtain a profile ID by applying to the ZigBee Alliance. The profile ID must be unique due to administrative issues. If the profile is defined for research purpose, users do not have to apply for permission. Table 2.7 shows the ZigBee public profiles and corresponding IDs. The public profile ID range is from 0x0000 to 0x7fff. The private profile ID ranges from 0xbf00 to 0xffff.

Attributes indicate the function or data of a device connected to a node. For example, the attribute of a light switch represents the position (status) of the

Table 2.7 ZigBee public profiles (Elahi and Gschwender 2009)

Profile name	Profile ID
Industrial process control and monitoring (IPM)	0x0101
Home automation (HA)	0x0104
Commercial building management (CBM)	0x0105
Telecom applications (TA)	0x0107
Personal, home, and hospital care (PHHC)	0x0108
Advance metering initiative (AMI)	0x0109

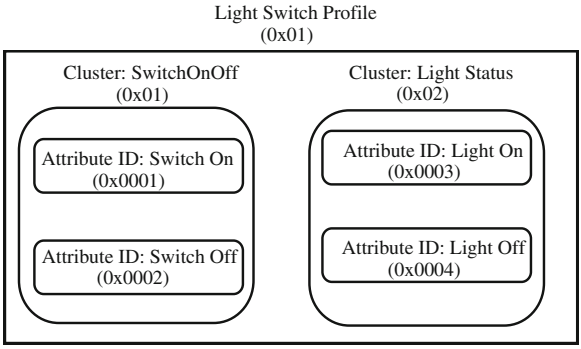


Fig. 2.23 Example of using ZigBee profile

switch, which can be on or off. The attribute identifier is a 16-bit number used to specify the actual data item, i.e. the attribute, passes between the ZigBee devices.

A cluster is a collection of attributes and commands that is used to perform a specific function, which associates with data flowing out of, or into the device. From example, a SwitchOnOff cluster is used to turn on or off a switch device. The cluster identifier is an 8-bit number which lies on at the second level of the profile management. The cluster identifier is unique in the scope of a specific profile. An application profile could have more than one cluster. By associating with the cluster id, the attribute id expresses the actual command to the applications.

Figure 2.23 illustrates an example of using profile management. In Fig. 2.23, a light switch profile with profile id 0x01 is set to manage the lighting system. Two clusters are defined: SwitchOnOff and LightStatus. The cluster SwitchOnOff is for controlling the instruction implementation from the users. By recognizing the incoming command containing the corresponding cluster id and attribute id, the local system is able to correctly execute the instruction of switching on or off a light. The cluster LightStatus is for outputting the light status upon receiving a request from the users. By examining the formatted message containing the cluster id (LightStatus) and the attribute id (Light On/Off), the users will be shown the current status of the light.

2.4.4.3 Device and Service Discovery

To simplify and standardize the service provision, the ZigBee specification provides the mechanism of discovering devices and services in the network. The device discovery command supports both the IEEE 64-bit and 16-bit network address and can be sent either as a broadcast or unicast message. A network should have a primary discovery cache device, which can be a router or a coordinator, for storing node descriptors of the devices that are in sleep mode. Before any device goes into sleep mode, it transmits its descriptor information to the primary discovery cache. It is the primary discovery cache device that responds when the requested device is in sleep mode. The actual implementation is performed by

the ZigBee Device Objects (ZDO). For example, if a ZigBee device newly joining the network wishes to know the network address of the ZigBee coordinator but it only knows the coordinator's 64-bit extend MAC address, or if the device wishes to know the network address of the devices which can provide the function of controlling a light, the ZDO can help send out the formatted broadcast queries to the network, or a unicast query to a specified device, and obtain the results when the discovery is finished. The ZDO is a kind of protocols defined in the ZigBee stack. Each ZigBee device running a ZigBee stack has it's own ZDO instance, which can deal with the information processing under the ZDO management without any user intervention. What the developers should consider is how to design the query submission and process the returned results.

2.4.4.4 ZigBee Binding

A useful feature of the ZigBee specification is its support for the concept of binding, which is a logical relation between two endpoints located in different devices. In the sensor network application development, the situation of sending control message from a single point to multiple destinations or from multiple destinations to a single point, or from a single point to a single destination is often encountered. The situation is shown in Fig. 2.24, in which three situations are present:

- A single light switch controls more than one light. The situation often happens when a central switch is designed for use in a warehouse.
- A single light switch controls a single light. This is the normal situation happening in daily life.
- More than one light switches control a single light. This is often used for the design of the light control in the corridor or on the stairs.

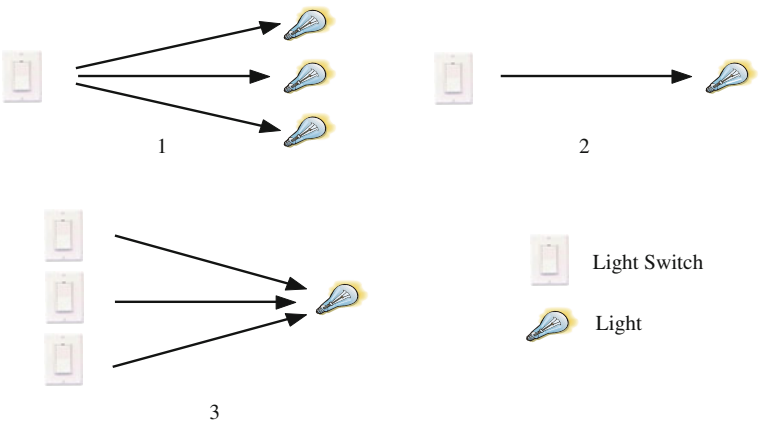


Fig. 2.24 Message sending in wireless sensor network for lighting control

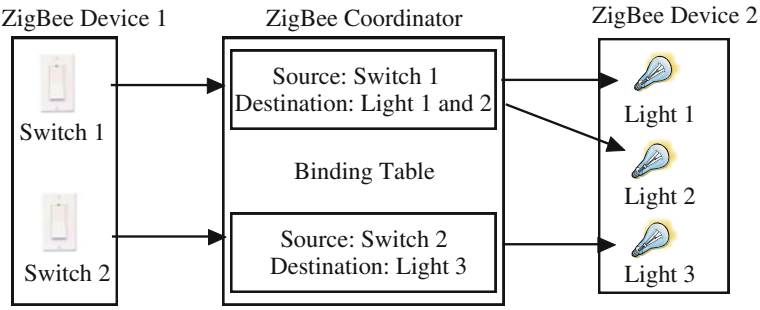


Fig. 2.25 Implementation of binding in ZigBee networks

As the procedure for handling the situations described in Fig. 2.24 will introduce massive duplicated of work using the traditional method, the mechanism of binding can make the operations much more convenient. The implementation of binding is shown in Fig. 2.25.

In Fig. 2.25, the coordinator is selected to store the binding table as it is supposed to be power on during the whole network's lifetime. There are two entries created and recorded in the table, the first entry records the source address and endpoint of Switch 1 and the matched addresses and endpoints of Light 1 and Light 2. The second entry records the address and endpoint of Switch 2 and the matched address and endpoint of light 3. If Switch 1 needs to turn on Lights 1 and 2, it can send the instruction and its own address to the coordinator. Upon receipt of the instruction, the coordinator will search the table and find out the two addresses of Lights 1 and 2. Then the coordinator replaces the destination addresses of the instruction with those of Lights 1 and 2 and automatically sends the instruction out. Therefore, Switch 1 can control Lights 1 and 2 via the binding between them. And the instruction can be processed quickly, which improves the overall execution efficiency.

2.5 6LoWPAN and Wireless Sensor Network

6LowPAN stands for IPv6 over IEEE 802.15.4 low-power wireless personal area networks (L-WPAN) and was developed by the Internet Engineering task Force (IETF) in 2007 (Kushalnagar et al. 2007). IPv6 is the newest version of the Internet Protocol. 6LoWPAN enables IPv6 directly working over IEEE 802.15.4 low-power wireless sensor networks. Consequently, individual wireless node in a 6LoWPAN based wireless sensor network become accessible from the Internet. A straightforward technical definition of 6LoWPAN given by Shelby and Bormann (2009) is *6LoWPAN standards enable the efficient use of IPv6 over low-power, low-rate wireless networks on simple embedded devices through an adaptation layer and the optimisation of related protocols.*

The benefits of making wireless sensor networks Internet enable include (Kushalnagar et al. 2007):

- It brings interoperability as it allows the use of existing network infrastructure based on IP-based protocols.
- Wireless devices can be connected easily to the Internet without the need for gateways.
- Enabling IP also enables the network to use all of the IP-based technologies such as proxies, which are well known and proven to work for higher level services in a large-scale network.
- Established Application protocols and data models such as HTTP, SNMP and DPWS etc. can be used.
- Transport protocols can be used to provide some reliability in a network with unreliable links.
- IP technology promotes innovation by providing all the standards and related documents available to anyone.
- Many protocols are already available for commissioning and managing the IP-based networks which can be used.

Figure 2.26 shows the protocol architecture for 6LoWPAN, where an adaptation layer, or called a LoWPAN layer, is added between the MAC layer and the IPv6 network layer. The function of adaptation layer is to perform the following tasks:

- Compress the IPv6 header
- Fragment the IPv6 payload
- Compress the UDP header

The detail can be found in the 6LoWPAN specification (Kushalnagar et al. 2007). The UDP (user datagram protocol) and the Internet control message protocol (ICMP) are used in 6LoWPAN. Adaptation between IPv6 and IEEE 802.15.4 is performed by routers at the edge of 6LoWPAN, referred to as edge routers.

Figure 2.27 shows the position of edge routers in the integration of WSN and the Internet. Each LoWPAN consists of an edge router, a number of LoWPAN routers (R) and a number of LoWPAN hosts (H). Additionally, there is a remote server on the Internet. 6LoWPAN enables IPv6 for simple embedded devices over low-power wireless networks by efficiently compressing headers and simplifying IPv6 requirements. When connecting a LoWPAN to the Internet or another IP network, there are several issues to be considered (Shelby and Bormann 2009):

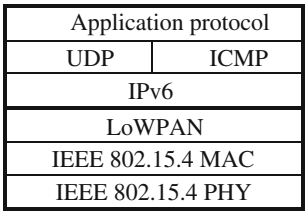


Fig. 2.26 6LoWPAN protocol stack

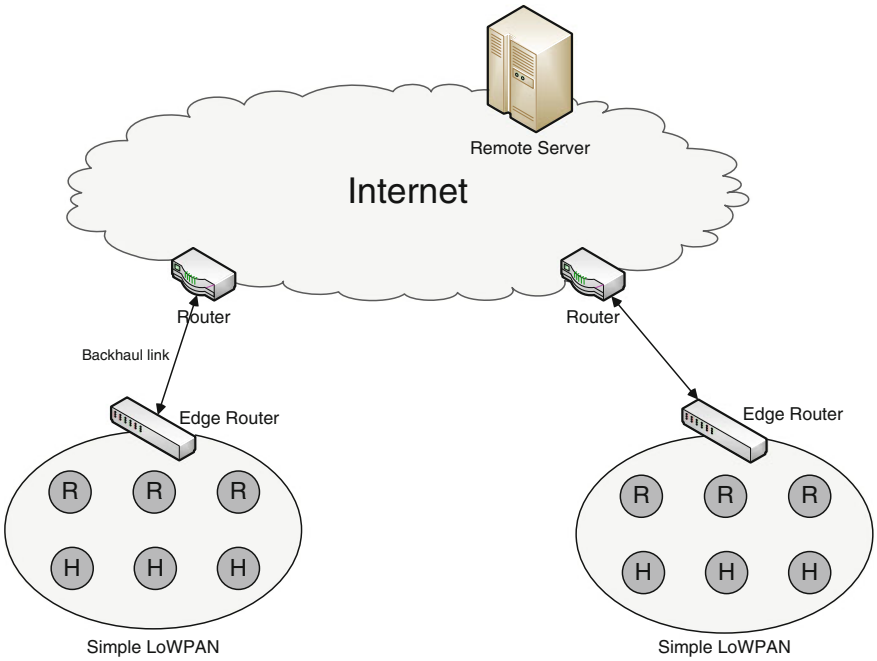


Fig. 2.27 Example of 6LoWPAN deployment

- Maximum transmission unit: Applications of 6LoWPAN should minimize packet sizes and avoid forcing a LoWPAN to fragment IPv6 packets.
- Application protocols: End-to-end application protocols should make use of UDP and compact payload formats to suit for use with 6LoWPAN nodes.
- Firewalls and network address translators: When connecting 6LoWPAN with the Internet, the issues of firewalls and network address translators are not avoidable.
- IPv4 interconnectivity: IPv4 and IPv6 are concurrently used in the Internet. It might be necessary for 6LoWPAN nodes to interact with IPv4 nodes or across IPv4 networks.
- Security: Connecting 6LoWPAN nodes with the Internet brings benefits and risks as well. Security should always be a major concern.

2.6 Summary

This chapter introduces the principle of wireless sensor networks, particularly IEEE 802.15.4, ZigBee and 6LoWPAN. Both ZigBee and 6LoWPAN are built on top of IEEE 802.15.4. Therefore, IEEE 802.15.4 lays down the foundation for low-rate, low-power wireless sensor networks.

It might be true that developers of wireless sensor networks often deal with ZigBee and 6LoWPAN stacks rather than IEEE 802.15.4 stack. These two LoWPAN stacks are the most popular and are currently independent of each other. ZigBee cannot communicate directly with the Internet due to its lack of native IP stack processing. There are a few approaches, which aim to interconnect 6LoWPAN with ZigBee. The first approach is to put the IPv6 stack on top of a ZigBee network layer. A global unicast IPv6 address is assigned to every ZigBee node; conversely every IPv6 node is assigned with a ZigBee short address. The gateway is responsible for dealing with sending and receiving all network traffic. It will then handle encapsulation and decapsulation of all the transmitted packets from and to an IPv6 network, or WAN respectively (Wang et al. 2007). The second approach is a dual stack design, with both the 6LoWPAN stack and the ZigBee stack working on the same IEEE 802.15.4 MAC layer. This approach allows both the 6LoWPAN and ZigBee stack to coexist on the same 802.15.4 MAC. Although it enables both IPv6 and ZigBee functions to be applied to a same node, however only one of the specifications can be used at any time. Finally, a gateway that allows both 6LoWPAN and ZigBee devices to be converted to IPv6 would be desirable (Hossen et al. 2010).

References

Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., Cyirci, E.: A survey on sensor networks. IEEE Commun. Mag. **40**(8), 102–114 (2002)

Aschenbernner, J.R.: Open systems interconnection. IBM Syst. J. **25**(3/4), 369–379 (1986)

Elahi, A., Gschwender, A.: ZigBee Wireless Sensor and Control Network. Prentice Hall, NJ (2009)

Gutierrez, J.A., Callaway, E.H., Barrett, R.L.: Low-Rate Wireless Personal Area Networks Enabling Wireless Sensors with IEEE 802.15.4. IEEE Press, New York (2004)

Hossen, M.S., Kabir, A.F.M.S., Khan, R.H., Azfar, A.: Interconnection between 802.15.4 devices and IPv6: impleications and existing approachs. Int. J. Comput. Sci. **7**(1), 19–31 (2010)

IEEE: Part 15.4: Wireless medium access control (MAC) and physical layer (PHY) specifications for low-rate wireless personal area networks (LR-WPANs) (2003)

Kushalnagar, N., Montenegro, G., Schumacher, C.: IPv6 over low-power wireless personal area networks (6LoWPANs): Overview, assumptions, problem statement, and goals. RFC4919, Internet Engineering Task Force (2007)

Lewis, F.L.: Smart environments: Technology, protocol and applications. In: Cook, D.J., Das, S.K. (eds.) Wireless Sensor Networks, 1st edn, pp. 13–46. Wiley, New York (2004)

Nicopolitidis, P., Obaidat, M.S., Papadimitriou, G.I., Pomportsis, A.S.: Wireless Networks. Wiley, New York (2003)

Schurgers, C., Srivastava, M.B.: Energy efficient routing wireless sensor networks. In: Military Communications Conference on Communications for Network-Centric Operations: Creating the Information Force, vol. 1, pp. 357–361

Shelby, Z., Bormann, C.: 6LoWPAN—The Wireless Embedded Internet. Wiley, New York (2009)

Wang, R.C., Chang, R.S., Chao, H.C.: Internetworking between ZigBee/802.15.4 and IPv6/802.3 Network. In: SigComm Conference on IPv6 (IPv6'07), Kyoto, Japan

ZigBee: ZigBee specification, version 1.0. Available at www.zigbee.org (2004)

ZigBee: ZigBee stack advanced user guide, JN-UG-3045 Revision 1.2, 6 Mar 2008

Chapter 3
Hardware Design for WSNs

Keywords Microcontroller · Sensor · Hardware design · Power management · Energy scavenging

3.1 General Wireless Sensor Node Architecture

The first task in establishing any WSN is to develop the sensor nodes, which have to meet the requirements that arise from the specific applications. Because a large numbers of sensor nodes are deployed in WSNs, they should be small, cheap, energy efficient and have sufficient storage, computation, and communication capabilities. Therefore, the sensor nodes cannot use long lasting, large capacity batteries or the main power supply as their power sources because of the size constraint. The requirements for cheap cost and energy efficiency determine that the sensor nodes should use low power processors and small radio transceivers with a limited bandwidth and transmission range. Hence, sensor node design is constrained by the required computation and communication capabilities. Normally, the sensor nodes are comprised of four main subsystems:

- a sensing subsystem consisting of one or more sensors and actuators to monitor the physical environment;
- a computing subsystem consisting of a microcontroller or microprocessor with memories to store and process data collected by the sensing subsystem;
- a communication subsystem consisting of a short range radio system for wireless data communication;
- a power supply subsystem, which normally use batteries, which powers the whole sensor node. A power generator might be included in this power supply subsystem if energy-harvesting technologies are employed.

A typical wireless sensor node architecture is depicted in Fig. 3.1. The sensing subsystem can be divided into two parts. The first, is a basic sensing device that includes sensor elements, which acquire information from the physical world around the node, and converts it into an analogue signal, an Analogue–Digital

S.-H. Yang, *Wireless Sensor Networks*, Signals and Communication Technology,
DOI: 10.1007/978-1-4471-5505-8_3,

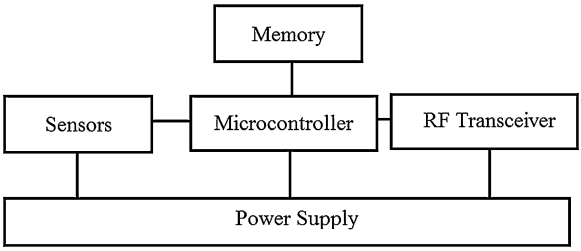


Fig. 3.1 Sensor node architecture

Converter (ADC) then converts this analogue signal into a digital value. The second part is an intelligent sensing device that can provide additional functionality such as the pre-processing of sampling data and compensation for any measurement error. The sensing subsystem must provide an interface compatible with the computation task carried out by the microcontroller.

The computing subsystem carries out all of the computing work, such as processing the sensor data, implementing data fusion, managing system battery operation, setting parameters for the sensors, and executing high-layer protocol, for instance, the ZigBee specification. Power consumption here is mainly caused by the microprocessor. Working at a lower operating voltage is one solution for reducing power consumption. Another, is to divide the work period of the subsystem into different modes and to switch among these modes to ensure that the microprocessors always run in a power-saving mode.

The communication subsystem is responsible for transmitting and receiving data frames. It is well known that transmitters consume most of the energy for wireless communication and the distance of communication depends on the transmitting power. Most radio frequency (RF) modules provide a mechanism to enable the transmitting power to be controlled on demand by a program, so that the energy can be used more effectively.

The power supply subsystem consists of a battery and a converter from direct current to direct current (dc–dc) with an assistant control circuit. The dc–dc converter provides various voltages to support all devices in the system and enables them to work in different modes so that power consumption can be reduced.

3.2 System-on-Chip and Component-based Design

There are two ways to design wireless sensor nodes in terms of the RF modules available in the market. One is based on a System-on-Chip (SoC) solution and the other is based on a component-based design. Many RF module manufacturers such as Chipcon, Microchip, Freescale, and others, provide SoC RF modules where a RF module, microprocessor, flash memory, RAM, ADC, and some special electronic circuits are integrated onto a single chip. These SoC RF modules make the

hardware design of wireless sensor nodes quick, easy and reliable, as there are only a few extra components, which need to be added. The drawback of the SoC solutions is the lack of flexibility and, consequently, some special requirements may be difficult to meet. The component-based design provides the designers with a full range of flexibility as they can choose all the components required, such as RF module, microprocessor, and other electronic elements and design different layouts for the sensor node based on the components chosen. Consequently, it can achieve lower cost and higher performance. However, it might be complicated and time consuming. This chapter focuses on the ZigBee compliant SoC design as it can significantly shorten the time to bring the design to market. For reference purposes Table 3.1 gives the comparison of commonly used RF modules currently available in the market.

The Jennic JN5139 microchip, described here, is an example of these chip and used in the design case study in this chapter. As a typical SoC solution, the JN5139 modules integrate all of the 2.4 GHz RF components required. It consists of a JN5139 microcontroller, 1Mbit of serial flash memory and peripheral circuits. The 1Mbit serial flash memory holds the application code that is loaded into the microcontroller during the boot sequence. This Jennic module removes the need to perform expensive RF design and testing. Sensor nodes can be designed by simply supplying the power to the Jennic module and connecting switches, actuators and sensors to the module's IO pins. A block diagram of the Jennic module is shown in Fig. 3.2. The module is composed of a 32-bit RISC (Reduced Instruction Set Computer) CPU, a memory system, a rich mixture of analogue and digital peripherals, and an IEEE 802.15.4 compliant 2.4 GHz transceiver, all in one chip.

3.3 Design Guidelines

There are some design guidelines, which must be taken into consideration in WSN hardware design.

Table 3.1 ZigBee chips comparison

Manufacturer	Part number	Supply voltage (V)	Sleep current (UA)	TX current (MA)	RX current (MA)	TX power (DBM)	RX sensitivity (DBM)
Atmel	AT86RF231	1.3–3.6	0.02	14.3	13.2	3	−101
Freescale	MC13192	2.0–3.4	1	30	37	4	−91
Texas instruments	CC2420	2.1–3.6	20	17.4	18.8	0	−95
Microchip	MRF2J40	2.4–3.6	2	19	23	0	−95
Jennic	JN5139	2.7–3.6	2.6	37	37	2.5	−96
Ember	EM2420	2.1–3.6	0.5	17.4	19.7	10	−94
ST	SN260	2.1–3.6	1.0	35.5	35.5	2.5	−100

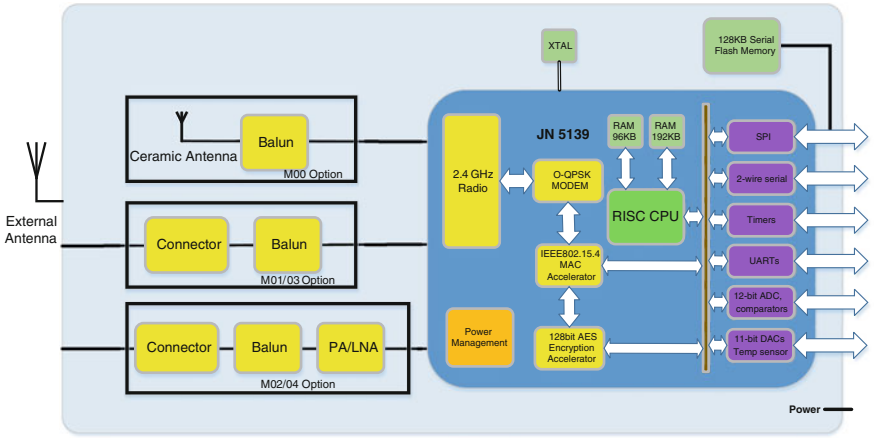


Fig. 3.2 JN5139 module block diagram (Jennic 2010)

- **Lifetime**: In many applications, sensor nodes are placed in locations that are difficult, if not impossible, for human access. Therefore, regularly replacing batteries for WSNs is not feasible and the lifetime of the sensor node relies heavily on the limited energy supply initially provided on the sensor node. For WSNs hardware design, each node must be designed to manage its local power consumption in order to maximize the total network lifetime. Hence, each circuit design should ensure minimal power consumption. Alternatively, any energy-scavenging device provided on the sensor node can increase the lifetime of WSNs.
- **Coverage:** WSNs must cover the area of interest, i.e. the coverage, determined by the application requirements. The transmission power and the deployment of sensor nodes should be determined by consideration the coverage required. When the coverage becomes larger, the sensor nodes will consume more power for data collection and transmission, which means that the lifetime of the WSNs will be shortened by a limited power supply.
- **Robustness:** The wireless sensor nodes might be required to work in an uncertain or harsh environment. The WSNs must be designed so that it can tolerate and adapt to individual node failures, while maintaining the functionality of the WSNs as a whole if such failure occurs.
- **Communication:** Sensor nodes should have low communication data rate and low communication power consumption.
- **Time Synchronization:** In order to save energy, sensor nodes should be kept in sleep mode once their tasks are completed, and they must be able to wake up periodically or on demand. Precise time synchronization will enable different sensor nodes to work together with other members of the network.
- **Security:** Some applications of WSNs such as many military uses require data confidentiality and a security algorithm must be fulfilled by the WSNs. Hence,

the microprocessor chosen for the individual sensor nodes must be capable of performing complex encryption and authentication algorithms.
- **Cost and Size:** The deployment of WSNs in an application may require thousands of sensor nodes. The cost and size of individual sensor nodes should be the minimal possible to make such deployments affordable.

Based on the above considerations, wireless sensor node hardware design can be divided into the following steps; microcontroller selection, RF communication device selection, sensing devices design, and power supply device design.

3.3.1 Microcontroller Selection

The microcontroller is the heart of the sensor node and is responsible for the acquisition, processing, compression, recording and storage of data. Generally, a SoC microcontroller integrates a CPU, flash storage, RAM, with analogue and digital peripherals onto a single chip. Using a SoC microcontroller in the sensor node design minimizes the design and testing costs and is considered as an ideal choice for WSNs. There are some practical factors, which should be borne in mind in choosing the correct type of microcontrollers.

- **Performance:** The performance level of the microcontroller can significantly impact the sensor node's power dissipation characteristics. This is because a microcontroller with better performance has a higher power consumption. As the required level of microcontroller performance varies with different application, the performance of an ideal microcontroller for the WSN system should meet the desired performance level of the application rather than choosing the best one.
- **Operating Mode:** To save energy, microcontrollers usually have various operating modes, including Active, Idle, and Sleep modes, where each mode is characterized by a different amount of power consumption. The transition times for entering and exiting the sleep mode are important in determining the total power consumption of the sensor nodes. The faster a microcontroller can enter and leave the sleep mode, the more often the sleep state can be used and the less energy overall the node will consume. Thus, different modes' power consumption levels, the transition time, the transition power, and the amount of time spent by the microcontroller in each mode all have a significant impact on the total energy consumption of the sensor node.
- **Voltage Requirements:** The operating voltage range of a microcontroller can have a significant impact on the system performance and sensor selection. Traditional low voltage microcontroller with the operating voltage range between 2.7 and 3.3 V are selected.
- **CPU Speed:** As the power consumption of the microcontroller scales linearly with frequency, the optimal speed of the CPU is determined by the amount of

data analysis and in-network processing that must be performed on a sensor node.

- **Peripheral Support:** As the microcontroller is specifically designed to interact with external devices, it should have general-purpose digital I/O pins, Analogue-to-Digital Converter (ADC), Comparators and some digital communication interfaces such as RS-232, UART, I2C or SPI.
- **Memory:** Depending on the size of the application programs of WSNs, the microcontroller should have a large enough memory to host the application program. Programs usually reside in the flash memory, which is writable by the microcontroller in boot mode.
- **Software Available:** The available software base can also influence the microcontroller selection. For instance, many researchers and engineers prefer to write their code in C or C++. Hence, a microcontroller supporting these software-developing environments might be considered as an ideal one for them.
- **Cost and Size:** Low cost and compact size of the microcontroller is another consideration in microcontroller selection. Normally, a chip with integrated MCU and radio is often chosen because of its low cost, small size, and ease of development.

3.3.2 Communication Device Selection

A communication device is used to exchange data between sensor nodes. Normally, a communication device consists of a low power radio system, which includes a RF transceiver (antenna), a power amplifier, and a digital baseband. The radio system is usually the largest consumer of power in the WSNs, therefore optimizing its power consumption can result in a significant improved lifetime for the whole system. There are several factors that affect the radio system selection.

- **Wireless Technologies:** There are several wireless technologies available for commercial applications, such as ZigBee, Wi-Fi and Bluetooth. These three technologies are briefly compared in Table 3.2. The ZigBee-based technology is normally selected for WSNs to achieve simple wireless communications with short-range distances, limited power, low data throughput, low cost, and small size.
- **Transmission Range**: Radio transmission range determines the minimum distance between any two sensor nodes and therefore the coverage of the WSNs. Several factors, such as the transmission power, the range of the transceiver, the sensitivity of the receiver, the gain and efficiency of the antenna, and the channel encoding mechanism control the range. Optimizing the transmission power can significantly save energy in sensor nodes. Higher transmission power will result in a higher SNR (signal-to-noise ratio) and lower BER (bit-error rate). Moreover, the more energy put into a signal transmission the farther it should travel, which could increase the coverage as well as reduce inference from other wireless systems. It is essential to set the power at a suitable level to ensure both

Table 3.2 A comparison between ZigBee, Wi-Fi, and bluetooth wireless technologies

Name	ZigBee	Wi-Fi	Bluetooth
Standard	802.15.4	802.11a,b,g	802.15.1
System resources	50–60 KB	>1 MB	>250 KB
Battery life (days)	100–1000	1–5	1–7
Network size	65,536	32	7
Bandwidth (kb/s)	20–250	11,000	720
Maximum transmission range (m)	100+	100	10
Security	128 AES plus application layer security	–	64 and 128 bit encryption
Operating frequency	868 MHz (Europe) 900–928 MHz (NA), 2.4 GHz (worldwide)	2.4 and 5 GHz	2.4 GHz

satisfactory communication quality and coverage, together with reasonable power consumption. Normally, Omni-directional antennas are preferred in ad-hoc WSNs because they allow nodes to effectively communicate in all directions.

- **Modulation Type:** Within RF communication devices, one of the functions is to convert the digital signal into an analogue signal for transmission. This process is known as modulation. The standard modulation mechanisms include amplitude modulation such as *amplitude-shift keying* (ASK), frequency modulation such as *frequency-shift keying* (FSK) and phase modulation such as *phase-shift keying* (PSK). ASK uses changes of amplitude to represent zero and one. FSK uses changes of frequency to represent zero and one. PSK uses the phase of the signal to represent the binary data. Normally, *Quadrature phase-shift keying* (QPSK) modulation is selected for WSNs, where each signal is shifted by increments of 90-degree phase.
- **Bit Rate**: Unlike many other high performance data networks, WSNs do not require high bit rates communication. 10–200 kbps of raw network bandwidth is normally sufficient for most applications.
- **Turn-on Time:** A radio's ability to quickly enter and exit low power sleep modes is extremely important for efficient operation in WSNs. If a radio's *turn-on-to-receive* time is more than a few tens of milliseconds, it quickly becomes impossible to achieve the required duty cycle of less than 1 %, where the duty cycle is defined as the fraction of time that a system is in an active state.

3.3.3 Sensing Device Design

A sensor is a device that measures a physical quantity and converts it into an electrical signal, which can be read by other electronic devices. It provides an interface between the physical environment and electronic equipment. There are a number of ways to classify sensors. From the power supply point of view, all the

sensors can be divided into two categories: passive and active. A passive sensor does not need any additional power source and directly generates an electrical signal in response to an external stimulus, such as a photodiode. An active sensor requires external power for its operation, such as temperature sensitive resistor. In terms of the type of sensor output singles, sensors can be classified into digital sensor and analogue sensor. Digital sensors output binary values to the microcontroller. Analogue sensors generate an analogue signal, usually a voltage, in response to an external variable. The third classification of sensors is based on their measurement properties, in which the sensors can be classified into thermal sensors, mechanical sensors, chemical sensors, magnetic sensors, radiant sensors, and electrical sensors. Table 3.3 summaries sensing principles used in WSNs. These sensors can be located at a position either inside or close to the phenomenon to be measured.

An ideal sensor should have high sensitivity, accuracy and repeatability together with low power dissipation, and cost while being easy to use. Unfortunately, we usually cannot have all these factors in one sensor and have to choose

Table 3.3 Sensor types and transduction principles (Cook and Das 2004)

Measured	Transduction principle
Physical properties	
Pressure	Piezoresistive, capacitive
Temperature	Thermistor, thermo-mechanical, thermocouple
Humidity	Resistive, capacitive
Flow	Pressure change, thermistor
Motion properties	
Position	E-mag, E-vision, GPS, contact sensor
Velocity	Doppler, Hall effect, optoelectronic
Angular velocity	Optical encoder
Acceleration	Piezoresistive, piezoelectric, optical fibre
Contact properties	
Strain	Piezoresistive
Force	Piezoelectric, piezoresistive
Torque	Piezoresistive, optoelectronic
Slip	Dual torque
Vibration	Piezoresistive, piezoelectric, optical fibre, sound, ultrasound
Presence	
Tactile/Contact	Contact switch, capacitive
Proximity	Hall effect, capacitive, magnetic, seismic, acoustic, RF
Distance/Range	E-mag(sonar, radar, lidar), magnetic, tunnelling
Motion	E-mag, IR, acoustic, seismic (vibration)
Biochemical	
Biochemical agents	Biochemical transduction
Identification	
Personal features	Vision
Personal ID	Fingerprints, retinal scan, voice, heat plume, vision analysis

application-specific sensors in the design of WSNs. Three considerations should be taken into account in the sensor selection process.

- Environmental condition such as the operating temperature, pressure, light, humidity and position to locate the sensor node.
- Design parameters such as the purpose of measurement, output signal type, data transmission technique to be used, microcontroller to be used, and signal conditioning required.
- Sensor parameters: sensor package size, response time to the change of environment to be measured, accuracy of the measurement, transducer lifetime, power requirement, available accessories, measurement range, maximum error that can be tolerated, transducer excitation voltage, current drawn from the excitation supply and cost.

The following list gives a group of typical sensor requirements. The requirements from any application could consist of one or more of them. Table 3.4 describes the most commonly used definitions in technical specifications of any sensor (Ristic 1994).

Table 3.4 Descriptions of sensor technical specifications

Specification	Description
Absolute sensitivity	The ratio of the change of the output signal to the change of the measure (physical or chemical quantity)
Relative sensitivity	The ratio of a change of the output signal to a change in the measure normalized by the value of the output signal when the measure is 0
Cross sensitivity	The change of the output signal caused by more than one measure
Direction dependent sensitivity	A dependence of sensitivity on the angle between the measure and the sensor
Resolution	The smallest detectable change in the measure that can cause a change of the output signal
Accuracy	The ratio of the maximum error of the output signal to the full-scale output signal expressed in a percentage
Linearly error	The maximum deviation of the calibration curve of the output signal from the best fitted straight line that describes the output signal
Hysteresis	A lack of the sensor's capability to show the same output signal at a given value of measurement regardless of the direction of the change in the measure
Offset	The output signal of the sensor when the measure is 0
Noise	The random output signal not related to the measure
Cutoff frequency	The frequency at which the output signal of the sensor drops to 70.7 % of its maximum
Dynamic range	The span between the two values of the measure (maximum and minimum) that can be measured by a sensor
Operating temperature range	The range of temperature over which the output signal of the sensor remains within the specified error

- The sensor should be cost-effective and installable at a reasonable cost.
- The operation and maintenance of the sensor should be easily performed with minimal special training requirements.
- The sensor should be able to function continuously and reliably for a desired period of time, and should be replaceable or renewable at a reasonable price.
- The sensor should be controllable by the microcontroller.
- The sensor should be small in size and portable so it can be easily carried.
- The sensor should be energy efficient with low energy consumption if it is battery driven.
- The sensor might be required in use at hazardous locations and/or harsh environments.

3.3.4 Power Supply Device Design

The power supply is a crucial part of wireless sensor nodes. Sensor nodes in most WSNs are battery driven and the lifetime of the chosen battery directly determines the lifetime of the sensor node. Battery lifetime is determined by parameters such as battery dimension, type of electrode material used, and diffusion rate of the active materials in the electrolyte. Average current consumption, maximum current consumption, and cost of batteries should be taken into consideration in the selection of batteries for sensor nodes.

Average Current Consumption: The average current consumption of sensor nodes is the most significant factor that affects the choice of battery. The current drawn from the battery varies with the state of the sensor node. To calculate the average current, it is necessary to determine the consumption for each state of the sensor node and the amount of the time spent in each state. The average current consumption for each sensor node $I_{average}$ can be determined using the following equation:

$$I_{average} = I_1 \times \frac{t_{on1}}{t_{total}} + I_2 \times \frac{t_{on2}}{t_{total}} + \ldots + I_n \times \frac{t_{onn}}{t_{total}} \tag{3.1}$$

where $I_i, i = 1, 2, \ldots, n$ is the current in the given state i, $t_{oni}, i = 1, 2, \ldots, n$ is the time spent on drawing current in the state i, and t_{total} is the total time spent in a cycle of the states.

Once the average current consumption is calculated, Eq. (3.2) can be used to calculate the capacity required to power the device over the desired battery lifetime as

$$I_{average} \times T_{runtime} \tag{3.2}$$

where $T_{runtime}$ is the desired run-time of the battery in hours. This can calculate the least energy needed by the sensor node.

Maximum Current Consumption: Every battery has a rated current capacity, specified by the manufacturer. Drawing higher current than the rated value leads to

a significant reduction in the battery life. Hence, to avoid battery life degradation, the amount of current drawn from the battery should be kept below the rated current capacity, identified as the maximum current consumption.

Size and Cost: The size and cost of batteries are also important factors, which should be taken into consideration. When looking for a low-cost and small-size battery for a device, always check the specifications of the candidates, since the performance of a given battery type across different manufacturers can vary significantly.

3.4 Design Case

In this section we illustrate the design of a wireless temperature and CO (carbon monoxide) sensor node for building safety monitoring. We choose the Jennic JN5139 as the microcontroller and communication device and two AAA batteries as the power source. The discussion below focuses on the selection of the temperature and CO sensors and the circuit design.

3.4.1 Temperature Sensor Design

There are many different types of temperature sensors in the market, and they can be simply classified into four groups: Thermocouple, Resistance Temperature Detector (RTD), Thermistor and Integrated Circuit (IC) temperature sensors. A comparison of these four group sensors is summarized in Table 3.5.

Table 3.5 Comparison of temperature sensors

Attribute	Thermocouple	RTD	Thermistor	IC Temperature sensors
Temperature range	−190 to 1,821 °C	−200 to 850 °C	−90 to 130 °C	−55 to 150 °C
Accuracy	Poor	High	Medium	High
Response time	Fast	Moderate	Fast	Fast
Stability	Not as stable	Stable over long periods	Moderate	Stable over long periods
Linearity	Moderate	Good	Poor	Best
Sensitivity	Low	Medium	Very high	Very High
Interchange ability	Moderate	Excellent	Poor	Moderate
Repeatability	Poor	Good	Moderate	Excellent
Size	Small to large	Medium to small	Small to medium	Small to medium

Thermocouple devices, which were discovered by Thomas Seebeck in 1822, form one of the most common industrial thermometers. It consists of two dissimilar metals, joined together, which produces a small unique voltage at a given temperature. Normally, thermocouples are considered to be the smallest, fastest and most durable temperature measurement solution (Swanson 2010). It can be used over an extremely wide temperature range and in harsh environmental conditions. There are, however, three disadvantages with thermocouples. Firstly, temperature measurement with a thermocouple requires two temperatures to be measured. Secondly, the relationship between the process temperature and the thermocouple output voltage is nonlinear. Thirdly, a special compensation technique is required as a result of the low accuracy.

A RTD sensor is a positive temperature coefficient sensor that contains a resistor, which changes value as its temperature changes. It has developed a reputation for high accuracy, low drift, wide operating range, repeatability, and reasonable linearity. The limitations of RTD sensors are that it cannot be used for high temperature applications and also it is less sensitive to small temperature changes. A RTD is the sensor of choice when extreme stable and precise measurements are the most important criteria.

A thermistor sensor is also a type of resistor whose resistance varies with the change of temperature. But the difference between thermistors and RTDs is that the material used in a thermistor is generally a ceramic or polymer, while RTDs use pure metals. The most common thermistors have a negative temperature coefficient of resistance. Characteristics include moderate temperate range, low cost, poor but predictable linearity. Thermistors are ideal for measuring applications that require highly accurate sensitivity over a relatively narrow range of temperatures.

IC temperature sensors, which are produced in the form of ICs, are complete silicon based sensing circuits with either analog or digital outputs. The use of IC temperature sensors is limited to applications where the temperature is within the range −55 to 150 °C. But they have several advantages in comparing with other types of temperature sensors. Firstly, IC temperature sensors are considered to be a small, accurate, and inexpensive type of temperature sensors possessing excellent linearity. Secondly, they are easy to interface with other device such as amplifiers, and microcontrollers.

Because of the advantages of IC temperature sensors, the *Maxim* IC temperature sensor *DS18B20* is chosen as the temperature sensor in this design. The *DS18B20* digital thermometer provides 9-bits to 12-bits Celsius temperature measurements and has an alarm function with non-volatile user-programmable upper and low trigger points. The *DS18B20* requires one data line for communication with a central microprocessor. The features are shown below: (Dallas Semiconductor 2008):

- Power supply range: 3.0–5.5 V;
- Measures temperatures from : −55 to + 125°C;
- Accuracy: ±5°C from −10 to +85°C;

Table 3.6 Pins descriptions

DS18B20 (TO-92)	Name	Function
1	GND	Ground
2	DQ	Data Input/Output.
3	V_{DD}	Optional V_{DD}. V_{DD} must be grounded for operation in parasite power mode

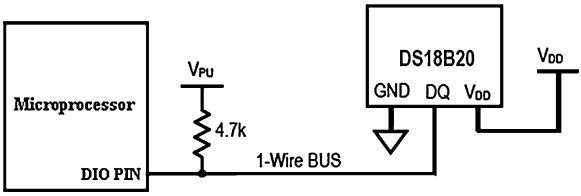


Fig. 3.3 *DS18B20* application diagram (Dallas Semiconductor 2008)

- Thermometer resolution: user selectable from 9 to 12 bits;
- Converting temperature to 12-Bit digital word in 750 ms;
- Available in 8-Pin SO, 8-Pin SOP, and 3-Pin TO-92 packages.

The 3-Pin TO-92 package is chosen for the design and Table 3.6 shows the Pin Descriptions of the *DS18B20*. The application schematic diagram is shown in Fig. 3.3. The *DS18B20* chip is powered by an external power supply to the V_{DD} pin. The DQ pin is connected to the DIO pin of the microprocessor.

3.4.2 CO Sensor Design

Gas sensors are widely used in the detection of various gases in the air as part of toxic and combustible gas detection. They interact with various gases, which triggers an electrical output. The following procedure should be taken into account in the selection of gas sensors:

- Determine the target gas and any possible background gases in the monitoring area. The presence of background gases may cause gas sensor failure.
- Determine the concentration of the target gas. Generally, the ranges or the concentration of the gases to be measured should be 3–5 times the actual monitoring concentration.
- Determine the range of the working temperature where the gas sensor is to be installed.
- Determine the acceptable power consumption, as many gas sensors demand high power supply.
- Determine the required response time of gas sensing, as many gas sensors have a long response time.
- Determine the affordable cost and size.

Gas sensors can be classified as electrochemical, semiconductor, catalytic, and infrared. Normally, electrochemical gas sensors contain two or three electrodes in contact with an electrolyte. They measure the concentration of the target gas by oxidizing or reducing the target gas at an electrode. The electrochemical reaction results in an electric current that passes through an external circuit. An external amplifying circuit is needed to measure this electric current, which indicates the concentration of the target gas. The benefits of electrochemical sensors include compact package size, robustness, no or little external power requirement and cost effective in large-scale production. However, the lifetime of electrochemical sensors is often less than 3 years and response time is around 30 s. The cost of replacement sensors is high, especially in a large-scale deployment. Electrochemical sensors are mainly used to detect toxic gases.

A semiconductor gas sensor uses a semiconductor at a gas-detecting device. A particular gas can be detected by measuring changes in an electrical characteristic of the semiconductor. The great interest in semiconductor gas sensors comes from their numerous advantages, such as small sizes, long life, quick response time and high sensitivities in detecting very low concentrations of gases. But they normally need a 5 V external power supply to ensure that the sensors remains in a working mode.

Catalytic gas sensor consists of a filament such as a platinum-iridium alloy, embedded within a ceramic pellet. Catalytic gas sensor uses heated bare coils of platinum wire to burn the target gas and the heat generated by the burning process produces a change in the resistance of the filament. This change is measured using a simple wheatstone bridge circuit. The sensor itself is quite simple in design and easy to manufacture. The advantage of using this technique is that it measures the gas directly. The sensors need additional power to heat the bare coils and consequently are not suitable for battery driven sensors. Catalytic gas sensors are used primarily to detect combustible gases.

Ai infrared gas sensor is considered to be a “non-reactive” gas sensor. The working principle is based on the principle that the target gas absorbs some of the infrared wavelengths of the light passing through it, while other wavelengths pass through uninterrupted. An infrared gas sensor uses an infrared light source to illuminate a volume of the gas and the amount of absorption that talks place is related to the concentration of the target gas. Long lifetime, no contact with the target gas, high accuracy and reliable concentration measurements are considered to be the primary advantages of this technology. However, high cost and high power consumption are the drawbacks of using infrared gas sensors.

For building the safety-monitoring example, an electrochemical CO sensor, Figaro *TGS5042*, is chosen as the sensor node design because it does not need power for the sensor itself. The features of the *TGS5042* CO gas sensor are summarized below: (Figaro 2010):

- Battery operable
- High repeatability/selectivity to CO
- Linear relationship between CO gas concentration and sensor output

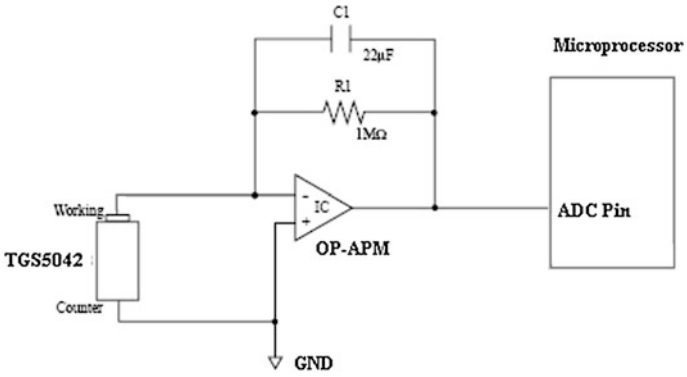


Fig. 3.4 TGS5042 application diagram (Figaro 2010)

- Simple calibration
- Long life
- Target gases: Carbon monoxide
- Typical detection range: 0–10,000 ppm
- Output current in CO: 1.2–2.4nA/ppm
- Operating temperature range: −40 to +70 °C
- Response time: less than 60 s

Figure 3.4 shows the basic measurement circuit of the *TGS5042*. The sensor generates a minute electric current, which is converted into the sensor output voltage by an op-amp/resistor combination. The output is fed into the Jennic 5139 module through an ADC pin.

3.4.3 Sensor Node Circuit Design

The temperature sensor shown in Fig. 3.3 and the CO sensor shown in Fig. 3.4 can be integrated with a microcontroller into a single circuit. Figure 3.5 illustrates the schematic diagram of a sensor node equipped with both a temperature and a CO sensor. The Jennic 5139 module is chosen as the microcontroller and the communication device, which is shown on the left hand side, and the circuit for both the CO sensor and the temperature sensor is shown on the top right. The bottom right is a circuit for hosting two AAA batteries as a power supply. This schematic diagram can be converted to a printed circuit board (PCB) design and therefore a wireless sensor node equipped with temperature and CO sensors can be manufactured. The list of components and parameters is given in Table 3.7.

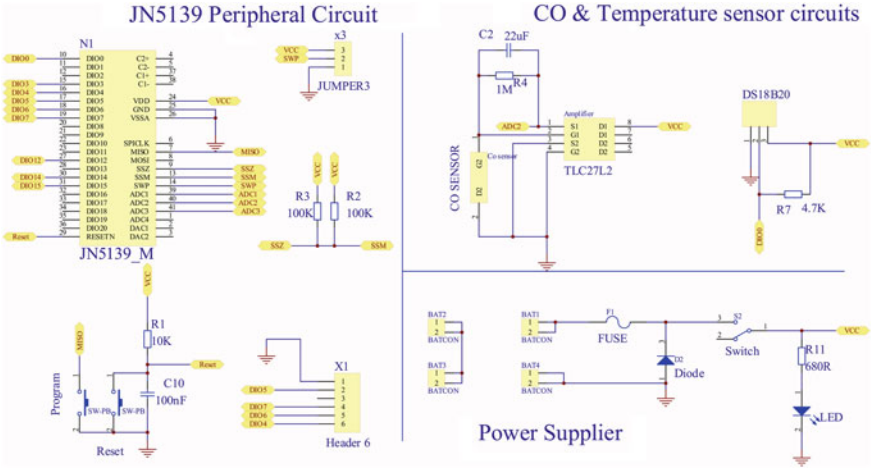


Fig. 3.5 Temperature and CO gas sensor schematic diagram

Table 3.7 List of components and parameters

Designator	Description
J1	6 pin Header 2.45 mm
J2	3 pin Header 2.45 mm
BAT1-BAT4	Battery holder
R1	Resistor 680 R
R2	Resistor 100 K
R3	Resistor 100 K
R4	Resistor 10 K
R5	Resistor 4.7 K
R6	Resistor 1 M
FUSE	500 mA Fuse
D1	BAS21 diode
Ds1	LED
C1	Capacitor100 nF
C2	Capacitor 22 uF
S2	Switch
Program, Reset	Buttons
U1	JN5139
U2	TGS5042 CO sensor
U3	TLC27L2 Op AMP
U4	DS18B20 Temperature sensor

3.5 Power Management

Power management is one way to prolong the sensor node lifetime when it is battery-driven. It aims to avoid energy waste and improve energy efficiency by turning off the power whenever possible or switching the sensor system to a

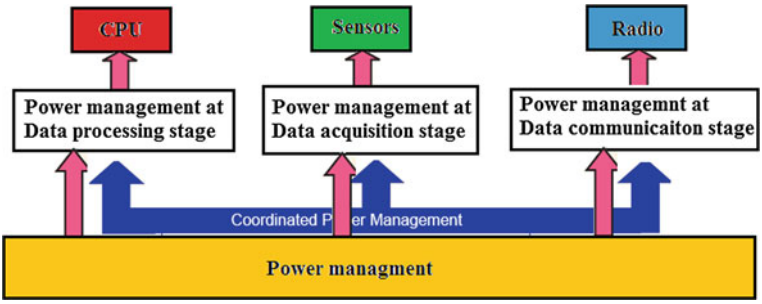


Fig. 3.6 Power management of sensor nodes

low-power state. Energy consumption in a sensor node can be classified into "useful" and "wasteful" sources. The useful energy consumption can be used for sensing environment, data processing, transmitting or receiving data, processing query requests and forwarding queries and data to neighbouring nodes. The wasteful energy consumption in WSNs can occur in data acquisition, data processing, and data communication. Therefore, power management should deal with energy consumption for these three functions as shown in Fig. 3.6. Note this section only addresses power management related to hardware design, i.e. chip level power management at the data acquisition stage. Power management at both the data processing and data communication stages will be addressed in the remaining chapters of the book.

The power consumption of the individual components is determined jointly by the voltage of the power supply, the current consumption of the individual components, together with their running time. The first two items are fixed once the electronic components are selected. The running time of the components can be divided into two parts—working time and idle time. Components in an idle state consume the same amount of energy as those in the working state. Power management at the data acquisition stage is to turn on the sensor power only after the node receives an acquisition command from the microcontroller and turn it off when the sensor enters the idle state.

Taking the designed temperature and CO sensor shown in Fig. 3.5 as an example, the *DS18B20* temperature sensor has a standby mode, which cost 0.003 mW power. The sensor driver, which will be introduced in next chapter, keeps the temperature sensor in the standby mode unless it is required for a sensing task. The *TGS5042* CO gas sensor current consumption is 4 mA in a working state and is the biggest power consuming part of the sensor node. Power consumption of idle sensors can be avoided by powering off the unit. This radical solution requires a controllable switch on the sensor power supply line. Hence, a P-channel switch *J177* as shown in Fig. 3.7 is employed in the CO gas sensor circuit to switch off the CO gas sensor when it is in idle mode. A control signal from the microcontroller turns off the CO gas sensor circuit when it is in idle state and consequently the current consumption at this state is zero. The energy consumption of the CO

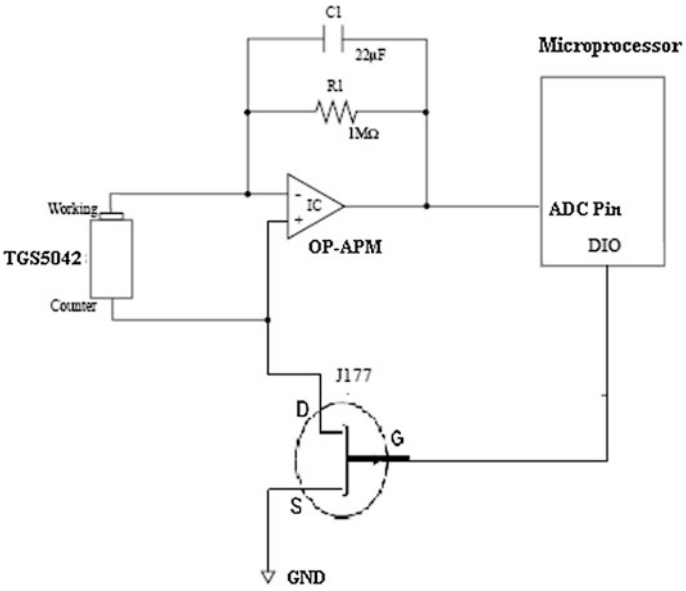


Fig. 3.7 CO gas sensor with a P-channel switch

gas sensor without the P-channel switch is 2,640,000 μc in a cycle, and can be reduced to 240,000 μc if equipped with a P-channel switch.

3.6 Energy Scavenging

Energy scavenging, also called energy harvesting, is another way to prolong the lifetime of a sensor node. Most people do not realize that there are abundant energies constantly around us such as solar, thermal, wind, and radio frequency energies. If these energies could be scavenged and transferred into electrical energy they can then be used to power the wireless devices, then the previously crucial battery limitations of WSNs could be removed. Depending on the different sensors and different environments these sensors are deployed in, various energy scavenging methods have been utilized including:

- Light energy: sunlight or man-made light, which can be captured via solar panels, photo sensors;
- Thermal gradient energy: waste thermal energy from heaters, furnaces and engines et al.;
- Radio Frequency energy: from satellites, TV base stations, mobile phones transmission stations, and other wireless electronics,
- Mechanical energy: vibration, mechanical stress, strain and wind;

Table 3.8 Power densities of energy harvesting resources

Energy source	Power density (uW/cm^3) 1 year lifetime
Solar (outdoors)	15,000-direct sun 150-cloudy day
Solar (indoors)	6-office desk
Vibrations	200
Acoustic noise	0.003 @ 75 Db 0.96 @ 100 Db
Daily temp. variation	10
Temperature gradient	15 @ 10 K gradient
Shoe inserts	330

- Human body: a combination energy generated from bio-organisms or through body movements;
- Other energy: chemical and biological sources.

Power densities of commonly used power scavenging sources are compared in Table 3.8 (Roundy 2003). The most accessible energy source is solar energy, which can be harvested through Photovoltaic (PV) conversion and has the higher power density of those compared. Therefore, a solar energy harvesting system is chosen as an example in this section to illustrate how localized energy harvesting systems can be designed to supplement battery supplies to prolong the lifetime of wireless sensor networks.

Solar is the most powerful source of nature light, and an inexhaustible source of energy. Photovoltaic (PV) technology is used to convert solar energy directly into electricity. In practice, a solar cell is commonly used to harvest solar power. Figure 3.8 depicts the functional architecture of a solar energy harvesting system, which comprises three subsystems: an energy harvesting unit, a maximum power point tracking (MPPT) unit, and a power management unit.

3.6.1 Solar Energy Harvesting Unit

A solar cell is commonly used to harvest light intensities. Numerous types of commercial solar cells are available. The trade-off between price, dimensions and the efficiency of solar cells needs to be considered and two Centennial Solar MC-zSP0.8-NF-GCS (Multicomp 2010), connected in parallel, were chosen as the main solar panel in the energy harvesting unit as a single solar cell would not be not sufficient to power the whole sensor node.

3.6.2 Maximum Power Point Tracking Unit

The main function of the MPPT unit is to deliver the maximum power from the solar panel to the energy reservoir. The MPPT unit consists of a Pulse Width Modulation (PWM) DC/DC converter and the MPPT peripheral circuit.

Because solar energy varies over time with the change of light intensity, an energy harvesting interface circuit with high power transfer efficiency is required to convert the scavenged power into a smooth value before storage in an energy reservoir. The type of DC/DC converter used, is determined by both the strength of the power harvested and the operating voltage of the energy reservoir. LTC3401 (Linear Technology 2001) DC/DC converter is chosen here as its conversion efficiency is over 85 % in the 10–50 mA output current range while its output voltage is set to 4.1 V (Park and Chou 2006). There are two advantages in using the PWM DC/DC converter rather than charging the energy reservoir by directly connecting a diode with the solar panel. Firstly, this PWM DC/DC converter enables energy harvesting to continue even when the open circuit voltage of the solar panel is lower than the voltage of the energy reservoir. Secondly, using a diode to block the reverse current flow from the reservoir to the solar panel causes a 0.7 V drop in the output voltage, but the PWM DC/DC converter can avoid this voltage drop.

The MPPT peripheral circuit consists of a miniaturized PV module and a comparator. There exists a linear relationship between the open-circuit voltage of the miniaturized PV module and the maximum power point of the main solar panel when both of them are exposed to the same light radiation. Therefore the comparator with an input from the main solar panel and a feedback from the miniaturized PV module can be used to perform MPPT in terms of the above linear relationship. In the design here, we have chosen a Hamamatsu S1087 (Hamamatsu 2002) as the miniaturized PV module and use it as a radiance sensor. There is no need of any additional power supply for this radiance sensor.

3.6.3 Power Management Unit

A power management unit is used to store the scavenged energy and ensure its effective use. The power management unit shown in Fig. 3.8 consists of a primary buffer, a secondary buffer and a control charger circuit. There are two reasons for the use of multiple buffers in the power management subsystem here. As the light intensity in the environment changes, the generating voltage would vary over time, and consequently it is hard for the energy-harvesting unit to power the target system directly. Therefore, a high-density energy storage element such as a rechargeable battery must be employed to accumulate the available energy delivered by the energy-harvesting unit. On the other hand, the rechargeable battery has limited recharge cycles and lifetime, which limits the lifespan of the whole system. In order to prolong the system's lifespan for as long as possible, the

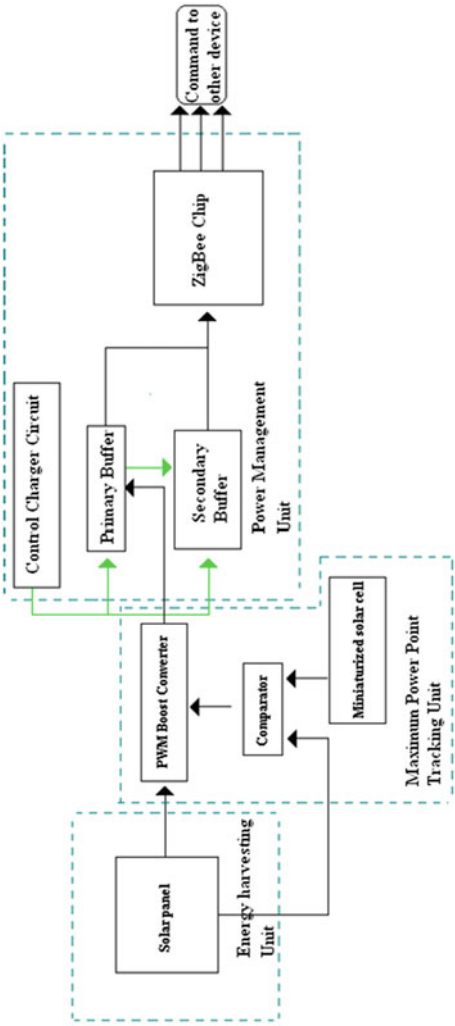


Fig. 3.8 Function diagram of solar energy harvesting system (Lu and Yang 2009)

access to the rechargeable battery must be minimized and consequently, the target system should be directly powered by the energy-harvesting unit most of the time. Therefore another energy buffer is required. The two-buffer design here is in the same spirit as the design by Prometheus (Jiang and Polastre 2005). A primary buffer, a super-capacitor, which is directly charged by the harvesting panel, powers the target system when enough power is available. Otherwise, the target system draws current from the secondary buffer, a rechargeable battery. Furthermore, if a sufficient light source is available, the primary buffer charges the secondary buffer and powers the target system simultaneously.

The primary buffer is directly charged by the energy-harvesting unit and its main purpose is to minimize access to the secondary buffer in order to prolong the

lifetime of the energy harvesting system. The primary buffer must have the capability to handle high levels of energy throughput and frequent charge cycles but does not need to hold energy for a long time. Basically, super-capacitors have a much longer lifetime, higher efficiency, higher power density, fast and simpler charging circuit than rechargeable batteries. This means that super-capacitors fit all the requirements for the primary buffer. Therefore, two 22F super-capacitors have been chosen as the primary buffer in this design.

The secondary buffer is used only when the energy in the primary buffer is exhausted, and needs to hold energy for a long period of time, i.e. have low current leakage. Rechargeable batteries have higher energy density, lower breakdown voltage, and lower leakage current. For these reasons, rechargeable batteries are the ideal option for the secondary buffer.

A control charge circuit is needed to optimize the use of the harvested power for the sensor node. We adopted the Ambimax design (Park and Chou 2006) for the control charge circuit. By comparing the terminal voltage of the super-capacitors with a pre-defined threshold voltage, the control charge circuit determines which power source, either the primary buffer or the secondary buffer, should power the target system at any moment. When the rechargeable batteries are not fully charged and the voltage of the sup-capacitor is higher than a second pre-defined threshold voltage and the rechargeable batteries are replenished by the super-capacitor. Furthermore, overcharging and undercharging of the rechargeable battery is protected by software installed in the ZigBee chip.

3.6.4 Design Case

A complete circuit for a solar energy- harvesting system is shown in Fig. 3.9. In the schematic diagram, a solar panel with the DC–DC converter circuit is shown at the bottom left side to harvest solar energy from environment. The MPPT circuit, shown at the top left side, is employed to keep the solar cell working at the

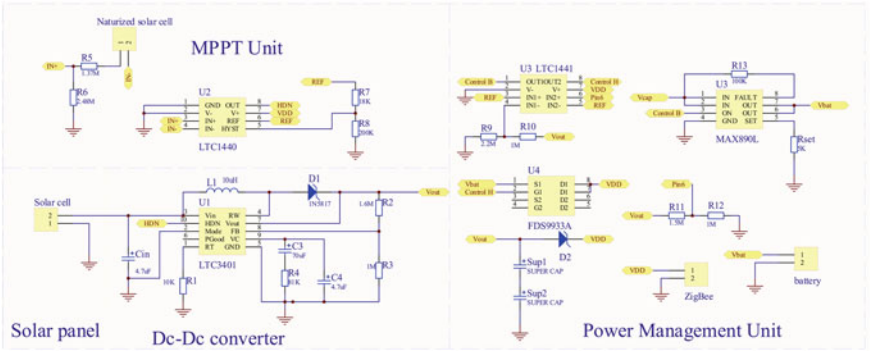


Fig. 3.9 A complete solar energy harvesting circuit

Table 3.9 List of components and their parameters

Designator	Description
Battery	Rechargeable battery
C3	Capacitor 70 uF
C4	Capacitor 4.7 uF
Cin	Capacitor 4.7 uF
D1	Diode1N5817
D2	Diode 1N5817
L1	Inductor CDR, 10uH
Naturized solar cell	Small size solar cell
R1	Resistor 10 K
R2	Resistor 1.6 M
R3	Resistor 1 M
R4	Resistor 81 K
R5	Resistor 1.37 M
R6	Resistor 2.48 M
R7	Resistor 18 K
R8	Resistor 200 K
R9	Resistor 2.2 M
R10	Resistor 1 M
R11	Resistor 1.5 M
R12	Resistor 1 M
R13	Resistor 100 K
Rset	Resistor 5 K
Solar cell	Solar cell
Sup1	22F Super capacitor
Sup2	22F Super capacitor
U1	LTC3401 Dc–Dc converter
U2	LTC1440 Comparator
U3	LTC1441 Dual Comparator
U4	MAX890L Charge control chip
ZigBee	Jennic sensor board

maximum power point. The right hand side is the power management circuit, which is used to maximize the lifetime of the system. It consists of a LTC1441 dual comparator, a charge control chip Max890L, two 22F super capacitors, and two rechargeable batteries. A ZigBee temperature and CO sensor node is connected with the solar energy harvesting system and the whole system was tested in an outdoor environment for one week. The sensor node worked autonomously as expected without any additional power requirement. During the daytime, the node was powered most of the time by the super-capacitor and the super-capacitor charged the battery when they had sufficient power. During the night, the node switched to use the batteries. The list of components and parameters is given in Table 3.9.

3.7 Conclusion

Hardware design is one of the most crucial steps in the design of WSNs, where energy consumption is the most critical concern. This chapter groups the basic structure of sensor nodes into a sensing part, a microcontroller part, a RF transceiver part, and a power supply part. Many wireless electronics manufacturers provide microcontroller, RF transceiver, and their peripheral circuits on an integrated circuit board, discussed above as the SoC solution. Sensor node designs based on the SoC solution are quicker, easier, and more reliable than the components based design. Various considerations on microcontroller selection, communication device selection, sensor device design, and power supply device design, have been summarized in this chapter and the use of them have been illustrated by a temperature and CO sensor node design. Power management and energy scavenging are two ways to overcome the constraints caused by the energy consumption and prolong the lifetime of WSNs. Switching the power supply when the sensor node is not in an active mode can help in the reduction of the energy consumption. A complete solar energy harvesting system has been designed in this chapter, showing the promising future of using this type of technologies in the design of WSNs.

References

Cook, D.J., Das, S.K.: Smart Environments: Technology, protocols and Applications. Wiley, London (2004)

Dallas Semiconductor: Maxim DS18B20 programmable resolution 1-wire digital thermometer. Available at http://datasheets.maxim-ic.com/en/ds/DS18B20.pdf (2008)

Figaro: TGS 5042. Available at http://www.figaro.co.jp/en/data/pdf/20101202115721_7.pdf (2010)

Hamamatsu: S1087 photodiode, 56NM, Ceramic, Max voltage 10 V. Available at http://www.farnell.com/datasheets/104399.pdf (2002)

Jennic: JN5139 Module datasheet, http://www.jennic.com/files/product_briefs/JN5139-xxx-Myy-PB_v1.2.pdf (2010)

Jiang, X., Polastre, J., and Culler, D.: Perpetual environmentally powered sensor networks, Information Processing in Sensor Networks, IPSN, pp. 463–468 (2005)

Linear Technology: LTC3401 1A, 3 MHz micropower synchronous boost converter. Available at http://cds.linear.com/docs/Datasheet/3401fb.pdf (2001)

Lu, X., Yang, S.H.: Solar energy harvesting for ZigBee electronics, Sustainability in energy and buildings, pp. 19–27 (2009). doi: 10.1007/978-3-642-03454-1

Multicomp: MC-SP0.8-NF-GCS Solar panel. Available at http://www.farnell.com/datasheets/925851.pdf (2010)

Park, C., Chou, P.H.: AmbiMax: autonomous energy harvesting platform for multi-supply wireless sensor nodes, IEEE SECON 2006 Proceeding, pp. 168–177 (2006)

Ristic, L.: Sensor Technology and Devices. Artech House, London (1994)

Roundy, S.J.: Energy scavenging for wireless sensor nodes with a focus on vibration to electricity conversion. Internal report of Engineering-Mechanical Engineering in the Graduate Division of The University of California, Berkeley (2003)

Swanson, C.: Sensor Selection Guide, Watlow Electric Manufacturing Company. Available at http://www.newark.com/pdfs/techarticles/Accurate_Temperature_Measurement-Sensors.pdf (2010)

Chapter 4
Embedded Software Design for WSNs

Keywords Sensor driver · Network establishment · Network management · Embedded software design · IEEE 802.15.4 · ZigBee

4.1 Introduction

Embedded software design is the most important and difficult task in wireless sensor network design and development. Here 'Embedded' has the actual meaning of 'build-in'. Embedded systems are everywhere. Typical examples are mobile phones, microwaves, digital cameras etc. Embedded software is computer software that plays an integral role in the electronics of the device it is supplied with. Embedded software is usually written for an application-specific microprocessor, which has low computation capability, low-cost, limited memory, and low power consumption. Embedded software is often run on a real-time operating system (RTOS) (Laplante 2004), and communication protocols designed for use in embedded systems are available as closed source from the microprocessor manufacturing companies. Embedded software must be uploaded to, verified and run on the appropriate microprocessor and integrated in the electronics after its design is completed. Ideally, embedded software should be designed after the associated hardware becomes available. With the development of various embedded software simulation environments embedded software design can be conducted in parallel with or even before the hardware design. An example of such an environment is COOJA (http://www.contiki-os.org/start.html), which is a WSN simulator using the Contiki operating system (OS), embedded software design can be conducted in parallel with or even before hardware design.

Embedded software and the corresponding hardware form an embedded system for a particular application. The process of embedded system design is shown in Fig. 4.1. Starting with a set of system requirements, then moving on to system architecture design and microprocessor selection, software design and hardware design are carried out in parallel before they are integrated together (Labrosse et al. 2008).

S.-H. Yang, *Wireless Sensor Networks*, Signals and Communication Technology,
DOI: 10.1007/978-1-4471-5505-8_4,

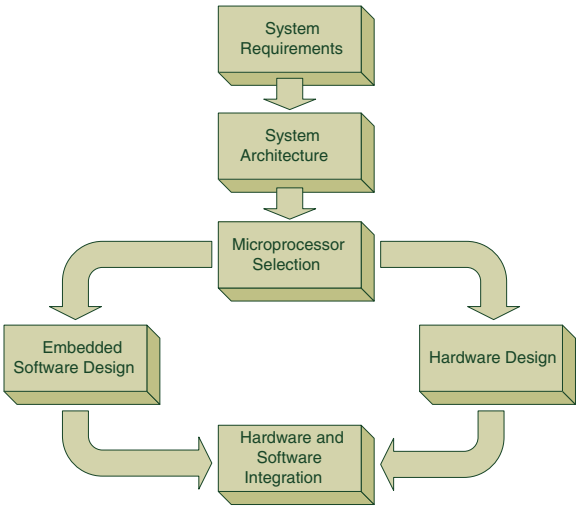


Fig. 4.1 Embedded system design process

4.2 Embedded Software Design for WSNs

The architecture of embedded software varies from system to system, depending on the microcontrollers and communication protocols used in the embedded systems. A WSN protocol stack was given in Chap. 2 (Fig. 2.2), which also forms the WSN software architecture. For the sake of simplicity, the protocol stack shown in Fig. 2.2 consists of four layers. These are from bottom to top: the hardware layer, the MAC (medium access control) layer, the network layer, and the application layer. Figure 4.2 gives a typical implementation of an embedded software architecture for a WSN (Jennic 2010). An application sits above the 802.15.4 stack, which in turn sits directly above the baseband hardware. The 802.15.4 stack is provided with defined entry points to request 802.15.4 actions and to initialize and register callbacks to the application. The application queue API, Integrated Peripherals API and Board API sit logically to the side of the 802.15.4 stack and are independent of it. Operations, such as communicating, accessing sensors, controlling the system, and scheduling tasks, can be implemented by calling an application programming interfaces (API). In other words, the user's application consists of a series of calls based on these APIs. As shown in Fig. 4.2, the communication services are provided by calling the IEEE 802.15.4 stack API; Accessing sensors and connecting with a local PC is implemented by calling the integrated peripherals API; hardware interruption is invoked by calling the

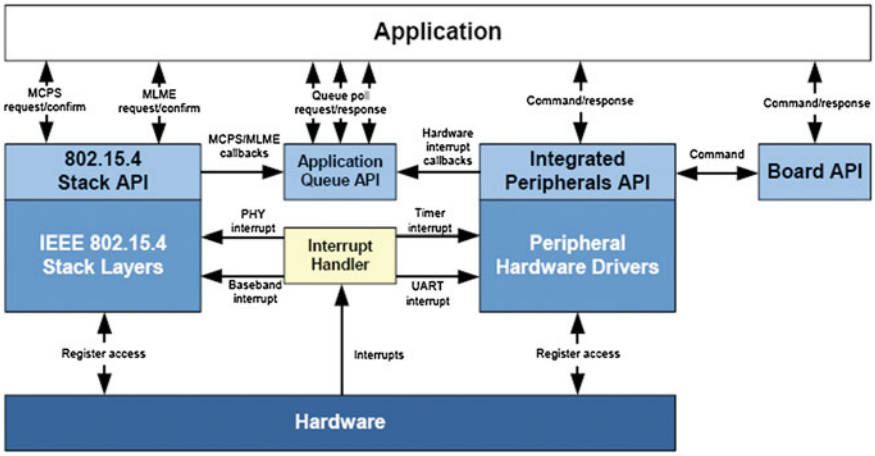


Fig. 4.2 Software architecture (Jennic 2010)

application queue API. The board API is available for development kits. Therefore, the design of embedded software for WSN in most applications becomes a process of using the APIs for various tasks.

4.2.1 Jennic ZigBee Application Development

Jennic provides a ZigBee application development process presented as a generic embedded software structure based on a Jennic microchip, as shown in Fig. 4.3. There are 10 predefined functions such as user tasks, and BOS/Stack tasks. The latter are Basic Operating Systems (BOS) tasks which do not need user intervention. The embedded software development task is to implement these 10 predefined functions.

When a sensor node is powered up, the program starts by executing the function "*AppColdStart*", in which the system is initialized. Any user variables or system peripherals such as timers or UART ports can be initialized in this function. Moreover, the essential ZigBee system parameters such as the radio channel and the network identification are configured here, to enable sensor node to join the correct WSN. Finally, the BOS is initialized and started, forcing the sensor node to process the hardware events. After initializing the system, the BOS then performs some internal functions before calling the function "*JZA_vAppDefineTasks*", which is where the user application can register any additional task with the BOS. By executing another initialization function "*JZA_boAppStart*", the registered ZigBee device can be run as a ZigBee Coordinator, Router or End device through a call to the ZigBee stack.

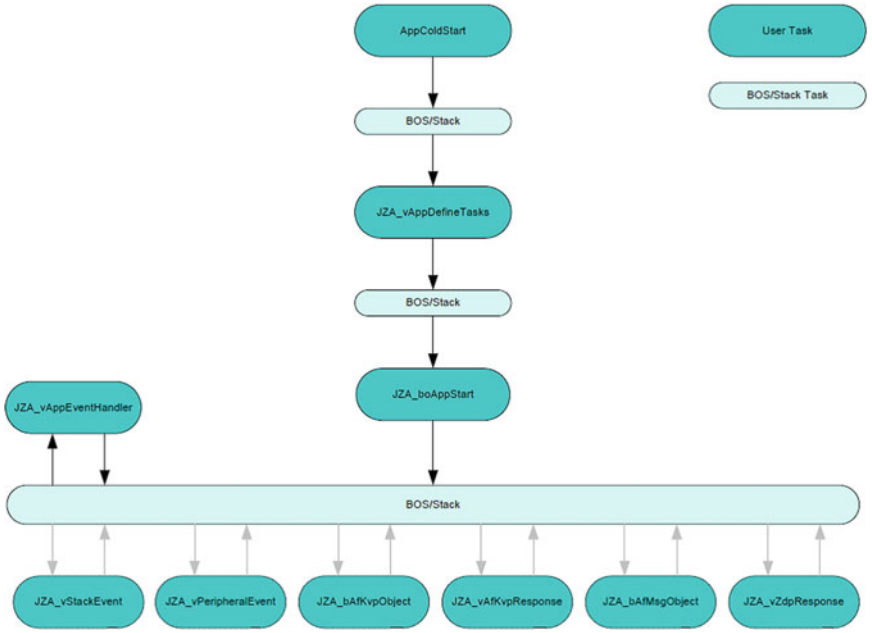


Fig. 4.3 Generic ZigBee embedded software structure (Jennic 2008b)

Once the BOS and the ZigBee stack are started, the BOS passes control to the user application through the following functions:

- *JZA_vAppEventHander*: This is a user application function, which is called by the BOS at regular intervals. Any user application code that requires regular execution should be placed here.
- *JZA_vStackEvent*: This function is called to handle miscellaneous events from the lower layers of the stack.
- *JZA_vPeripheralEvent*: This function is called when an interrupt is generated by a system peripherals, such as a timer firing or a DIO line being asserted. It is called while the processor is running in an interruption mode. The information about the interrupt is recorded in a simple FIFO queue to be later read by *JZA_vAppEventHandler*().
- *JZA_bAfKvpObject*: This function is called only when a Key Value Pair (KVP) command frame has been received from another node via radio. The application code should have been added in this function to handle the incoming command and, if necessary, generate a response.
- *JZA_bAfMsgObject*: This function is called only when a MSG frame has been received from another node, via radio. The application code should be added to this function to handle the incoming message.

- *JZA_vAfKvpResponse*: This function is called when a KVP response frame has been received from another node. The application code should be added in this function to receive and act on the response frame.
- *JZA_vZdpResponse*: This function is called when a response is received from a ZigBee Device Profile object.

After implementing these functions, the design of the embedded code for a sensor node is completed. Compiling and downloading the embedded code to the hardware should then follow.

4.2.2 Contiki 6LowPAN Application Development

Contiki (2012) is another embedded software environment for WSNs. All Contiki programs are called processes, which is a piece of code that is executed regularly by the Contiki operating system. Processes in Contiki are typically started when the system boots, or when a module that contains a process is loaded into the system. Processes run when an event happens, such as a timer fires or an external event occurs.

Embedded code in Contiki can run in one of two execution contexts: cooperative or pre-emptive. Code running in the cooperative execution context run sequentially with other code in the same context. Cooperative code must run to completion before any other cooperatively scheduled code can run. Pre-emptive code may interrupt the cooperative code at any time. When pre-emptive code interrupts the cooperative code, the cooperative code will not resume until the execution of the pre-emptive code has terminated. The concept of Contiki's two scheduling contexts is illustrated in Fig. 4.4. Processes always run in the cooperative context. The pre-emptive context is used by interrupt handlers in device drivers and by real-time tasks that have been scheduled for a specific deadline.

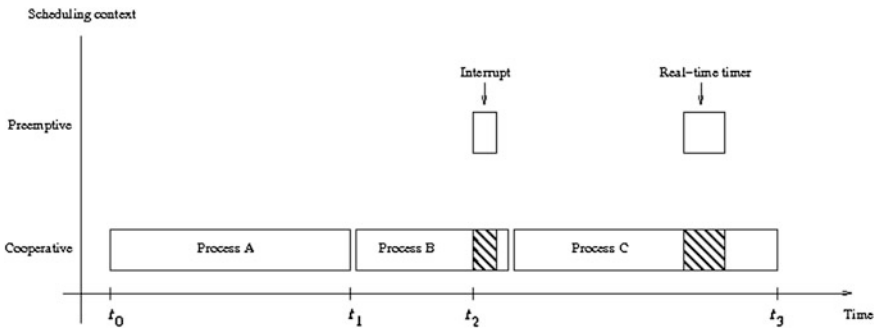


Fig. 4.4 Execution context of Contiki

A Contiki process consists of two parts: a *process control block* and a *process thread*. The process control block, which is stored in RAM, contains run-time information about the process such as the name of the process, the state of the process, and a pointer to the process thread. Its size might be very small and it requires only a couple of bytes of memory. The process thread is the code of the process and is stored in ROM.

A process control block is not declared or defined directly, but through the *PROCESS*() macro. This macro takes two parameters: the variable name of the process control block, which is used when accessing the process, and a textual name for the process, which is used in debugging and when printing out lists of active processes to users. The definition of the process control block corresponding to the Hello World example is shown below.

```
PROCESS(hello_world_process, "Hello world process");
```

As described previously, a process thread contains the code of the process. The process thread is a single proto-thread that is invoked from the process scheduler. An example is given below.

```
PROCESS_THREAD(hello_world_process, ev, data)
{
  PROCESS_BEGIN();
  printf("Hello, world\n");
  PROCESS_END();
}
```

Figure 4.5 gives a generic picture of a typical Contiki Process. It shows the sequence of definitions, as they should be included to write a working Contiki process. After the definition of the process control block 'AUTOSTART_PROCESS' is called to start the process followed by the definition of 'PROCESS_THREAD'. 'PROCESS_BEGIN' and 'PROCESS_END' must be called to start and terminate the process. The process code is inserted in the 'While' loop

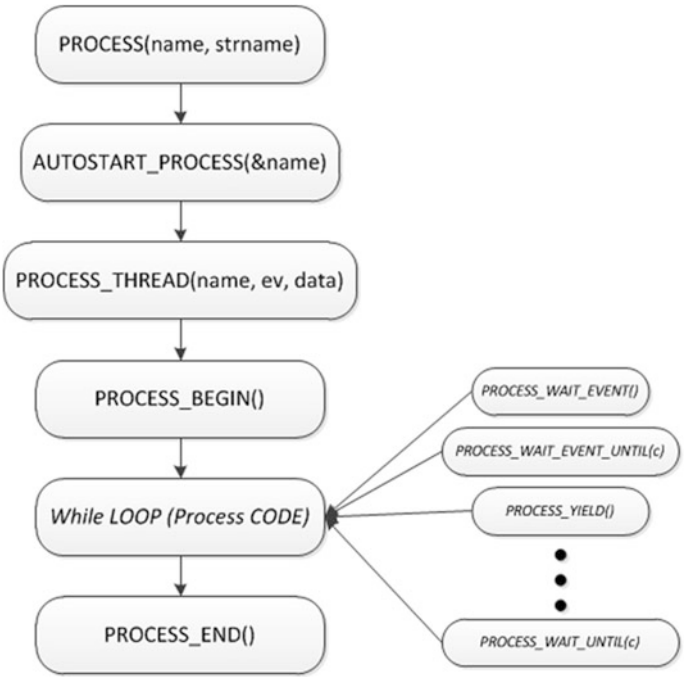


Fig. 4.5 Generic Contiki embedded software structure

between 'PROCESS_BEGIN' and 'PORCESS_END'. Many particular functions such as event waiting and event handling functions can be included there.

In the following example a timer is used in order to print out a particular line of text every three seconds. The process name is 'example_process'. A timer structure 'etimer' is implemented to generate timer event, named 'PROCESS_EVENT_TIMER'. The 'PROCESS_WAIT_EVENT_UNTIL' function is used in the 'WHILE' loop.

```
#include "contiki.h"
#include<stdio.h>
PROCESS(example_process, "Example process");
AUTOSTART_PROCESSES(&example_process);
PROCESS_THREAD(example_process, ev, data){
static struct etimer timer;
PROCESS_BEGIN();
  etimer_set(&timer, CLOCK_CONF_SECOND*3);
  while(1){
```

```
PROCESS_WAIT_EVENT_UNTIL(ev == PROCESS_EVENT_TIMER);
printf("Hello Mr. Yang To Saudi Arabia\r\n");
etimer_reset(&timer);
  }
PROCESS_END();
}
```

4.3 Sensor Driver Development

From the view of the hardware components, a typical wireless sensor network consists of sensors and wireless transmission modules. To enable the operations of these components, embedded software is needed, components of which are usually called "drivers". Through the specific drivers, the "silent" hardware can be "active" to implement the assignments. As shown in Fig. 4.2 wireless transmission module drivers are included in peripheral hardware drivers, and provided by the microcontroller manufacturers. This section focuses only on sensor driver development.

The connected sensors are the actual devices for obtaining environmental information. Many commonly used sensors are capable of being encapsulated inside a tiny unit, and controlled by a specific miniature control unit. It is not necessary for the embedded software developers to access the physical layer of the sensors. The control units will take the responsibility of accomplishing the designed functionalities of the sensors and interacting with the external control system.

The embedded software design for the sensor driver development is to enable the wireless sensor node to acquire the sensor data from the intended sensors, or set user instructions to the sensors. Although the working principle of various sensors might be completely different, the sensor interactions (i.e. output and input interface) can be briefly divided into two forms: digital interface and analog interface. Communications through the digital interface are accomplished by using a certain period of high level and low level voltages to express the binary signal of "1" and "0". By combining meaningful "0" and "1" using specific rules, the

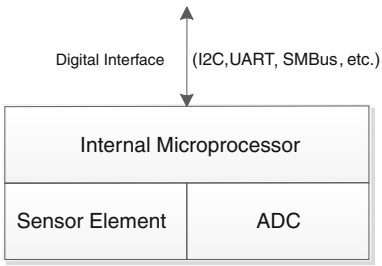


Fig. 4.6 Digital sensor components

developers can send and receive content which is able to be understood by both the sensor control unit (sensor microprocessor) and the outside control system (wireless chip). Figure 4.6 gives a generic structure of digital sensors. An internal ADC (analog-to-digital converter) works with the microprocessor to output digital signal through a digital interface. There are many digital communication standards available for constructing the digital interface (i.e. communication rules), such as the serial communication protocol, the SMBus protocol (System Management Bus, defined by Intel®), the I2C protocol (Inter-Integrated Circuit, defined by the company who previously were Philips Semiconductors, but now are NXP Semiconductors), the Universal Asynchronous Receiver/Transmitter (UART), the 1- Wire interface (defined by Maxim Integrated Products, Inc) and so on.

An analog communication interface is simpler compared with a digital interface. Normally the analog sensors output a corresponding voltage level, which is related to the change in the sensing phenomenon. It can be used by the external control system when it has been converted into the form of a digital signal by using an ADC. Some kinds of analog sensors output consist of a sending a series of pulses, which are correlated with the intensity of the sensing phenomena (voice, light, temperature, etc.) to the external controller. The number of pulses will be sampled by the outside control system for a certain period and then converted into a meaningful value using a predefined formula before being finally presenting to the users.

4.3.1 General Procedure of Sensor Drivers

A sensor driver should provide the outside controllers with the capability of obtaining the sensor readings, and conveying the user's instructions to the sensors. In general, a complete sensor driver design consists of the steps of sensor initialization, sensor parameter setting, sensor data acquisition, and sensor power management (sleep, wait, and standby). Figure 4.7 illustrates a general flow chart for designing sensor drivers.

We will describe these processes in more detail below:

- Sensor Initialization: The first process is "Sensor Initialization", which is responsible for initializing all the default sensor parameters. The stages this involves include: powering on the sensor, setting up the communication interface, and restore the default settings for the sensor. Many sensors store the parameter settings in a connected non-volatile memory, for example, an Electrically Erasable Programmable Read-Only Memory (EEPROM), where the user configurations or specific manufacturer calibrations can be kept safe and are easily accessible for restoration.
- Error Report: During the start up period, the sensors may not response correctly, which means some kinds of errors might have occurred. To protect the outside control system from a faulty sensor, some corresponding error handling process is needed at this stage.

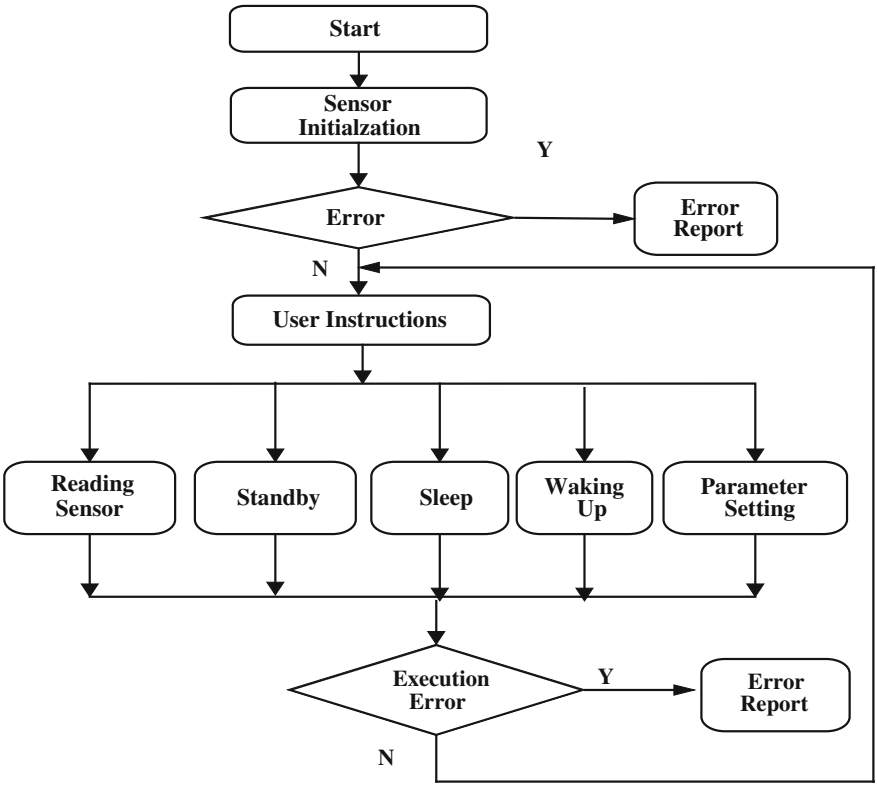


Fig. 4.7 General flow chart of sensor driver

- User Instructions: After successfully initializing the sensors, the driver should be ready to execute the user instructions. Before reaching the end of its working period, the sensor driver can be in one of the five following states: Reading the Sensor, Standby, Sleep, Waking Up, and Parameter Setting. The characteristics and usages of the sensors depend on the application requirements. It is impossible to give a detailed description for all sensor driver design. However, the five stages above are the basic units, which should be included no matter which sensors are used.

  (a) Reading the Sensor: "reading the sensor" is the most important function in sensor driver design. To obtain the sensor data, the outside controller will send a request command to commence the reading procedures. On the completion of the reading procedures, the sensor data will be available as output. Since the nature of the sensing material varies, many of the sensors (temperature sensor, vibration sensors, humidity sensor, etc.) require a certain period of time to perform the sensing task, i.e. a sampling period. There are two processing methods by which driver waits for the sensor

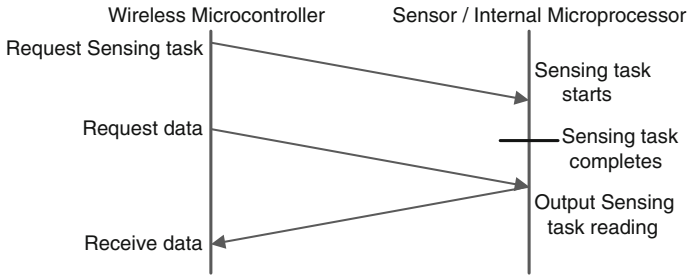


Fig. 4.8 On-request sensor reading

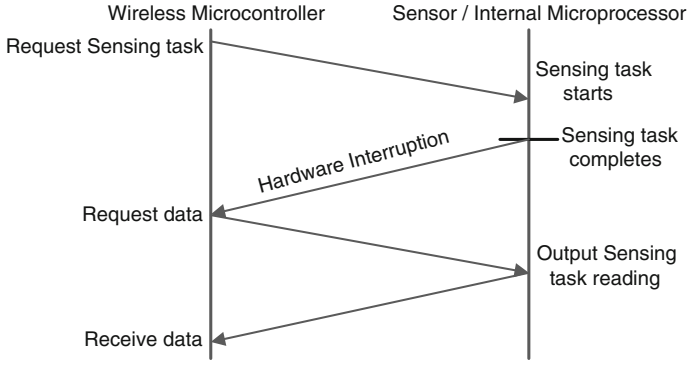


Fig. 4.9 On-request sensor reading with hardware interruption

output: firstly it can exclusively seize the communication interface until the sampling period expires as shown in Fig. 4.8, or, alternatively, using hardware interruption it can leave the communication interface after sending the data request task and then return to the communication interface again after the sampling is finished, as shown in Fig. 4.9. The first method can ensure the driver's programming coherence, which means the function can return with the desired result. Its disadvantage is that as the sensor driver holds the thread of the processor while waiting for the sensor data, rather than giving it back to the system, other normal tasks might be interrupted. This is particularly serious if the processor of the outside control system is also the controller of the wireless communication module. To avoid interposing the control system, the later method can be considered. The sensor driver can ask the outside controller to set up an interruption which will be trigged when the sensor has finished its sampling task and is ready to output the result. Upon receiving the interruption, the outside controller can start to read the sensor. Subsequently, the system will be less likely frozen. The shortcoming of the second method is an increment in the system overheads. As the indication of the completion of the sampling period requires the

controller's assistance, the sensor driver cannot be completely independent from the system. If the selected controller is changed, the sensor driver might also have to be modified.

(b) Standby, Sleep, and Waking Up: The wireless sensor network normally works with extremely limited resource. In particular power management is important for extending the system's lifetime. The actual specification of power management may vary from case to case, as sensors' characteristics are different. Generally, three functions (standby, sleep, and waking up) must always be included in the driver design. The sensors should be kept in the sleep state as long as possible so long as no sensing task is needed. However, under some situations, e.g. high frequency sensor data reading, if the procedure of waking up the sensor from the sleeping mode consumes more energy than the amount consumed in the normal state (some electronic chemistry sensors need a period to rebuild the steady state), it's better to keep the sensor in the standby state, if applicable. Carefully calculating the energy cost is the first step in designing the power management module.

(c) Parameter Setting: If the sensor configuration is adjustable, the sensor driver should allow the applications (i.e. users) to change the parameters when required.

(d) Execution Error: If the execution of the user instructions fails, the design should monitor the returned error of the instruction executions and put it into the corresponding error handling function. These functions are an effective measure for maintaining the system's stability.

4.3.2 Sensor Driver for an Analog Flow Sensor

The FT110 flow sensor (Gems Sensors 2008) is used in this section to illustrate how to design and program a sensor driver for an analog sensor with hardware interruptions. The specification of the flow sensor is listed in Table 4.1.

According to Table 4.1, the FT110 flow sensor generates 2,200 pulses per litre. Hence, the driver of the flow sensor is to use the wireless microcontroller to record the number of pulses generated by the flow sensor, and convert the number into a flow rate. In this application, a capture function from a timer (Timer 0 on a JN5139 chip is selected here) is used to count the pulses. The timer is configured to generate an interruption when it has counted 22 pulses. Every 22 pulses indicate 10 ml of water usages. The

Table 4.1 Specifications of FT110 flow sensor

Mode	FT110
Operating temperature	−20 to 100 °C
Flow rate range	0–15 l/min
Pulse per litre	2,200
Accuracy	±3 %

variable "*u16MillLitre*" is defined to record the number of the interruptions generated by the timer. The final measurement of the flow is recorded in the variable "*u32Litre*". The sensor driver of the flow sensor consists of the following three steps.

Step 1: Enable the timer Timer0 interruption. The constant E_AHI_TIMER_0 represents the timer Timer0. The predefined functions *vAHI_TimerEnable* and *vAHI_TimerClockSelect* are used to initialize the timer.

```
vAHI_TimerEnable(E_AHI_TIMER_0,
  0,
  FALSE,
  TRUE,
  FALSE);
vAHI_TimerClockSelect(E_AHI_TIMER_0, TRUE, FALSE);
```

Step 2: Preset a value for the timer Timer0. When Timer0 counts 22 pulses from the FT110 sensor, it generates an interruption. The predefined function *vAHI_TimerStartRepeat* is used to set the value for a timer to generate an interruption.

```
vAHI_TimerStartRepeat(E_AHI_TIMER_0,
  0x0000,
  22); //number 22 is set for FT110 sensor
```

Step 3: Process the flow sensor reading. When the JN5139 chip detects an interruption generated from Timer 0, it processes the flow sensor reading. The constant E_AHI_DEVICE_TIMER0 indicates that the interruption is received by the JN5139 wireless microcontroller.

```
case E_AHI_DEVICE_TIMER0:
  vProcessFlowMeter();
PRIVATE void vProcessFlowMeter(void)
{
/*water usage reading increase by 1 (Milliliter) when an interruption is
detected. */
```

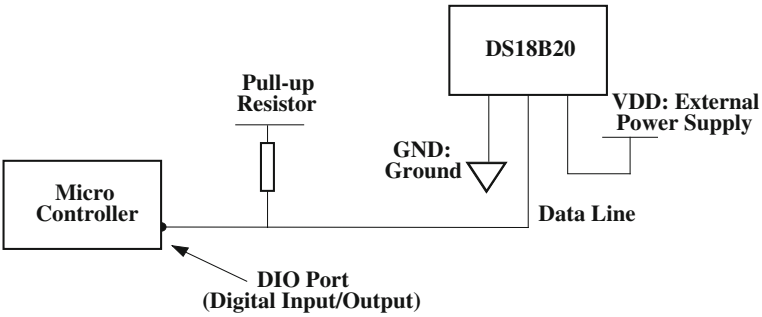


Fig. 4.10 Connection between the DS18B20 sensor and an outside microcontroller

```
/*Every 100 interruptions, u32Liter increase by 1 (liter) and
u16MillLitre = 0*/

  u16MillLitre++;
  if(u16MillLitre == 100)
  {

u32Litre++;
u16MillLitre = 0;
  }
}
```

4.3.3 Sensor Driver for a Digital Temperature Sensor

A Maxim DS18B20 (Maxim 2009) digital temperature sensor is chosen as an example to demonstrate the sensor deriver design for a digital sensor. The DS18B20 sensor is a typical low power consumption digital sensor. Its features of requiring a low power supply (3.0–5.5VDC) and a wide measurement range (−55 to +25 °C) make it suitable for many applications. All operations of the DS18B20 sensor is achieved through its unique 1-Wire ® interface. Figure 4.10 illustrates the connection between the DS18B20 sensor and the outside microcontroller. The temperature sensor communicates with the outside microcontroller (a JN5139 chip is chosen here) through the 1-Wire interface, in which the outside microcontroller is defined as a master device and the DS18B20 sensor is defined as a slave device.

Figure 4.11 shows the block diagram of the DS18B20 sensor. “64-Bit ROM AND 1-Wire PORT” and “SCRATCHPAD” are two memory spaces the sensor driver needs to operate. The 64 bit sensor identification and the 1-Wire interface protocol are permanently stored in the “64-Bit ROM AND 1-wire PORT” and a

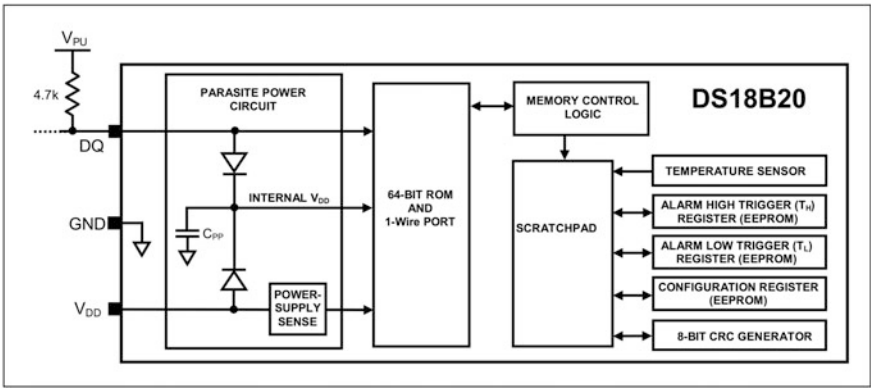


Fig. 4.11 DS18B20 block diagram (Maxim 2009)

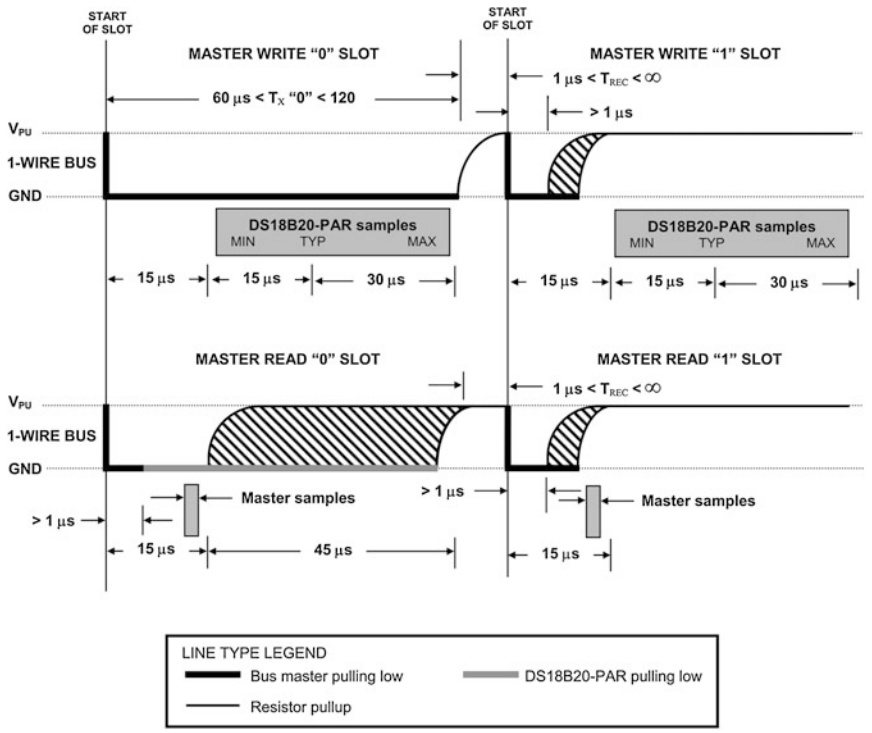


Fig. 4.12 Read/Write time slot timing diagram (Maxim 2009)

memory map for the users to conduct configuration is provided by the scratchpad. The scratchpad consists of 9 bytes, containing various parameters. For example, the first and the second bytes in the scratchpad store the sensor reading, the third

and fourth bytes are the upper and lower thresholds of measurements, the fifth to seventh byte are reserved for other use such as setting-up the resolution, and the last byte is the CRC value for validation. As shown in Fig. 4.11 the scratchpad is linked with the temperature sensor, the alarm registers, the configuration register and the CRC generator.

As shown in Fig. 4.7, the core operations of a sensor driver are sensor initialization, parameter setting, reading from the sensor, and writing commands to the sensor such as moving the sensor into a standby, sleep, or wake-up mode. Figure 4.12 illustrates the "Read/Write Time Slot Timing" of the DS18B20 sensor. From left to right and from top to bottom the master device (a JN5139 in this case) writing logic '0', '1' and reading logic '0' and '1' are shown. The sensor changes its logic status to meet the operation requirements of the master device.

In detail, when the microcontroller is to issue a writing operation, it should:

1. Poll the 1-Wire bus low
2. If the microcontroller is to write a logic "1" to the temperature sensor, the 1-Wire bus must be released within 15 μs. Then the 1-Wire bus will be pulled high by the pull-up resistor. The status of "high" should last for at least 15 μs or at most 45 μs.
3. If the JN39 chip is to write logic "0" to the temperature sensor, the 1-Wire bust must be pulled low for the minimum duration of 60 μs.

If the microcontroller is to read from the temperature sensor, it should:

1. Pull down the 1-Wire bus for at least 1 μs and then releases the 1-Wire bus.
2. The DS18B20 sensor starts to transmit a logical "1" or "0" on the 1-Wire bus to the microcontroller. The DS18B20 sensor transmits a "1" by leaving the bus high and transmits a "0" by pulling the bus low.
3. The temperature sensor will release the 1-Wire bus at the end of the reading slot, and the bus will be pulled back to a high state. The output data from the temperature sensor is valid for 15 μs after the falling edge that initiated the read time slot. Hence, the microcontroller must release the 1-Wire bus and start to sample the bus within 15 μs from the start of read time slot.

Before writing or reading the temperature sensor, the initialization of the temperature sensor should be carried out.

```
PUBLIC bool_t Init_DS18B20(void)
{
//adding initialization here
}
```

Normally, each DS18B20 sensor has its unique 64-bit identification code stored in the "64-Bit ROM AND 1-Wire PORT". The sensor ID is used to distinguish an individual sensor when multiple sensors are connected to the microcontroller through the 1-Wire bus interface. The function *u64ReadSensorID*() is called and the return value is the sensor ID, a 64-bit value, i.e. *uint64*.

```
PUBLIC uint64 u64ReadSensorID(void)
{
//Reading the sensor ID from the 64_Bit Rom and 1-Wire Port by sending a
command COMMAND_READSENSORID to the sensor
}
```

The second configuration parameter is the temperature resolution. According to the manufacturer datasheet (Maxim 2009), the conversion resolution is programmable from 9 to 12 bits. The higher conversion resolution takes a longer conversion time. The following two functions are designed to complete the temperature conversion resolution adjustments: *u8ReadConResolution*() and *u8SetConResolution*(). The former is used to read the conversion resolution from the sensor and the latter to set a new conversion resolution. The return value type *uint8* is an 8-bit value.

```
PUBLIC uint8 u8ReadConResolution(void)
{
//Reading the conversion resolution by sending a command COM-
MAND_READSCRATCHPAD to the scratchpad
}
PUBLIC uint8 u8SetConResolution(unit8 u8Resolution)
{
//Setting the new conversion resolution by sending a command COM-
MAND_WRITESCRATCHOAD to the scratchpad
}
```

After initializing and configuring the temperature sensor, the temperature reading is achieved by using two function *vStartConversion*() and *u16ReadTemperature*(). The DS18B20 sensor is required to start the temperature conversion. Once the function is called, the system must start a timer, which will expire after a certain period. The length of the period is determined by the resolution of the conversion.

```
PUBLIC void vStartConversion(void)
{
//Starting temperature conversion by sending a pre-defined command
COMMAD_STARTCONVERSION to the scratchpad
}
```

The temperature reading is obtained by calling the function *u16Read Temperature*(). The sensor reading is saved in the first and second bytes of the scratchpad before being read out by the microcontroller. The upper and lower ranges of the reading are set in the third and fourth bytes of the scratchpad. The reading is validated by the CRC generator by calling the function *u8CRCCaculation*().

```
PUBLIC uint16 u16ReadTemperature(void)
{
//Obtaining the sensor reading by sending a pre-defined command COM-
MAD_READSCRATCHPAD to the scratchpad
}
PRIVATE uint8 u8CRCCaculation(unit8* content, uint8 u8Length)
{
//validating the return value from the function u16ReadTemperature. The
first parameter is the reading, and the second is the length
}
```

In summary, any digital sensor driver should follow the seven steps below.

Step 1. Initialize the sensor.
Step 2. Read the unique sensor ID from the sensor.
Step 3. Read the conversion resolution from the sensor.
Step 4. Set the conversion resolution to the sensor.
Step 5. Start the reading conversion.
Step 6. Read the sensor reading.
Step 7. CRC validation.

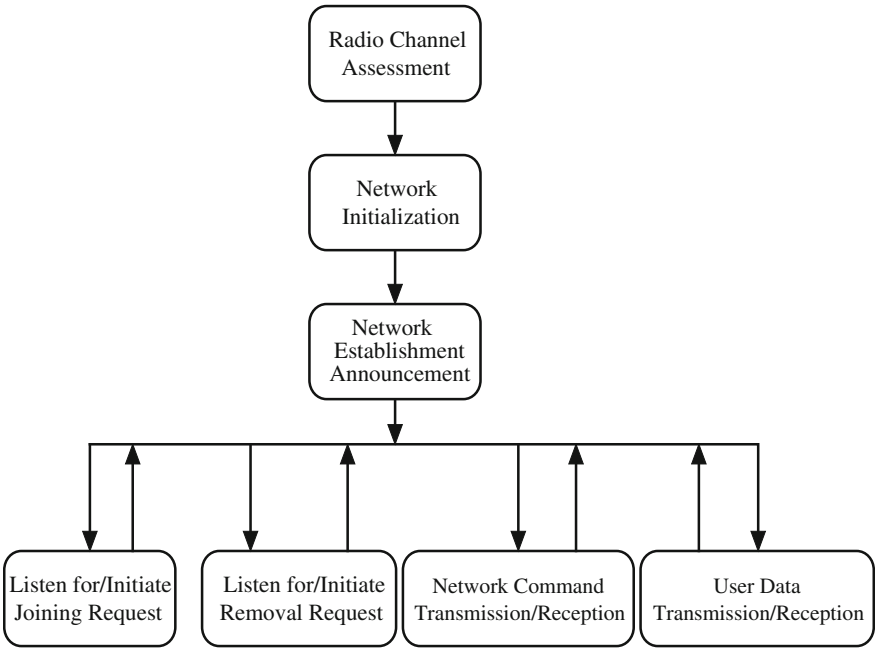


Fig. 4.13 Procedures for starting a wireless sensor network

4.4 Implementing a WSN with IEEE 802.15.4

Figure 2.8 in Chap. 2 shows the general procedure for establishing a wireless sensor network, which consists of seven stages: (1) radio channel assessment, (2) network initialization, (3) network establishment announcement, (4) listen for and initiate joining request, (5) listen for initiate removal request, (6) network command transmission and reception and (7) user data transmission and reception. Section 2.3 gives the detail of each stage in the procedure. This section presents the implementation of this procedure from the viewpoint of embedded software design. We reproduce Fig. 2.8 here as Fig. 4.13 for the sake of readers' convenience.

The procedure shown in Fig. 4.13 is equally applied to the implementation of adopting the ZigBee stack or an IEEE 802.15.4 stack. The processes of constructing a WNS are similar in either case. The difference is that the ZigBee stack provides more convenient management functions for rapid application development such as the multi-hop routing protocol etc. Because the ZigBee stack is built on top of an IEEE 802.15.4 stack, this section will only provide the implementation of the procedure for a ZigBee stack, by adopting the syntax of the Jennic IEEE 802.15.4 stack (Jennic 2008a).

Two main function calls from the network (NWK) layer to the MAC layer which include the MAC layer management entity (MLME) and the MAC common part sublayer (MCPS) are implemented as follows:

```
void vAppApiMlmeRequest(
    MAC_MlmeReqRsp_s *psMlmeReqRsp,
    MAC_MlmeSyncCfm_s *psMlmeSyncCfm);
```

The *vAppApiMlmeRequest* routine is used to pass a MLME requests from the NWK layer or the Application to the MAC. The *psMlmeReqRsp* parameter is a pointer to a structure holding the request to theMLME. The *psMlmeSyncCfm* parameter is a pointer to a structure holding the results of the MLME request.

```
void vAppApiMcpsRequest(
    MAC_McpsReqRsp_s *psMcpsReqRsp,
    MAC_McpsSyncCfm_s *psMcpsSyncCfm);
```

The *vAppApiMcpsRequest* routine is used to pass MCPS requests from the NWK layer or the Application to the MAC. The *psMcpsReqRsp* parameter is a

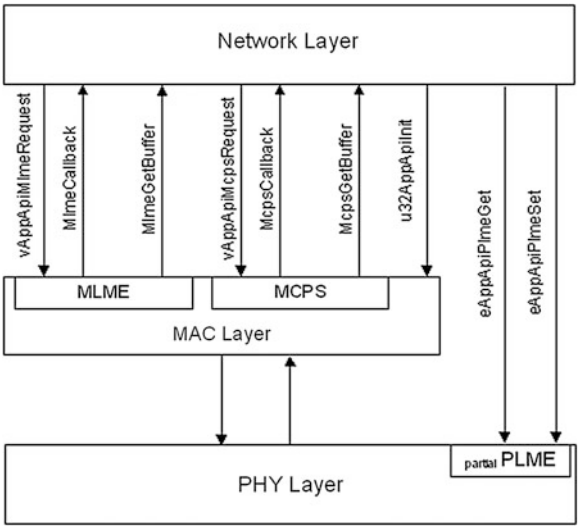


Fig. 4.14 Overview of the interface between NWK layer, MAC layer, and PHY layer (Jennic 2008a)

pointer to a structure holding the request to the MCPS. The *psMcpsSyncCfm* parameter is a pointer to a structure holding the results of the MCPS request.

There are a number of other functions required in the establishment of an IEEE 802.15.4 WSNs. Figure 4.14 provides an overview of the functions (call and callback) making up the interface between the network layers and the MAC layer. These call functions allow requests from the Network layer to the MAC layer and then the callback functions allows the MAC layer to request the Network layer to allocate buffer space and to pass information back to the Network layer.

- Radio Channel Assessment

Radio channel assessment is the first task to be conducted when an IEEE 802.15.4 network PAN coordinator starts a WSN. The PAN coordinator should make sure that the desired radio channel is available for use and there is no existing network conflict.

The following code segment is used to submit an energy scan request by calling the function *vAppApiMlmeRequest*(). There are two parameters required by this function. The first parameter is a pointer to the data structure *MAC_MlmeReqRsp_s* which holds the request to MLME such as, scan type, scan duration and scan channel etc. The second parameter is a pointer to the data structure *MAC_MlmeSyncCfm_s,* which holds the results of the MLME request.

```
#define SCAN_CHANNELS 0x07FFF800UL
#define ENERGY_SCAN_DURATION 3
//Structures to hold the parameters for scan request and scan response
MAC_MlmeReqRsp_s sMlmeReqRsp;
MAC_MlmeSyncCfm_s sMlmeSyncCfm;
//The request type is scan
sMlmeReqRsp.u8Type = MAC_MLME_REQ_SCAN;
//The size of the request
sMlmeReqRsp.u8ParamLength = sizeof(MAC_MlmeReqStart_s);
//The scan type is energy scan
sMlmeReqRsp.uParam.sReqScan.u8ScanType = MAC_MLME_SCAN_
TYPE_ENERGY_DETECT;
//Set scan channels
sMlmeReqRsp.uParam.sReqScan.u32ScanChannels = SCAN_CHANNELS;
//Set the scan duration
sMlmeReqRsp.uParam.sReqScan.u8ScanDuration = ENERGY_SCAN_z
zDURATION;
//Submit energy scan request
vAppApiMlmeRequest(&sMlmeReqRsp, &sMlmeSyncCfm);
```

The active scan uses a similar code structure but with the scan type MAC_MLME_SCAN_TYPE_ACTIVE. After the completion of the energy and active scans, the PAN coordinator should know whether or not a specified channel and network identifier are available for use.

- Network Initialization

If the radio channel assessment has been successfully, the PAN coordinator can start the network initialization procedure, which includes the channel number, network ID, beacon and superframe orders etc.

```
//Structures for holding the request information and response
MAC_MlmeReqRsp_s sMlmeReqRsp;
MAC_MlmeSyncCfm_s sMlmeSyncCfm;
//The request type is starting network
sMlmeReqRsp.u8Type = MAC_MLME_REQ_START;
//The size of the request
sMlmeReqRsp.u8ParamLength = sizeof(MAC_MlmeReqStart_s);
//The network ID. Make sure it does not conflict with other PANs in the
vicinity
sMlmeReqRsp.uParam.sReqStart.u16PanId = 0x1234;
//Define the working channel
sMlmeReqRsp.uParam.sReqStart.u8Channel = CHANNEL_CLEAN;
//Define the beacon order and superframe order. The duty cycle is 50 % in
this case
sMlmeReqRsp.uParam.sReqStart.u8BeaconOrder = 0x03;
sMlmeReqRsp.uParam.sReqStart.u8SuperframeOrder = 0x02;
//The network is started by the PAN coordinator
sMlmeReqRsp.uParam.sReqStart.u8PanCoordinator = TRUE;
```

- Network Establishment Announcement

The procedure for announcing the establishment of the network is determined by the network protocols used. The actual announcement of the existence is usually implemented by sending out regular beacon signals. If the protocol does not support regular beacon signal emission, it should be capable of responding to any valid request that is sent by those devices executing radio channel assessments.

```
//Regularly sending the beacon signals
vAppApiMlmeRequest(&sMlmeReqRsp, &sMlmeSyncCfm);
```

- Listen for/Initiate Joining Request and Listen for and Initiate Removal Request

As well as other wireless networks, the wireless sensor network should be able to extend its coverage or expand its capacity by adopting new devices, or remove existing networked devices when requested to do so on the receipt of valid instructions. Its network stack must have a predefined mechanism to cope with such network changes.

To avoid creating duplicated entries, a simple mechanism is provided in the IEEE 802.15.4 stack. By checking the unique 64-bit extended address of the requesting device, the PAN coordinator can determine if this device already exists in the local list. If it is, it grants the request with the same 16-bit network address. If it is not in the local list, and if there is space available for adopting any new network devices, it grants the request with a new 16-bit network address formed by adding 1 to the previously allocated 16-network address. In the following code segment, the16-bit network address is saved in the variable *u16ShortAdr*. The joining request and removal request are implemented by submitting a response with the request type as MAC_MLME_RSP_ASSOCIATE and MAC_MLME_RSP_DISASSOCIATE respectively. MAX_END_DEVICES is the maximum number of devices the PAN coordinator is able to accept. The current number of end devices is saved in the field *numEndDevices*.

```
MAC_MlmeReqRsp_s sMlmeReqRsp;
MAC_MlmeSyncCfm_s sMlmeSyncCfm;
//The address of end devices starts from 0x0000
#define END_DEVICE_START_ADR 0x0000
uint16 u16ShortAdr = 0;
//If local space is still available
if (PANCoordinator.numEndDevices < MAX_END_DEVICES)
//Generate a new 16-bit network address
u16ShortAdr = END_DEVICE_START_ADR
+ PANCoordinator. numEndDevices;
//Create the association response. The request type is associate response
sMlmeReqRsp.u8Type = MAC_MLME_RSP_ASSOCIATE;
//Length of response
sMlmeReqRsp.u8ParamLength = sizeof(MAC_MlmeRspAssociate_s);
```

```
//Store the 16-bit address of the new device
sMlmeReqRsp.uParam.sRspAssociate.u16AssocShortAddr = u16ShortAdr;
//Submit the associate response
vAppApiMlmeRequest(&sMlmeReqRsp, &sMlmeSyncCfm);
```

Unlike the joining request, the removal request is initiated by a network device. The request type is MAC_MLME_RSP_ASSOCIATE, and the 16-bit network address will be the coordinator short address denoted as *CoordShortAddr*.

```
/Structures for holding the disassociate request
MAC_MlmeReqRsp_s sMlmeReqRsp;
MAC_MlmeSyncCfm_s sMlmeSyncCfm;
//The request type is disassociate request
sMlmeReqRsp.u8Type = MAC_MLME_REQ_DISASSOCIATE;
//Length of response
sMlmeReqRsp.u8ParamLength = sizeof(MAC_MlmeReqDisassociate_s);
//the 16-bit address disassociated with
sMlmeReqRsp.uParam.sReqDisassociate.sAddr.uAddr.u16Short
= CoordShortAddr;
//Submit the disassociate request
vAppApiMlmeRequest(&sMlmeReqRsp, &sMlmeSyncCfm);
```

- Network Command Transmission and Reception

Normally the network commands are kept invisible to the users, but occasionally they need user intervention. A robust network is supposed to be capable of implementing the necessary algorithm to deal with the various situations occurring inside the network without any user intervention. However, the application software is still required to maintain administrative capability on the network management. For example, when a network system starts to increase the security level for adopting new devices, the details of any requesting device need to be reviewed by the upper layer management system, rather than automatically being decided by the stack itself. Network command transmission and reception is also conducted by submitting the corresponding request.

```
//Submitting the request when required
vAppApiMlmeRequest(&sMlmeReqRsp, &sMlmeSyncCfm);
```

- User Data Transmission/Reception

As described in Sect. 2.3.7 user data transmission and reception can only be indirect for beacon-enabled networks using a slotted CSMA-CA, but can be direct or indirect for nonbecon-enabled networks using an unslotted CSMA-CA. An indirect transmission from a coordinator to a network device is shown in the following code segment. When the coordinator has a packet to transmit to a network device, it stores the data in the local buffer (the field *sFrame.au8Sdu* in the following example), and link the "address pending list" (*pu8Payload* here) with the packet information. Finally it submits the data transmission/reception request by calling the function *vAppApiMcpsRequest*, the network layer or application to MAC common part sublayer (MCPS).

```
//Structures for holding the data transmission request
MAC_McpsReqRsp_s sMcpsReqRsp;
MAC_McpsSyncCfm_s sMcpsSyncCfm;
//a pointer to the outgoing packet
uint8 *pu8Payload;
//The request type is data request
sMcpsReqRsp.u8Type = MAC_MCPS_REQ_DATA;
sMcpsReqRsp.u8ParamLength = sizeof(MAC_McpsReqData_s);
//Generate an id for the outgoing data packet
sMcpsReqRsp.uParam.sReqData.u8Handle = u8CurrentTxHandle;
//Prepare the coordinator ID (PAN-ID) and 16-bit short address
COORDINATOR_ADR as//the source address
sMcpsReqRsp.uParam.sReqData.sFrame.sSrcAddr.u16PanId = PAN_ID;
sMcpsReqRsp.uParam.sReqData.sFrame.sSrcAddr.uAddr.u16Short =
COORDINATOR_ADR;
//Prepare the destination 16-bit short address for transmission
sMcpsReqRsp.uParam.sReqData.sFrame.sDstAddr.u16PanId = PAN_ID;
sMcpsReqRsp.uParam.sReqData.sFrame.sDstAddr.uAddr.u16Short =
u16DestAdr;
//Use indirect transmission for a coordinator to generate data request. The
data will be stored in the local buffer until being fetched by the network
```

```
device, rather than directly//transmitting to the network device. Acknowl-
edgement is required.
sMcpsReqRsp.uParam.sReqData.sFrame.u8
TxOptions = (MAC_TX_OPTION_ACK|MAC_TX_OPTION_INDIRECT);
//Link the address of the pu8Payload to the corresponding component in the
outgoing//packet.
pu8Payload = sMcpsReqRsp.uParam.sReqData.sFrame.au8Sdu;
//Put the data into the packet
…….
//Finally, store the payload length in the structure
sMcpsReqRsp.uParam.sReqData.sFrame.u8SduLength = EFFECTIVE_
PAYLOAD;
//Submit the data transmission request
vAppApiMcpsRequest(&sMcpsReqRsp, &sMcpsSyncCfm);
```

4.5 Bridging WSNs with an External Public Network

WSNs are designed to collect data from sensor fields and pass the collected data to a local or remote data processing station via their gateway, which would be a named sink node for ZigBee network and a border router for 6LowPAN network. The local or remote data processing station is usually operating in a public network

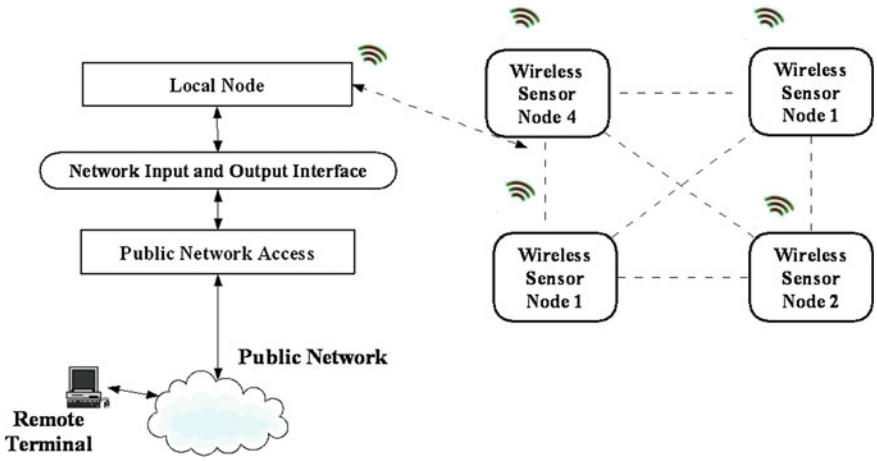


Fig. 4.15 Connection between a wireless sensor network and a public network

Table 4.2 Message format for network input and output interface

Byte 1	Command length (1 – N)
Byte 2	Command type
Byte 3	Parameter 1
Byte 4	Parameter 2
…	…
Byte N	CRC (Cyclic Redundancy Check)

with a different communication protocol. Frequently, a WSN is linked with a WiFi system and then via the WiFi system the WSN is linked with the Internet. Figure 4.15 illustrates the connection between a 4-node wireless sensor network and an external public network.

In Fig. 4.15, the local node can be a sink node or a border router. All the collected information is forwarded from the individual sensor nodes to the local node. Then the information is passed to the public network access where there is a specially designed protocol to handle the data delivery to the remote terminal through the public network. The instructions sent from the remote terminal to the wireless sensor network should follow the reverse path. Since the public network and the wireless sensor network normally do not use the same communication protocol, a network input and output interface is needed to perform translator functionality between these two systems. In most cases particularly designed hardware is used to implement the network input and output interface. For example, the connection between a WiFi and a ZigBee WSNs is implemented by wiring a Digi connect Me module (Digi International 2010) with a Jennic ZigBee chip JN5139 through a high-speed serial interface (UART) in a home automation system (Gill et al. 2009).

To the embedded software designer, it is particularly important to implement some kind of error checking mechanisms to detect errors, which occur during the data transmission and reception. Table 4.2 illustrates a simple way to define the format of interface message at both sides.

In Table 4.2, the message format consists of a serial of bytes. The first byte contains the length of the whole message. On reception of the first byte, the receiver can determine how many bytes it should expect in the current transmission. The message body contains the regular information, including the command type and parameters. The last element of the message format should contain the CRC (cyclic redundancy check) value of all the message elements except for the CRC field itself. When the receiver has obtained the complete message, it can calculate the CRC value for the first $(N - 1)$ bytes, and compare it with the last byte. If it matches, the reception was successful. Otherwise, the receiver will ask for the sender to retransmit, or perform other predefined algorithms.

4.6 Summary

Embedded software development is the core task in WSN design and implementation. This chapter starts with the embedded software design process, and then introduces the embedded software architecture. ZigBee application design and Contiki 6LowPAN application design are given to illustrate the embedded software structure for WSNs. Sensor driver design is emphasized in the chapter and illustrated by the derived design for an analog flow sensor and a digital temperature sensor. The implementation of the procedure of establishing WSNs is demonstrated by code segments using the IEEE 802.15.4 stack. This chapter is concerned with the understanding of the design process and the basic coding of embedded software for establishing a WSN. The detail syntax of the coding is ignored as it varies from system to system.

References

Contiki Wiki: Processes. http://www.sics.se/contiki/wiki/index.php/Processes#Autostarting_Processes (2012)

Digi International: http://www.digi.com/pdf/prd_ds_digiconnectfamily_usersguide.pdf (2010)

Gems Sensor: FT110 Flow sensor. http://docs-europe.electrocomponents.com/webdocs/023e/0900766b8023e8ed.pdf (2008)

Gill, K., Yang, S.H., Yao, F., Lu, X.: A Zigbee-based home automation system. IEEE Trans. Consum. Electron. **55**(2), 422–430 (2009)

Jennic: Jennic 802.15.4 Stack API Reference Manual. http://www.jennic.com/files/support_files/JN-RM-2002-802.15.4-Stack-API-1v8.pdf (2008a)

Jennic: ZigBeeStackUserGuide. http://www.jennic.com/files/support_files/JN-UG-3017-ZigBeeStackUserGuide-1v6.pdf (2008b)

Jennic: IEEE 802.15.4 Application Development Reference Manual. http://www.jennic.com/files/support_files/JN-RM-2024-IEEE802.15.4-App-Dev-2v0.pdf (2010)

Labrosse, J., Ganssle, J., Noergaard, T., Oshana, R., Walls, C., Curtis, K., Andrews, J., Katz, D.J., Bentile, R., Hyder, K., Perrin, B.: Embedded Software. Elsevier, Amsterdam (2008)

Laplante, P.A.: Real-Time Systems Design and Analysis. Wiley, New York (2004)

Maxim: DS18B20 Programmable Resolution 1-Wire Digital Thermometer. http://pdfserv.maxim-ic.com/en/ds/DS18B20.pdf (2009)

Chapter 5
Routing Technologies in WSNs

Keywords Routing protocol · AODV · Cluster-tree · Energy aware routing

5.1 Introduction

A routing protocol is software that sits in the network layer and is responsible for deciding on which output route an incoming packet should be transmitted. In other words, it is an algorithm for finding a data transmission path from a source node to a destination node. The destination node is normally called a sink node or a base station in WSNs. It could be some distance away from the source node or even out of the transmission range of that node. Therefore, the data may have to travel via multiple hops before reaching the sink node. Figure 5.1 shows a transmission path from a sensor to a sink node. However, due to the unique features and constrains of WSNs, the routing protocols developed for wired networks and other wireless networks such as MANET are often not suitable for WSNs. The typical features and constrains of WSN routing are described as follows (Al-Karaki and Kamal 2004):

- One of the main objectives of the routing protocol in WSNs is preserving energy and reducing the power consumption, while the other networks routing protocols are designed to achieve high Quality of Service (QoS) during data transferring.
- The wireless sensor nodes have many limitations such as limited power supply, limited memory size, limited computation capability, and limited bandwidth for the wireless channels between the wireless sensors.
- A WSN may contain a large number of sensor nodes. Consequently, using a global identification address to access each individual node may be infeasible.
- A WSN may have different applications requirements. Hence, the design of the WSN should be application-specific.
- A routing protocol should eliminate the redundancy of the sensed data, which arises when many wireless sensor nodes sense a same environmental

S.-H. Yang, *Wireless Sensor Networks*, Signals and Communication Technology,
DOI: 10.1007/978-1-4471-5505-8_5,

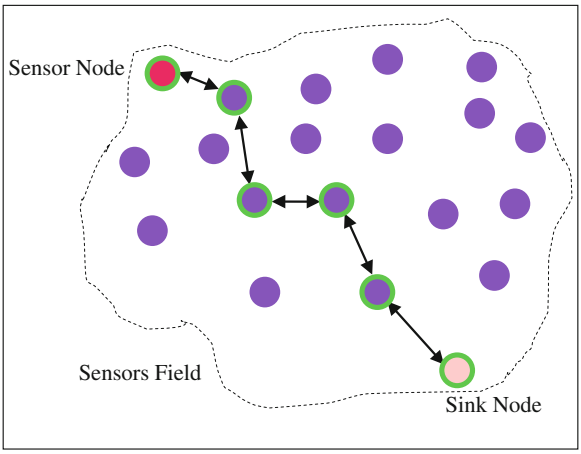


Fig. 5.1 Communication path from a sensor to a sink node

phenomena at the same time. Data aggregation is required to reduce the redundancy, including duplicate suppression, data fusion et al.

- The sensor nodes in WSNs are more prone to errors or failure, due to their low cost and battery driven. Therefore, the routing protocol should function effectively even when there are node failures in the networks. This fault tolerance feature requires that the routing protocol should have the ability to overcome any failure in the network by discovering and maintaining a new route to transmit data.

5.2 Classification of Routing Protocols in WSNs

Routing protocols can generally be classified as *proactive* or *reactive* depending on how the route is determined. Proactive routing protocols determine the routes before they are needed and update the routes when the network topology changes. When a data-sending request is made, the route can be found from the available route table and can be adopted without any further computation. Therefore, there is no additional latency for data delivery to be added. But proactive routing protocols are not appropriate for ad hoc networks where the network topology changes constantly. Reactive routing protocols, on the other hand, only invoke a route discovery procedure on demand. The reactive routing protocols are suitable for dynamic networks. But the time for determining a route can be significant and it may lead to increased latency for data delivery.

There are many other ways to classify routing protocols based on different criteria. Figure 5.2 gives another routing protocol classification, where all routing protocols are classified as network structure based or protocol operation based. In term of network structure, there are three sub-categories: Flat, Hierarchical, and Location-based routing protocols. In term of protocol operations, there are five

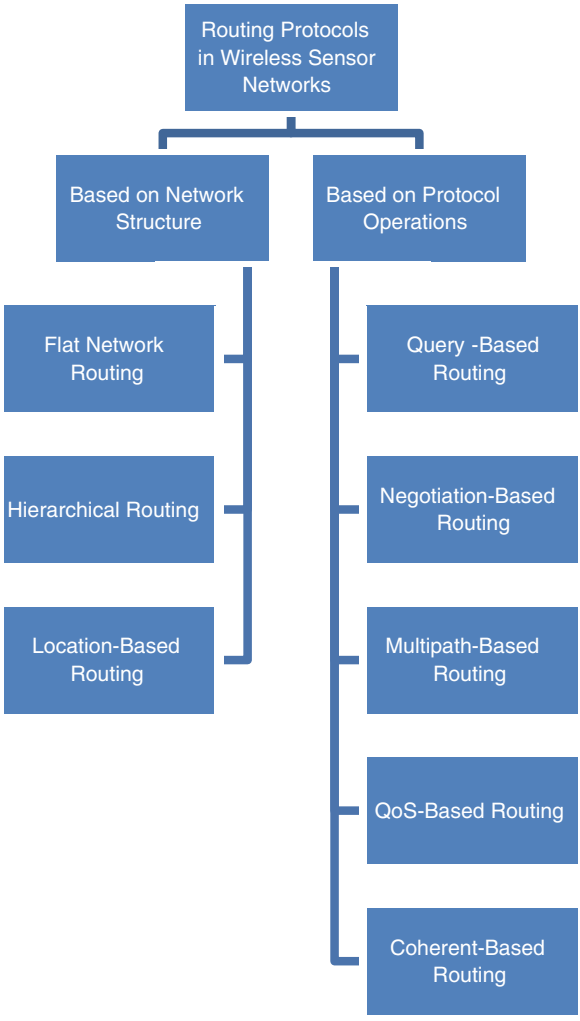


Fig. 5.2 Classification of routing protocols in WSNs

sub-categories: Query-based, Negotiation-based, Multipath-based, Quality of Service (QoS)-based, and Coherent-based routing (Vidhyapriya and Vanathi 2007; Akkaya and Younis 2005). These categories and sub-categories are not mutually exclusive since some routing protocols could be classified under more than one category and sub-category. This section briefly reviews only the routing protocols based on network structure.

The network structure can be classified as flat, hierarchical or location-based. In the flat network, all the sensor nodes have the same level of functionality and responsibilities. They forward the data to their neighbour nodes without any regards to the network topology. On the other hand, sensor nodes in a hierarchical network have different roles to play, and they are logically located at different levels. Many

hierarchical networks are divided into different clusters and each cluster designates a cluster head to aggregate and relay inter-cluster traffic. The location-based routing protocols depend on the physical location of the sensor nodes (Vidhyapriya and Vanathi 2007; Jolly and Latifi 2006).

5.2.1 Flat Routing Protocols

Flat routing protocols use data centric routing protocols for transferring data, where there is a base station responsible for sending requests and queries to other nodes and waiting for their responses. Data elimination and negotiation can be used to save energy in the network. A route discovery process can be initiated by flooding or broadcasting data to all the neighbour nodes without paying attention to any updates, occurring in the topology. This section will discuss the most popular flat routing protocols.

5.2.1.1 Flooding Protocol

The flooding protocol is the most basic flat routing protocol and can easily be implemented over WSNs, because there is no need for any complex algorithm programming. The flooding protocol simply broadcasts data to all the neighbour nodes without considering the topology or the structure of the network (Jolly and Latifi 2006). Then the data can be delivered to the destination node by repeating the same process of broadcasting as shown in Fig. 5.3. Although, this protocol is simple and easy to implement, it has some critical problems. One of these problems is the generation of a large number of duplicate messages by many nodes. Another problem is called implosion. The implosion occurs when a certain node

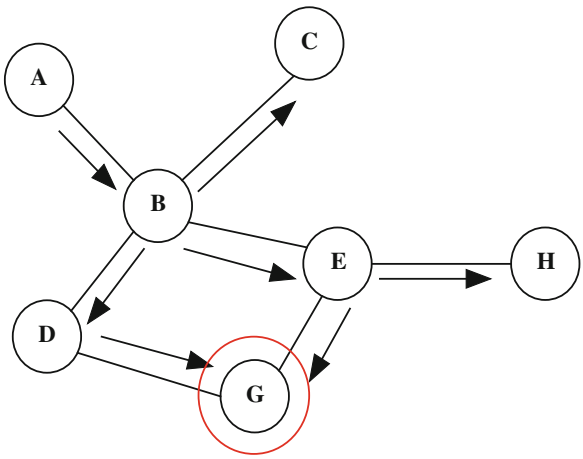


Fig. 5.3 Flooding protocol with implosion problem

receives the same data twice, because each node is sending the received data to its neighbours without knowing whether the neighbour nodes have received these data before or not. Node G in Fig. 5.3 shows the implosion problem.

5.2.1.2 Sensor Protocol for Information via Negotiation

Sensor Protocol for Information via Negotiation (SPIN) is another flat routing protocol. SPIN is an updated version of the flooding protocol. SPIN adds a negotiation system to the protocol. Instead of initially sending data to all neighbour nodes, SPIN first asks for any node which has interest in the data and then send that data only to those who have expressed an interest. There are three types of packets, data advertisement (ADV), data request (REQ), and data (DATA) packets.

A sensor node, which has data, sends an advertisement packet (ADV) to all its neighbour nodes. This ADV includes information about the sensed data. If one of the nodes, which received this ADV, has already received these data before, it will ignore the ADV packet. Otherwise, it will send the request packet (REQ) back to the source node. Finally, the source node will send the data only to these nodes by sending a data packet (DATA). This process is repeated until the data packet is received by the destination. Figure 5.4 illustrates the SPIN procedure, where Fig. 5.4a–c show the three steps occurring between node A and node B, and Fig. 5.4d–f show the transmission between node B and the rest of the nodes. This routing protocol is an attempt to solve the problems of duplicate data packets and the implosion problem. There are many enhanced versions of the SPIN protocol (Jolly and Latifi 2006).

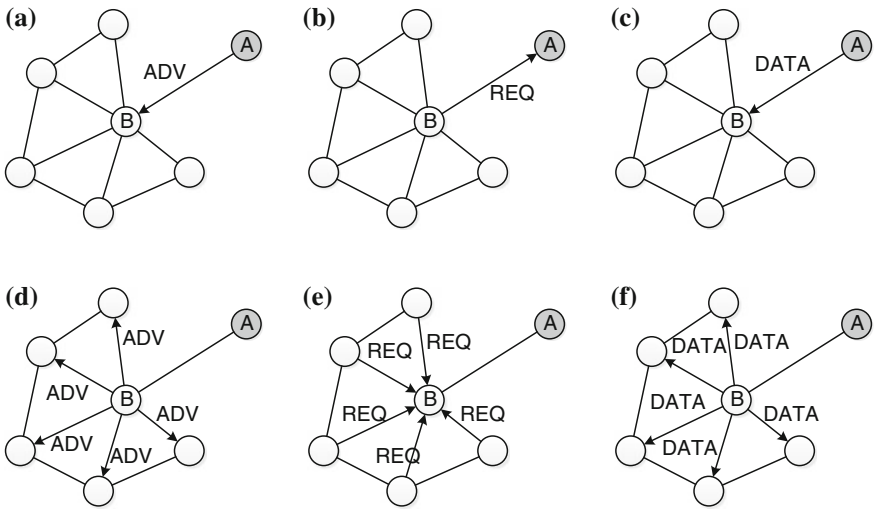


Fig. 5.4 SPIN procedure. **a** ADV is sent from A to B. **b** REQ is sent from B to A. **c** DATA is sent from A to B. **d** ADV is sent from B to the rest. **e** REQ is sent from the rest to B. **f** DATA is sent from B to the rest

Lower energy consumption can be achieved by SPIN than by flooding. However, the data distribution rates are similar. SPIN does not use the distance information between the neighbours to reduce the energy consumption. SPIN's negotiation system reduces the redundant data produced by half. Sometimes the destination nodes are located at some distance from the source node, and the nodes nearer to the source node may not be interested in these data. Therefore, sending an advertisement packet (ADV) cannot guarantee the delivery of the data to distant interested nodes in WSNs (Al-Karaki and Kamal 2004).

5.2.1.3 Directed Diffusion Protocol

Directed Diffusion is another data centric routing protocol, which is used in the flat network architecture. In the Directed Diffusion routing protocol, the process of collecting data is initialized by a sink node or a base station. This process happens in three Steps as follows (Al-Karaki and Kamal 2004; Jolly and Latifi 2006):

- Step 1: the sink node broadcasts an interest packet to all the neighbour nodes, and these neighbours will broadcast this interest packet to all their neighbours until the interest message reaches the source node that has this type of data. The interest message includes a gradient value, which includes attributes value and direction. Step 1 is shown in Fig. 5.5a.
- Step 2: the source node, which has the requested data, sends the data packet to the sink node using multi paths depending on the gradient. Step 2 is shown in Fig. 5.5b.
- Step 3: the best paths are reinforced by the sink node as shown in Fig. 5.5c. Selecting the best path, based on the gradient value, is dependant on the application; for example, some applications require the shortest path, and other applications, the lowest energy consuming path.

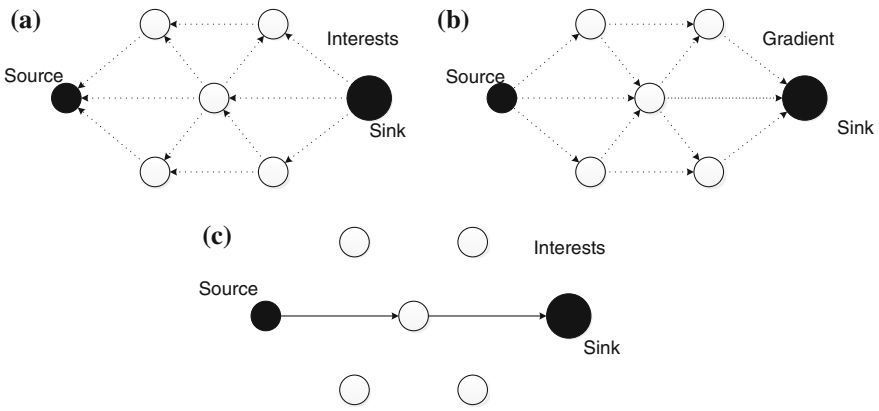


Fig. 5.5 Directed diffusion protocol. **a** Interest is sent from Sink to Source. **b** Data is sent from Source to Sink. **c** Best path from Source to Sink

Directed Diffusion is different from the SPIN or the flooding routing protocol. Data request packet in the Directed Diffusion are always sent from the sink node to the wireless sensor nodes, while in SPIN the wireless sensor nodes advertise that they have data to send, and allows any interested nodes to request it. On the other hand, all the transactions in the Directed Diffusion are neighbour-to-neighbour communication, and all the nodes have the ability to carry out data aggregation and caching. The Directed Diffusion routing protocol does not require a certain network topology and may be unsuitable for applications that need continuous data delivery.

5.2.2 Hierarchical Routings Protocols

Hierarchical routing was originally proposed as a method to route data in wired networks. However, it is also suitable for routing data in wireless networks with some enhancement related to network scalability and the efficiency of communication. The main concept of the hierarchical routing protocols depends on dividing the wireless sensor nodes into more than one level. Most hierarchical routing protocols consist of two routing layers, the first one is responsible for selecting the cluster-heads, and the second is related to routing decisions. For example, hierarchical routing protocols, that need to achieve very low power consumption, can divide the sensor nodes depending on their energy level. The nodes with a high energy level can be assigned to process and transmit data, while the nodes with a lower energy level can be assigned only to sense events. The formation of clusters within the network nodes can improve the efficiency and the scalability of the sensor nodes. There are many hierarchical routing protocols. This section presents only some of the most commonly used protocols.

5.2.2.1 Low Energy Adaptive Clustering Hierarchy

Low Energy Adaptive Clustering Hierarchy (LEACH) concentrates on saving energy and reducing the communication power consumption. In LEACH, a few wireless sensor nodes are selected randomly to act as cluster-heads. By repeating this cluster-head selection process, the wireless sensor nodes will share the energy consumption. If the cluster-heads are fixed, then they will die quickly as they consume more energy than ordinary nodes, which will prevent the other linked nodes from joining the network. LEACH works in two discrete phases. The first phase is the setup phase, which includes defining the cluster-heads. The second phase is the steady state phase, which includes transferring the data. In the setup phase, a group of nodes (P) choose themselves to act as cluster-heads. These nodes should select a random number between zero and one. If this random number is greater than a threshold value $T(n)$, then the node n cannot act as a cluster-head. The threshold $T(n)$ is calculated below, where G is the number of nodes that did not act as a cluster-head in the last rotation $(1/P)$.

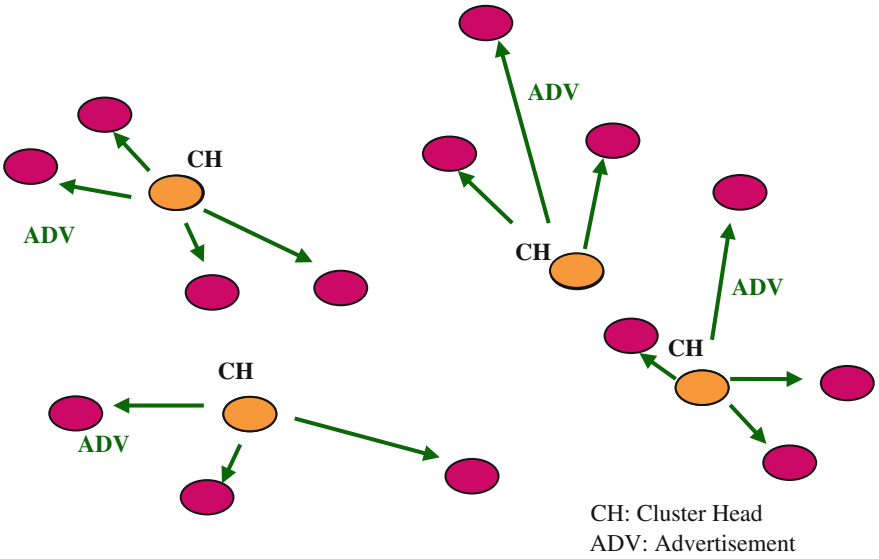


Fig. 5.6 Neighbour nodes join cluster-head in LEACH

$$T(n) = \begin{cases} \frac{p}{1-p\times(r \bmod (1/p))} & if\ n \in G \\ 0 & otherwise \end{cases} \tag{5.1}$$

where

p Is the desired percentage of cluster head (CH)
r Is the current round
G Is the set of nodes that have not been CHs in the last $1/p$ rounds

All the designated cluster-heads send an advertisement to all non-cluster-head nodes to join them as shown in Fig. 5.6. After receiving this advertisement, the non-cluster-head nodes will take a decision as to which cluster-head they want to join. This decision is based mainly on the strength of the signal from the cluster-heads that have reached the node. Therefore, the non-cluster-head will choose the cluster-head that requires the lowest communication energy. After that, the non-cluster nodes will report their decision about the choice of the cluster-head to the other cluster-heads (Al-Karaki and Kamal 2004).

Each cluster-head will build a Time Division Multiple Access (TDMA) schedule for all the nodes within its cluster. Each node will transfer the data to the cluster-head according to the time schedule. The cluster-head then aggregates the data in order to reduce the data size. Finally, the aggregated data will be sent to the sink node. In LEACH there is no way to systematically distribute the cluster-head role to the sensor nodes inside the network. Moreover, LEACH assumes that all the energy levels in the network are the same. LEACH also assumes that each node has data to send at a particular time.

5.2.2.2 Threshold Sensitive Energy Efficient Sensor Network

Threshold sensitive Energy Efficient sensor Network (TEEN) protocol is another Hierarchical routing protocol. Unlike LEACH, which only has a one-tier hierarchy, the network architecture in TEEN is based on multilevel hierarchical grouping. Figure 5.7 shows a two-tier hierarchical architecture, where the sensor nodes communicate with their first-level cluster-heads and these first-level cluster-heads communicate with their second-level cluster-heads. The second-level cluster-heads directly communicate with the sink node. This process takes place for all the levels. The CH in each cluster collects data from its cluster members, aggregates the data, and sends it to an upper-level CH or the sink node. This multilevel architecture enlarges the coverage of the sensor networks and reduces the effect of the constraints of power and transmission range of the sensor nodes as a node does not have to reach the sink node directly. The data from low-level clusters may travel through multiple CHs before reaching the sink node.

TEEN (Manjeshware and Agrawal 2001) is useful for applications measuring physical phenomena such as sensing temperature, and pressure. TEEN is also suitable for real time applications such as fire alarms. The sensing process in TEEN happens instantaneously, while the data sending process happens periodically. TEEN uses cluster formation for sending data. Two thresholds value will be sent by the cluster-head to the non-cluster-head nodes inside its cluster. One is called the hard threshold, which is the threshold value of the attribute beyond which the sensing node must switch on its transmitter and report the sensing data

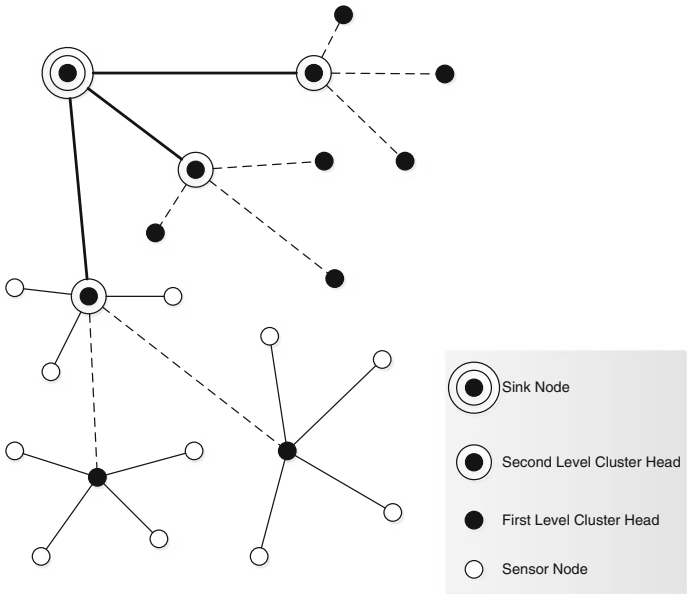


Fig. 5.7 An example of network hierarchies in TEEN

to it CH. The other is called the soft threshold, which contains the small change in the value of the sensed attribute, which triggers the node to switch on its transmitter and transmit the sensed data to the CH.

An enhanced version of TEEN is called the Adaptive Threshold sensitive Energy Efficient sensor Network routing protocol (APTEEN) (Manjeshware and Agrawal 2002). APTEEN aims at proactively capturing periodic data collections and reactively responding to time-critical events. The cluster-heads broadcast the hard and soft threshold, and schedule the transmission time to all the wireless sensor nodes within their cluster. These nodes are allowed to transmit the sensed data when the data values are above the hard threshold. The wireless sensor node will also transmit the data when the change of the attribute value is equal to or greater than the soft threshold. The maximum period between two successive reports for each node is called the count time. This count time is used to assign a certain time for each node to send the sensed data. If the sensor node does not transmit any data during the count time, a TDMA schedule will be used to assign a time slot for each node and force the node to sense and transmit the data. APTEEN supports three different query types (Hu and Cao 2010):

- Historical: to analyse past data
- On demand: to take a snapshot view of the network
- Persistent: to monitor an event for a period of time

The performance of TEEN and APTEEN is better than LEACH in terms of increasing the network lifetime and saving energy. On the other hand, the extra overhead of forming a cluster still exists for both protocols. The threshold functions and determination of the count time increase the complexity of implementation and overhead inside the network.

5.2.3 Location-Based Routings Protocols

The third category of the WSNs routing protocols, based on the network structure, is the location-based routings protocols. The main idea of routing protocols in this category is to utilize the advantage of the locations of the wireless sensor nodes in the routing of the data. The address of each node is determined, based on its physical location. The location of each node may be determined by satellite through the Global Positioning System (GPS) technique or other positioning techniques. The distance to the neighbours can be calculated depending on the signal strength. Two typical location-based routing protocols will be described in this section.

5.2.3.1 Geographic Adaptive Fidelity

Geographic Adaptive Fidelity (GAF) is concerned primarily with energy awareness. GAF was originally designed for the wireless ad hoc networks, but it is also suitable for WSNs. GAF saves energy without affecting the routing dependability.

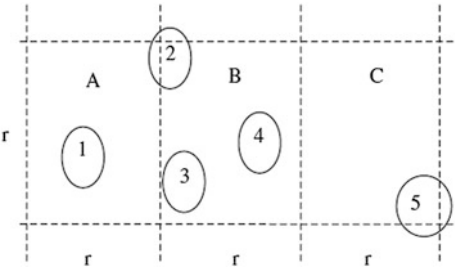


Fig. 5.8 Virtual grid zones in GAF

The main principle of GAF is to divide the sensors field into fixed virtual grid zones, as shown in Fig. 5.8. The cost of routing for each node in the same zone will be symmetric. Therefore, some of these nodes in the same zone can be ignored, by putting them into sleep mode, thus saving more power. GPS can be used to determine the position of each node in the same zone (Xu and Heidemann 2001). Figure 5.8 shows a sensor field divided into zones A, B, and C. Node 1 is located in zone A, nodes 2, 3 and 4 are located in zone B, and node 5 is located in zone C. r is the size of the virtual grid. Nodes 1 and 5 can communicate with nodes 2, 3 and 4, but nodes 1 and 5 cannot directly communicate with each other as they belong to zones A and C respectively and are separated by zone B.

There are three stages in GAF, the discovery stage, the active stage, and the sleep stage. The discovery stage includes discovering the neighbours of each node within the grid. In the active stage, nodes participate in routing data. The sleep stage consists of turning off the node's transmitter and setting the node to sleep mode. In Fig. 5.8 two of nodes 2, 3 and 4 in zone B can be turned off at the same time while keeping one of them awake for communication. It is obvious that this routing protocol depends on GPS technique to determine the positions of the wireless sensor nodes, and this is not always available especially for indoor applications. Moreover, this routing protocol places extra overhead on the memory unit in order to save each node's neighbours address.

5.2.3.2 Geographical and Energy-Aware Routing Protocol

The Geographic and Energy-Aware Routing protocol (GEAR) (Yu et al. 2001) attempts to deliver data to all the nodes inside a target region, which is common in data-centric WSN applications. The GEAR protocol uses the geographical information to route data to a certain area in the network. Routing data to a target region in the GEAR depends on energy and geographically information about the neighbouring regions. The main idea of GEAR is to reduce the number of interests in Direct Diffusion by only sending the interest packets to certain regions or direction in the network, rather than sending the interests to the whole network. This will preserve more energy than Directed Diffusion.

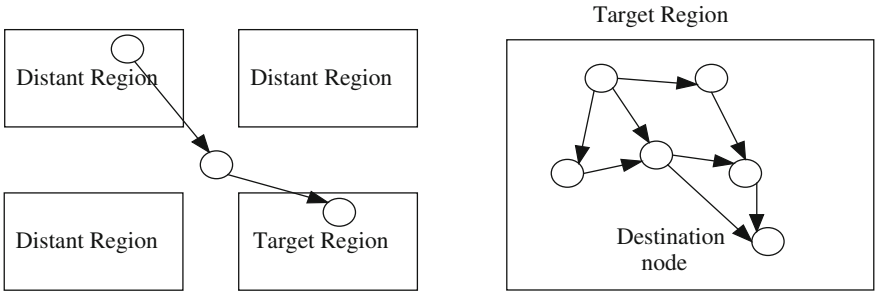
Each node in GEAR keeps two values: an estimated cost and a learned cost. The estimated cost is the combination of distance to the target region and the

remaining energy. When a node does not have any other nearby nodes except itself on a route to the target region, a network hole will be created. The change to the estimated cost caused by routing around network holes is the learned cost. The estimated cost will be equal to the learned cost, if there are no holes in the network. The learned cost will be transmitted back one hop every time the data packet reaches the target region. Based on the energy-aware information, GEAR picks the next hop neighbours intelligently to route the data to the target region in an energy-efficient way. Once the data reaches the target region, GEAR disseminates the data to all the nodes in the region.

There are two phases in the GEAR algorithm as shown in Fig. 5.9. The first phase is forwarding the packets towards the target region. The second phase is disseminating the packet within the target region. In the first phase, the wireless sensor node that received the data makes sure that there is at least one neighbour node closer to the destination area. If there is more than one neighbour node, then the sensor node will choose the one nearest to the target region. If there are no neighbours on the way to the target region, then this node is marked as a network hole. In the second phase, after the data packet has reached the target region, it can be distributed by restricted flooding or recursive geographic forwarding.

5.3 AODV Routing Protocols

AODV—Ad hoc On Demand Distance Vector routing protocol is one of the most discussed and advanced routing protocols. Its main developers are Charles E. Perkins (Nokia) and Elizabeth Belding-Royer (UCSB). The ZigBee standard implements AODV and Motorola's Cluster-Tree routing protocols in their stack, which makes the AODV and Cluster-Tree routing protocols widely used in industry. This section presents the principle of AODV and gives the implementation detail of a simplified version of AODV.



Phase I: Deliver data to the target region using the shortest path

Phase II: Flood data in the target region

Fig. 5.9 Two phases in GEAR

5.3.1 Principle of the AODV Routing Protocols

The Ad hoc On-Demand Distance Vector (AODV) algorithm is a dynamic, self-starting, multi-hop routing protocol that enables participating mobile nodes to establish and maintain an ad hoc network. AODV allows mobile nodes to obtain routes quickly for new destinations and does not require nodes to maintain routes to destinations with which it is not in active communication. AODV also allows mobile nodes to respond in a timely manner to link breakages and changes in network topology.

There are three types of messages defined in AODV, Route Requests (RREQ), Route Replies (RREP), and Route Errors (RERR). When a route to a new destination is needed, the node broadcasts a RREQ to find a route to the destination. Each node receiving the request caches a route back to the originator of the request in a backward table, so that the RREP can be unicasted from the destination along a backward path to that originator. A route can be determined when the RREQ reaches the destination or a node that offers reachability to the destination. The route is made available by unicasting a RREP back to the origination of the RREQ and setting-up a routing table at each node. Figure 5.10 illustrates the RREQ and RREP message transmission.

For nodes monitoring the link status of next hops for active routes, a HELLO message is regularly sent out to detect a link break in an active route, if there is no ACK received, the broken link is invalidated. Therefore a RERR message is typically transmitted to notify other nodes that the loss of that link has occurred. The RERR message indicates that the destination is no longer reachable by way of the broken link.

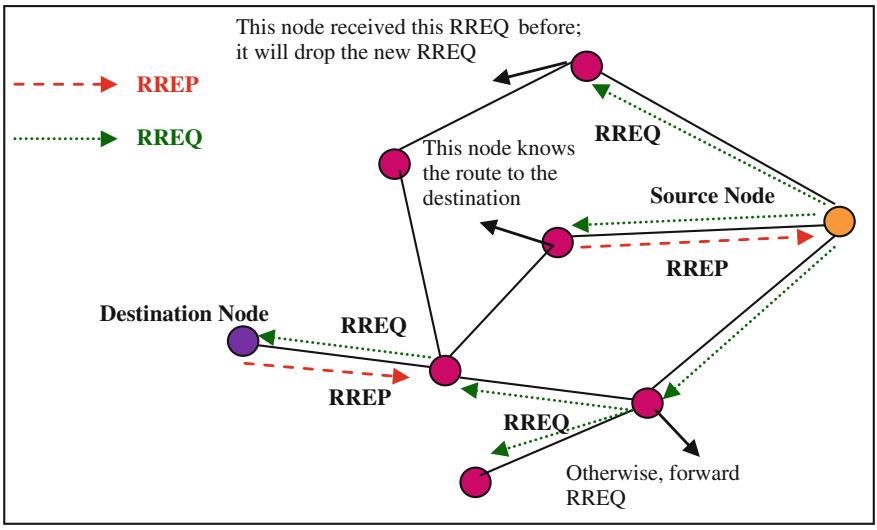


Fig. 5.10 RREQ and RREP transmission in AODV

5.3.2 AODV Message Formats

The formats of RREQ, RREP and RERR messages have been defined in the Internet Engineering Task Force (IETF) Mobile Ad hoc Networking Working Group document (Perkins and Royer 2003). Figures 5.11, 5.12, and 5.13 show the details of these formats. The meanings of the fields are listed in Table 5.1.

5.3.3 Implementation of a Simplified Version of AODV

This section shows the implementation of a simplified version of AODV realized in sensor nodes with a Contiki Operating System (OS). A WSN consisting of four sensor nodes is shown in Fig. 5.14, in which node S needs to send some sensed values to Node R. Nodes S and R are not in communication range. Transmitter nodes T_1 and T_2 are two routers for relaying the received messages. Node R is the destination node.

Initially, as Node S doesn't know a route to Node R, it must broadcast a RREQ message. Both Nodes T_1 and T_2 will receive the message, but, since they're not the destination nodes, they will retransmit the message to Node R. This could cause a collision, since both Nodes T_1 and T_2 could transmit at the same time. A very simple collision avoidance mechanism has been implemented: the transmitters wait for a random period of time before sending the message. When the message is delivered to Node R, it unicasts back a RREP message to the two transmitters T_1

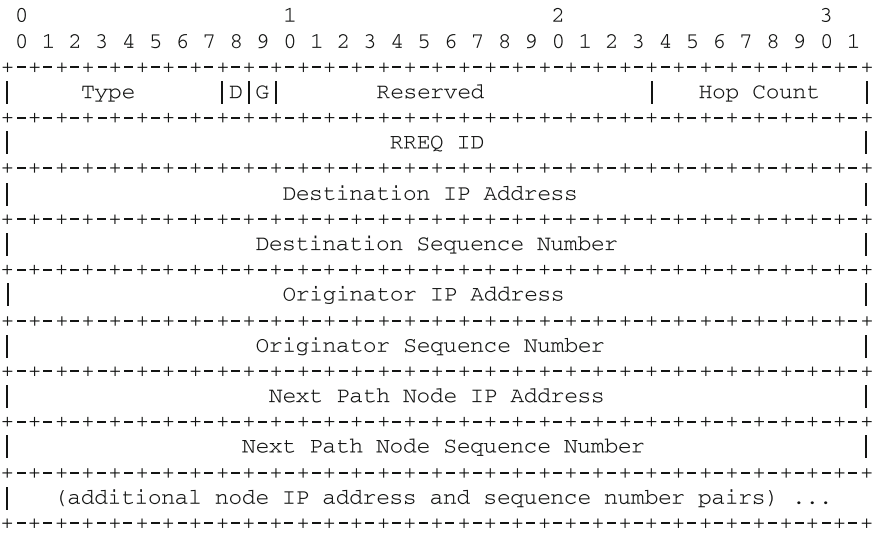


Fig. 5.11 Format of RREQ message

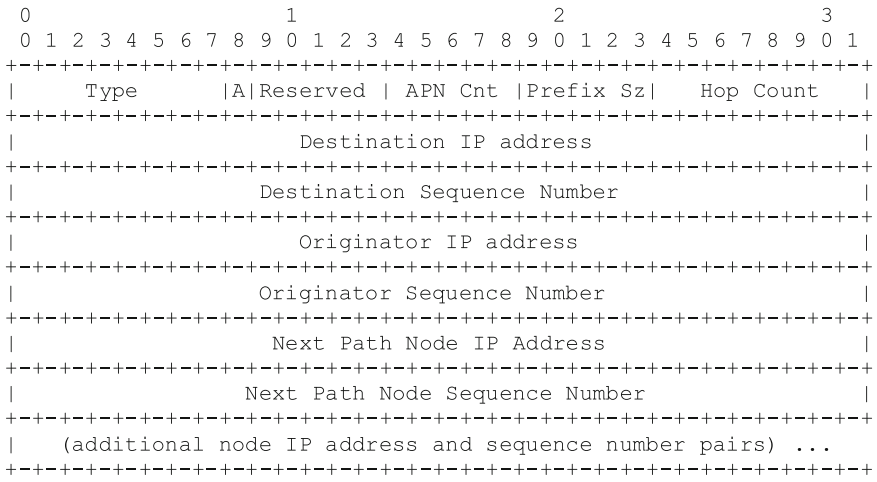


Fig. 5.12 Format of RREP message

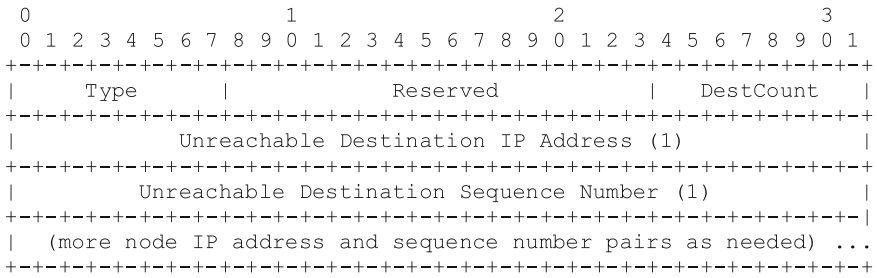


Fig. 5.13 Format of RERR message

and T_2, and then to Node S. Two criteria could be used for Node S to determine the route. Either the router with the higher RSSI is selected or the router with the highest battery level. Once the route has been discovered, it will be used by Node S to send the sensed values (in a unicast message) to Node R. The information flow of the routing protocol is shown in Fig. 5.15.

5.3.3.1 Types of Messages

Three types of messages are defined as below:
Route Request Message:

- Type (8 bit): identifies the message type (COMMAND_ROUTREQUEST)
- Destination address (16 bit): the Rime address of the destination for which a route is desired

Table 5.1 Meanings of the fields in AODV packets

Field	Description
Type	1—RREQ, 2—RREP, 3—RERR, 4—ACK (acknowledgement)
D	Destination-only flag. If set a intermediate node may not respond to this RREQ
G	Gratuitous Route Reply flag. Indicates whether a gratuitous RREP should be sent to the destination
Reserved	Sent as 0; ignored on reception
Hop count	The number of hops from the originating node to the node handling the request or destination node
Dest count	The number of unreachable destinations included in the message; MUST be at least 1
APN count	The number of accumulated path nodes appended to the RREP
Prefix size	The number of the nodes within a destination's subnet that are reachable via the same route
RREQ ID	A sequence number uniquely identifying the particular RREQ
Destination IP address	The IP address of the destination node
Destination sequence number	The sequence number associated with the destination node
Originator IP address	The IP address of the originating node that issued the Route Request (RREQ) or from which the route is supplied (RREP)
Originator sequence number	The current sequence number of the originating node
Next path node IP address	The IP address of the next node along the path from the Originator to the Destination
Next path node sequence number	The sequence number of the next node along the path from the originator to the destination
Unreachable destination IP address	The IP address of the destination that has become unreachable due to a broken link
Unreachable destination sequence number	The sequence number in the route table entry for the destination listed in the previous unreachable destination IP address field

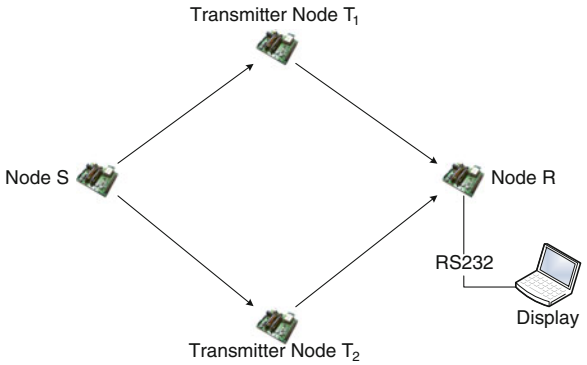


Fig. 5.14 A four nodes wireless sensor network

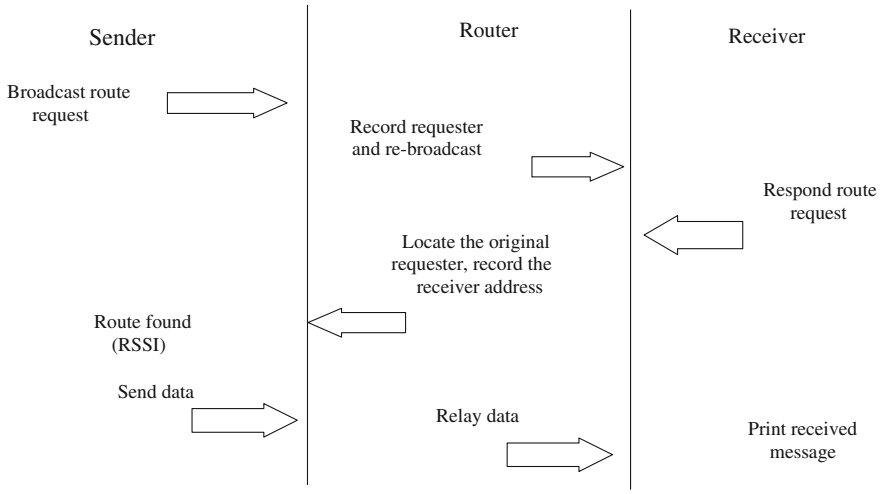


Fig. 5.15 Information flow of the routing protocol

- Broadcast counter (8 bit): specifies how many times the current message has been sent
- Broadcast limit (8 bit): the maximum number of times a message can be broadcasted
- Broadcast id (8 bit): identification number of the current broadcast message

Route Reply Message:

- Type (8 bit): identifies the message type (COMMAND_ROUTERESPONSE)
- Destination address (16 bit): the Rime address of the destination for which a route is supplied
- Battery (16 bit) (OPTIONAL): the battery level of the destination node

Data Transmission Message:

- Type (8 bit): identifies the message type (COMMAND_DATATX)
- Destination address (16 bit): the Rime address of the destination of the sensed data
- Source address (16 bit): the Rime address of the source of the transmitted data
- Temperature (16 bit): the sensed temperature data to be transmitted
- Battery (16 bit): the battery level data to be transmitted

A routing table *tsRouteTable* and a backward table *tsBwsTable* are defined as two data structures:

- *struct tsBwdTable*: is used only in the router and defines the structure of an entry of the backward table that contains:

  - BrdcstID: to identify the broadcast message (8 bit)
  - From: the source device from which the route request is received (16 bit)
  - Dest: the address of the destination for which a route is desired (16 bit)

- *struct tsRouteTable*: defines the structure of an entry of the routing table, that contains:
  - Dest: the destination address (16 bit)
  - NextHop: the next hop to reach the destination (16 bit)
  - Battery: the battery level (16 bit)
  - Rssi: the RSSI (Received Signal Strength Indication) level (bit)

5.3.3.2 Software Architecture Design

Two different software applications must be developed, one for the sender and one for the routers and receiver; the sender software application will be responsible for reading data, routing and forwarding, and the receiver software application will receive the data and forwards or displays it. The receiver software application should fit on both the routers and the receiver node as both kinds of nodes receive the same types of packets.

Figure 5.16 shows the flowchart for the sender application where two timers have been set. Timer 1 is set for determining the time interval of sensor reading. Timer 2 is for route responses. The detail of the route request is shown in Fig. 5.17. When a route discovery is required, the route request message (RREQ) must be broadcasted to the neighbours and wait for route response message (RREP). Afterwards, the sender program will figure out the best path to the destination. When the best path is calculated, the routing table will be updated and the sensor readings will be forwarded to the next hop as a unicast connection. The criteria used in selecting a route here is to use the node with a higher battery level first and then choose the node with a higher RSSI level. The logic of the criteria is shown in Fig. 5.17.

Figure 5.18 shows the abstract flowchart of the router application. The detail of each component is omitted for the sake of the simplicity. The router can receive a broadcast message (RREQ) or two types of unicast messages, RREP and Data Transmission (DATATX). If the packet is RREQ and the current node is not the destination node, the RREQ will be rebroadcasted. If the current node is the destination node, a RREP will be sent back. If the packet is DATATX, the packet will be forwarded to its receiver. If the packet is RREP, it will be forwarded to the sender node. The backwards table and the routing table should be updated as necessary. If the broadcast limit has been reached, it must discard the current message in order to avoid useless bandwidth occupation; otherwise the broadcast counter will be increased by 1.

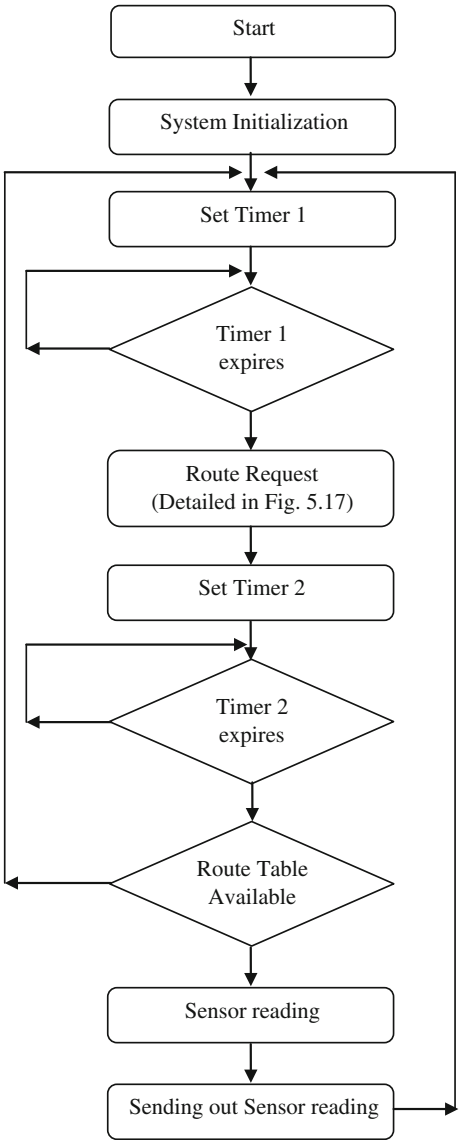


Fig. 5.16 Flowchart of the sender application

5.4 Cluster-Tree Routing Protocol

The cluster-tree routing protocol (Lee et al. 2006) is another routing protocol implemented in the ZigBee stack, and also has been widely used in industry. It is a self-organised protocol that supports network redundancy in order to achieve fault tolerance in the network. The cluster-tree protocol uses packets negotiation to form either a single cluster network or a multi-cluster network. The cluster formation

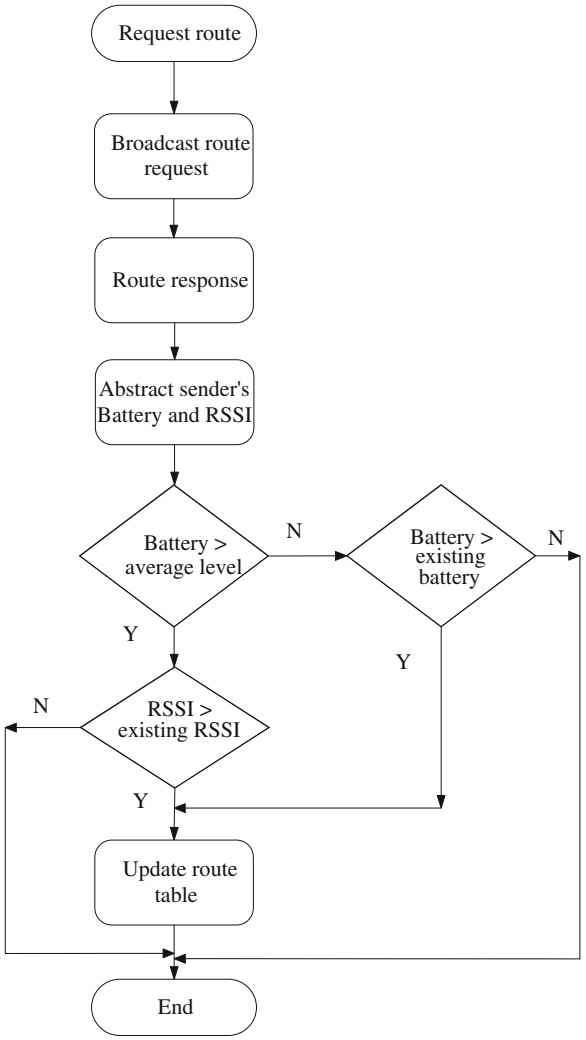


Fig. 5.17 Flowchart of route request in the sender application

process consists of two stages; select the cluster-heads of the WSN and subsequently, the non-cluster-head nodes in the WSN join the cluster-heads in order to form the clusters (Ergen 2004).

5.4.1 Single Cluster Network

A single cluster network contains only one cluster-head. All the nodes are connected to this cluster-head with one hop, and the network topology becomes a star

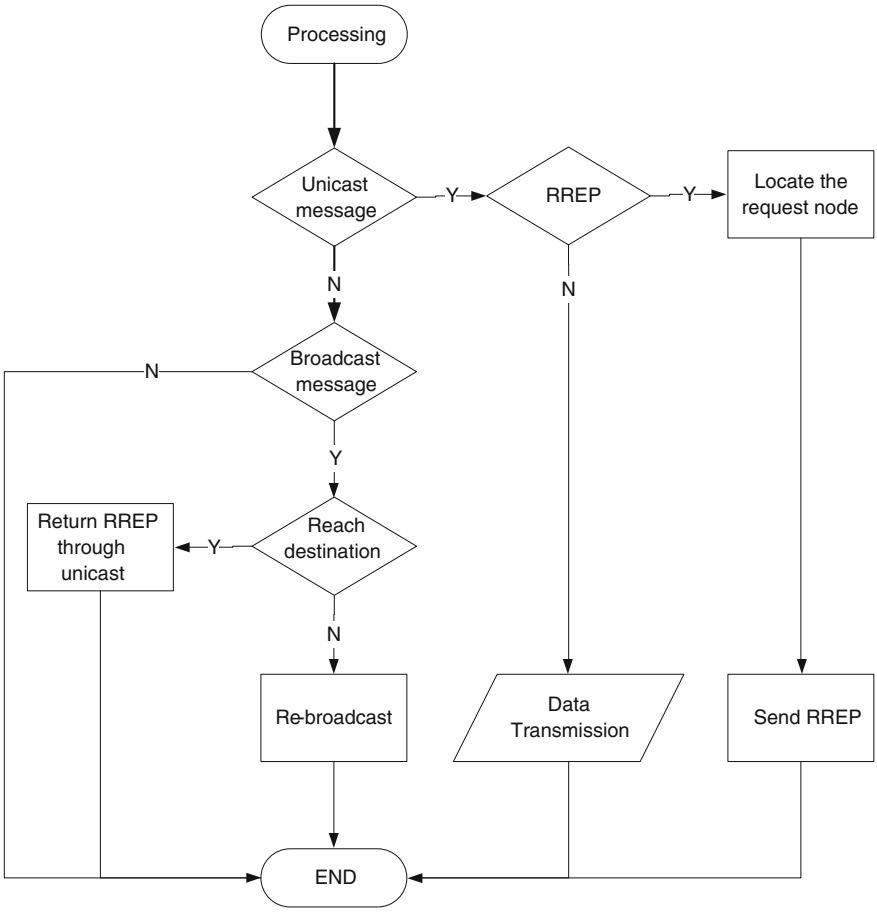


Fig. 5.18 Flowchart of router/destination nodes

topology. Each node in the network is waiting to receive a HELLO packet from the node that acts as a Cluster-Head (CH). The HELLO packet includes the cluster-head MAC address and the cluster-head ID number, which is equal to zero (0) in the single cluster network. If any node fails to receive a HELLO packet after a certain period of time, this node will be converted to act as a cluster-head. Then, it will distribute a new HELLO packet to all its neighbouring nodes, and waits to receive the CONNECTION REQUEST (CON REQ) packet from the neighbours. If it does not receive any CON REQ packets, it will turn back into regular node and wait again to receiving a HELLO packet. The cluster-head can also be selected based on some features such as the transmission range, power level, computing ability, or location information.

As shown in Fig. 5.19, once the cluster-head receives a CON REQ packet from a neighbour node, it will reply with a CONNECTION RESPONSE (CON RES)

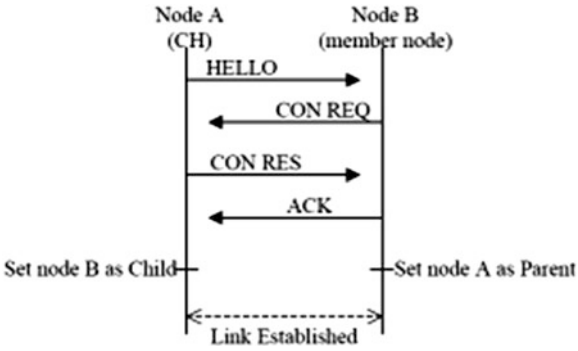


Fig. 5.19 Establish a link between a cluster head and a node (IEEE P802.15 2001)

packet. The CON RES packet includes the node ID of the non-cluster-head node. Finally, the non-cluster-head, that receives the node ID, will send an Acknowledgment packet (ACK) to the cluster-head node. If the cluster-head reaches the maximum limit of the node IDs, or reaches any other defined limitations, it would reject any new node connection request. This rejection is signalled by assigning a special ID to that node. The list entry of all neighbours and the routes would be updated periodically by sending HELLO packets. A node could receive an HELLO packet from node that belongs to other clusters. Consequently, the node saved the Cluster ID (CID) of the transmitting node in its neighbours list. After that, it would transmit the CID with the neighbour node ID inside the LINK STATE REPORT to its cluster-head. Subsequently, the cluster-head would know those clusters with which it has an intersection. The LINK STATE REPORT packet also allows the cluster-head to identify any existing problems in the network. If the cluster-head wants to update the topology of the network, it can be achieved by sending a TOPOLOGY UPDATE packet. If the cluster-head stops working, then the transmitting of the HELLO packet would also be stopped. Therefore, all nodes would know that they have lost the cluster-head. Subsequently, a new cluster-head will be reconfigured by repeating the same process (IEEE P802.15 2001).

5.4.2 Multi-Cluster Network

A multi-cluster network consists of many single clusters and is shown in Fig. 5.20. Multi-cluster networks need a Designated Device (DD) to give a unique Cluster ID to each cluster-head, and to calculate the shortest path from the cluster to the designated device. After the designated device has joined the network, it would act as a cluster-head, and would send HELLO packet to its neighbours. If the cluster-head receives the HELLO packet, it would send a CON REQ and would join the designated device to form the top-level cluster (Cluster 0). If the cluster-head is connected directly to the designated device, the cluster-head will become a border

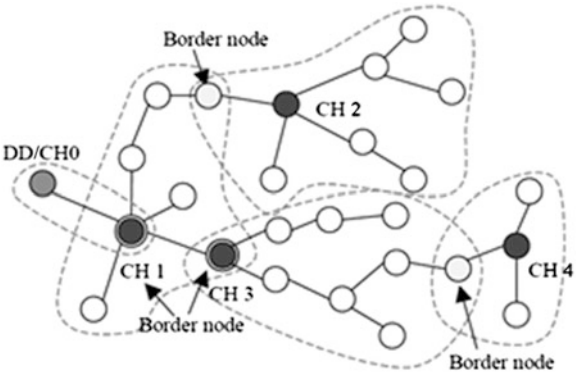


Fig. 5.20 Multi-cluster network consists of many single clusters (IEEE P802.15 2001)

node with two logical addresses. As shown in Fig. 5.21, if a regular node received the HELLO packet from the designated device instead of its cluster-head, it would act as a border node to its parent. The cluster-head would send a NETWORK CONNECTION REQUEST (NET CON REQ) packet to setup the connection with the designated device. Subsequently, the border node would send a CID REQUEST (CID REQ) packet to the designated device. If the designated device sent a CID RESPONSE (CID RES) packet, that contains the new cluster ID (CID), to the border node, the border node would send a NETWORK CONNECTION

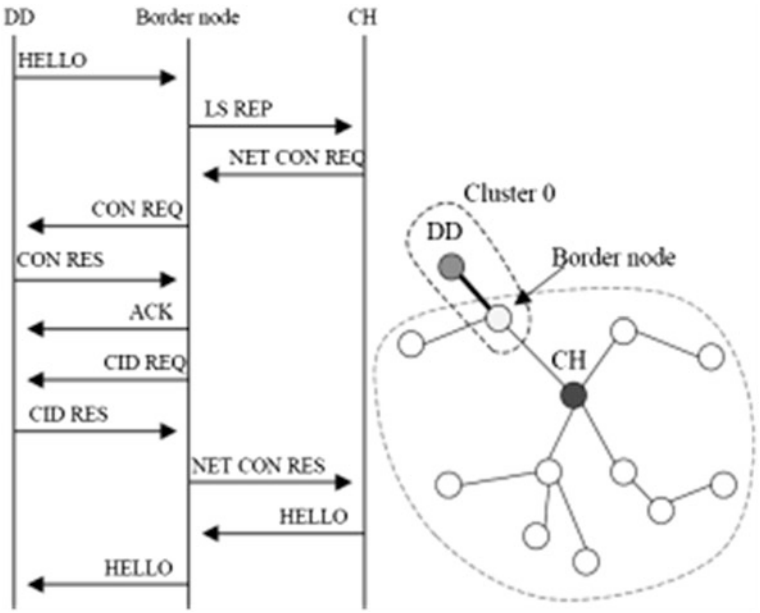


Fig. 5.21 Link CH with DD by border node (IEEE P802.15 2001)

RESPONSE (NET CON RES) packet to the cluster-head with the new CID. In addition, the cluster-head would inform its nodes about the new CID. This process is illustrated in Fig. 5.21.

5.5 Energy-Aware Routing Protocols

AODV is not an energy-aware routing protocol. AODV uses the same route to send all of the data from the source to the destination until this route dies. Therefore, the intermediate nodes on this route between the source and the destination nodes quickly expended their energy and die. As a consequence, the lifetime of the whole network will be effected, especially if the dead nodes are vital to the network such as being the coordinator or the router nodes. The cluster-tree protocol also does not consider the energy levels of the wireless sensor nodes in choosing the cluster head or determining the number of the clusters required in each network. It uses static nodes to act as cluster-heads during the whole lifetime of the network, which makes these nodes die quickly. This section considers the energy levels of the wireless sensor nodes, which can be applied in both AODV, and cluster-tree routing protocols (Abusaimeh and Yang 2008, 2009).

In order to maximize the lifetime of the PAN coordinator, the routers, and the whole network, it is necessary to balance power consumption and distributing the responsibilities of routing among the wireless sensor nodes especially the routers and the PAN coordinator. Therefore, it will be better to fully exploit the wireless sensor nodes to participate in the communication process and share data transmission (Abusaimeh and Yang 2008). Firstly, the energy model of the wireless sensor network is presented. This energy model consists of several formulas. The first formula is related to the energy consumption rate of each sensor node and can be defined as Eq. (5.2).

$$ConsSpeed = \frac{InitEng - \mathrm{Re}mEng}{TimePeriod} \tag{5.2}$$

where, *RemEng* is the current energy level of the node and *InitEng* is the initial level of the node energy when this node joined the network. *TimePeriod* is the period of time that the node takes to consume the energy. *ConsSpeed* is the energy consumption rate for the node. The next formula is related to the lifetime of each node, *LifeTime*. This lifetime is the period of time for which the node can be kept running before dying or stopping transmitting and receiving signals. The lifetime of a node can be measured by Eq. (5.3).

$$LifeTime = \frac{InitEng}{ConsSpeed} \tag{5.3}$$

The third formula is related to the remaining lifetime of the node, *RemainTime*, and described by Eq. (5.4). This remaining lifetime means the period of time left for the node to keep running and serving the route.

$$RemainTime = \frac{RemEng}{ConsSpeed} \tag{5.4}$$

The remaining energy of the node indicates the level of energy left in the node and can be measured by Eq. (5.5),

$$\mathrm{Re}mEng = IniEng - ((PktT \times txPower) + (PktR \times rxPower)) \tag{5.5}$$

where *PktT* is the count of packets transmitted by this node. *txPower* is the transmission energy required to transmit each packet. *PktR* is the count of the received packets. *rxPower* is the energy consumed by receiving one packet.

The lifetime of WSNs can be increased by distributing the role of routing and balancing the energy consumption among the whole collection of nodes. Lifetime maximization can be achieved by taking into account the changes in the energy level of the wireless sensor node batteries simultaneously with the path discovery process and the packet forwarding process. In the energy-aware routing protocol, most of the nodes will act as intermediate nodes between the source and destination nodes. A new route will be established if an intermediate node along the pass to the destination has a lower energy level or a higher energy consumption rate. The energy cost of establishing a new route, *EstRouteCost*, can be obtained by Eq. (5.6),

$$EstRouteCost = HopsNo \times txPower \times Time \tag{5.6}$$

where *HopsNo* is the number of hops between the source and destination node, *txPower* is the transmission power for one packet, and *Time* is the time needed to transmit these discovery packets. Because of the very short time needed to establish a new route, the whole energy cost of establishing a new route could be insignificant.

In addition, the energy-aware routing will aim to keep most of the nodes running for their maximum lifetime. Each node, which has a high energy consumption rate and a short remaining lifetime, should be turned off for a period of time. A high energy consumption rate is determined by comparing the node's energy consumption rate with other nodes. Turning off a node will make the energy-aware routing protocol choose an alternative node or change the whole route to the destination node. Repeating this process can distribute the routing role among most of the nodes; therefore balance the power consumption in the network as a whole.

Figure 5.22 shows the additional steps necessary for the energy-aware routing within the existing routing protocols such as AODV or Cluster-tree routing. These steps are as follows:

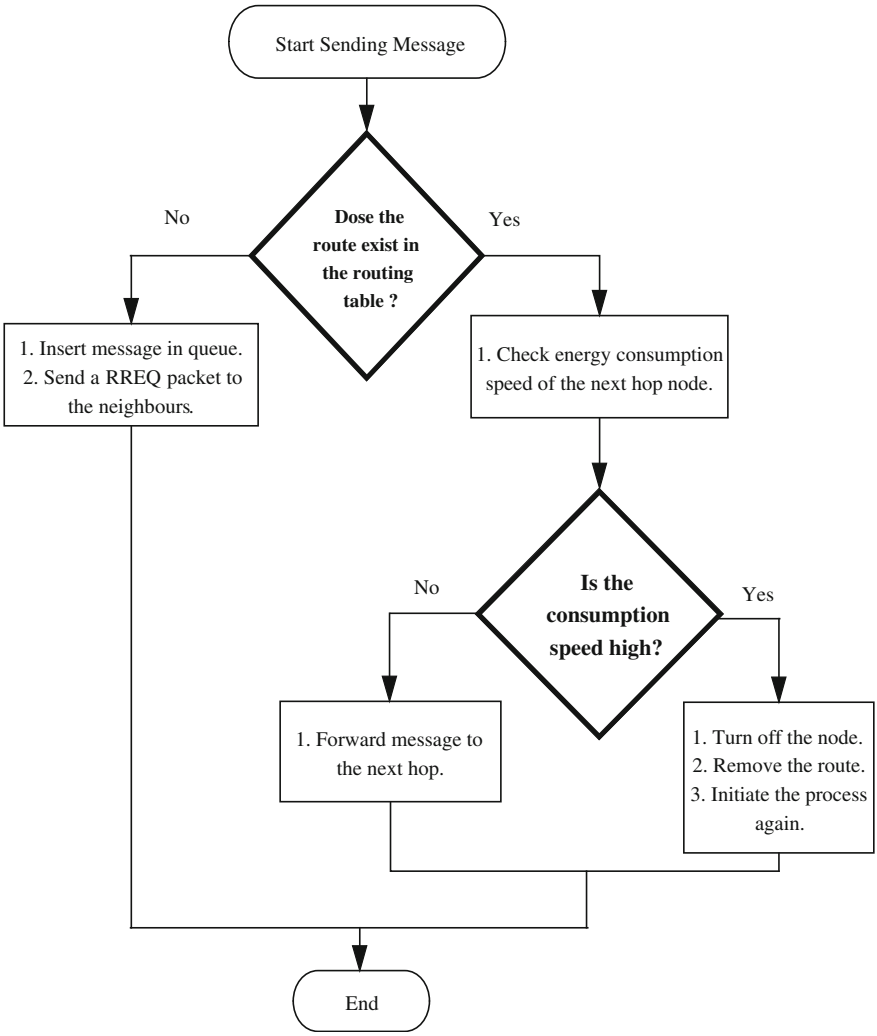


Fig. 5.22 Energy-aware AODV

Step 1: If a sensor node needs to send a message, it has to check its routing table to find a path to the destination node. Therefore, if the route is available in the routing table, it will forward the message to the next node. Otherwise, the message will be saved in a queue, and the source node will send the RREQ packet to its neighbours initiating the discovery process.

Step 2: Before forwarding the message to the next hop, the energy consumption rate of the next hop is checked.

Step 3: If the energy consumption rate is high, then the next hop will be turned off for a prescribed period of time. The route will then be removed from the routing table, which will cause the source node to initiate the discovery process again and find a new path to the destination node.

5.6 Summary

In this chapter, we initially present the features and constrains in designing routing protocols for WSNs. The classifications of the routing protocols for WSNs in terms of network structure are elaborated; these are flat routing, hierarchical routing, and location-based routing. A few typical routing schemes for each category are described. AODV and Cluster-tree routing protocols, which are implemented in the ZigBee standard, have been introduced in detail. The implementation of a simplified version of AODV has been presented to show the detail of the algorithm. Methods to make AODV, Cluster-tree or other routing protocols energy-aware are presented at the end of Chapter to inspire further research in this rich area.

References

Abusaimeh, H., Yang, S.H.: Balancing the power consumption speed in flat and hierarchical WSN. Int. J. Autom. Comput. **5**(4), 366–375 (2008)

Abusaimeh, H., Yang, S.H.: Dynamic cluster head for lifetime efficiency in WSN. Int. J. Autom. Comput. **6**(1), 48–54 (2009)

Akkaya, K., Younis, M.: A survey of routing protocols in wireless sensor networks". Elsevier Ad Hoc Netw. J. **3**(3), 325–349 (2005)

Al-karaki, J.N., Kamal, A.E.: Routing techniques in wireless sensor networks: a survey. IEEE Wirel. Commun. **11**(6), 6–28 (2004)

Ergen, S.C.: ZigBee/IEEE 802.15.4 summary. Available online at http://www.sinemergen.com/zigbee.pdf (2004)

Hu, F., Cao, X.: Wireless Sensor Networks Principles and Practice, pp. 109–149. CRC Press, Boca Raton (2010)

IEEE P802.15 Working Groups for WPANs: Cluster tree network (2001)

Jolly, V., Latifi, S.: Comprehensive study of routing management in wireless sensor networks. In: Proceeding of the 2006 International Conference on Wireless Networks, pp. 37–44. Las Vegas, USA (2006)

Lee, K.K., Kim, S.H., Choi, Y.S., Park, H.S.: A mesh routing protocol using cluster label in the ZigBee network. In: Proceeding of the 2006 IEEE International Conference on Mobile Ad hoc and Sensor Systems, pp. 801–806. Vancouver, BC, Canada (2006)

Manjeshware, A., Agrawal, D.P.: TEEN: a routing protocol for enhanced efficiency in wireless sensor networks. In: Proceeding of the 15th International Workshop on Parallel and Distributed Computing, pp. 2009–2015. San Francisco, California, USA (2001)

Manjeshware, A., Agrawal, D.P.: APTEEN: a hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks. In: Proceeding of the International Workshop on Parallel and Distributed Computing, pp. 195–202, Ft. Laudeerdale, FL, USA (2002)

Perkins C.E., Royer, E.M.: Ad-hoc on-demand distance vector routing, The mobile Ad-hoc network (MANET) working group IETF, pp. 173–219 (2003)

Vidhyapriya, R., Vanathi, P.T.: Conserving energy in wireless sensor networks. IEEE Potentials **26**(5), 37–42 (2007)

Xu, Y., Heidemann, J.: Geography-informed energy conservation for ad hoc routing. In: International conference on mobile computing and networking, pp. 70–84, Rome, Italy (2001)

Yu, Y., Govindan, R., Estrin, D: Geographical and energy-aware routing: a recursive data dissemination protocol for wireless sensor networks, UCLA computer science department technical report, UCLA-CSD TR-01-0023, pp. 1–11 (2001)

Chapter 6
Optimization of Sink Node Positioning

Keywords Sink node positioning · Static mobile, dynamic sink node · Evolutionary computation

6.1 Introduction

Although WSNs may take various topologies such as star, ring, mesh or tree, the signals from the individual sensors of WSNs are often transferred to the Internet or other terminals via several sink nodes. A sink node is a designated device connected to ordinary sensor nodes but more powerful than them and bridging a sensor network with the end users. The sink nodes may be envisioned as a laptop computer, receiving data from the network, or a much smaller and dedicated micro-controller providing the gateway function. As illustrated in Fig. 6.1, each sensor node has the capabilities to collect data and route it back to the sink node and the end users. Data are routed back to the end user through the sink node.

There might be more than one sink nodes simultaneously working in a sensor network, depending on the application requests. Figure 6.2 shows two sink nodes *X* and *Y* exhibiting identical interest in an event occurring in sensor node *B*. In Fig. 6.2, when two sink nodes are deployed, sensor node *A* is one hop away from its nearest sink node *Y*, and many hop away from sink node *X*. Thus, by employing two sink nodes instead of one sink node, sensor node A will need fewer hops and less power to transmit its signal to a sink node. We know the energy consumed in routing a message for any sensor node to its nearest sink node is proportional to the number of hops the message has to travel. Employing multiple sink nodes effectively reduces the energy consumption per message delivered. As the number of sink nodes is increased, the path length from a sensor node to a sink node is decreased and the lifetime of the sensor node is increased. However, the number of sink nodes is constrained financially because the cost of the sink node is more expensive than the sensor node. In addition, on some occasions, employing multiple sink nodes might not be physically practical.

S.-H. Yang, *Wireless Sensor Networks*, Signals and Communication Technology,
DOI: 10.1007/978-1-4471-5505-8_6,

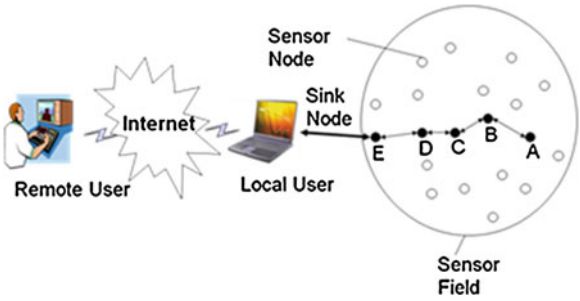


Fig. 6.1 Sink node with scattered sensor nodes

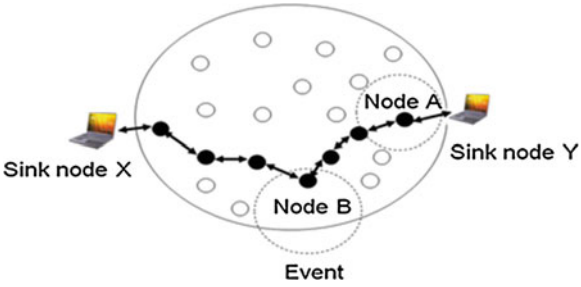


Fig. 6.2 Multiple sink nodes with scattered sensor nodes

Location of the sink node can influence the network performance. Heinzelman et al. (2000) demonstrated through experimental results that the sensor nodes which are one-hop away from a sink node drain their energy faster than other nodes in the sensor network. This is because nodes which are one-hop away from a sink node need to forward messages originating from many other nodes, in addition to delivering their own generated messages. The workload of these nodes, which are near the sink node, is much bigger than ones further away from the sink node. Therefore, these sensor nodes deplete their energy more quickly and finish their lifetime. If too many sensor nodes around the sink node die, i.e. become non-operative, other living sensor nodes will be unable to communicate with the sink node via these dying sensor nodes and the sensor network becomes non-operative.

6.2 Challenges of Sink Node Positioning

There are many challenges in optimally locating sink node positions in wireless sensor networks. These include a large, if not infinite solution space, an excessively large number of parameters involved, differences in routing algorithm, different application needs, the involvement of a huge number of sensor nodes, and different sensor node capabilities.

- **Large, if not infinite solution space**: sink nodes can be located everywhere in the environment and are not enumerated (Oyman and Ersoy 2004). The number of possible solutions is very large when no restrictions are applied.
- **Involvement of large number of sensors**: Sensor networks may involve thousands of sensor nodes. The involvement of a huge number of sensor nodes turns the sink node positioning problem into NP-complete.
- **Dynamic topological changes**: the deployed sensor nodes may fail due to manufacturing defects or energy depletion, which consequently require a change in the sensor network topology and the sink node location.
- **Differences in the node capability**: sensor nodes are not identical, e.g. some are made with variable transmitters while others are not. In this case, minimizing the communication distance in sensor networks consisting of sensor nodes with fixed transmitters and various communication ranges provides better energy consumption, but no energy saving can be obtained if only sensor nodes with a fixed range transmitter are used.
- **Network architecture differences**: there are two types of typical sensor network architecture: flat and hierarchical. Flat sensor networks disseminate data in multi-hops through intermediary nodes in the network whereas in hierarchical sensor networks, data are disseminated to the sink node through cluster-heads.
- **Routing algorithm differences**: routing algorithms are not created equally and they employ different techniques to optimize data communication tasks in sensor networks. Each routing algorithm creates a unique data delivery structure in transmitting data to the sink node. Such differences lead to different energy models.
- **Sampling mode differences**: wireless sensor networks need to support periodical data sampling or is operated in an event-driven mode. The need to support both sampling modes increases the complexity of the optimal sink node positioning problems. On the other hand, choosing to optimize for a specific data sampling mode may need different considerations (Sohraby et al. 2007).

6.3 Categories of Sink Node Positioning Approaches

The bulk of the research on sink node positioning has focused on the careful selection of the sink node location, which may affect various performance metrics such as energy consumption, delay, and throughput. They emphasise structural quality metrics, such as distance and network connectivity, and/or base their analysis on a fixed topology. Therefore, we classify them as static approaches. However, dynamically adjusting the sink node location can further increase the dependability of WSNs because the optimality of the initial position for the sink node may become void during the operation of the network, because of changes in the state of the network or various external factors. For example, in a target tracking application, it is reasonable to relocate the sink node close to where targets are detected and where traffic volume is high. We classify such methods as dynamic approaches.

6.3.1 Static Positioning of Sink Nodes

In static sink node positioning, the sink node does not have any capability to move. Its location remains fixed throughout the network lifetime operation. Considerable research has been done on optimization at network set-up time, the positioning of single or multiple sink nodes in WSN. The published works can be categorised according to the assumptions made, the network model considered, the network state information available, and the metrics to be optimized.

Extending the network lifetime is the key objective of static sink node positioning. Multiple variants of the sink node positioning were pursued (Akkaya et al. 2007). The difference is either due to the definition of network lifetime, the network operation mode, or the network state parameters that are included in the optimization objective. Although some users consider that the network lifetime is the time until the first sensor node dies, many other use the failure of a percentage of deployed sensors as indicative of the network lifetime. Other works strive to extend the network lifetime by minimizing the total power consumed in collecting the readings of all sensors.

The network topology and system model considered also are a differentiating factors. In a flat network topology, sensors are homogenous, having the same amount of initial energy and usually forming multiple routes to relay their data to the sink node. In a hierarchical topology sensor nodes are grouped into clusters with a designated cluster head. In such a case, the scope of the sink node positioning problem is reduced to the inter-cluster head network. Figure 6.3 shows a two-tiered hierarchical architecture of WSNs, where SN means sensor node, AN

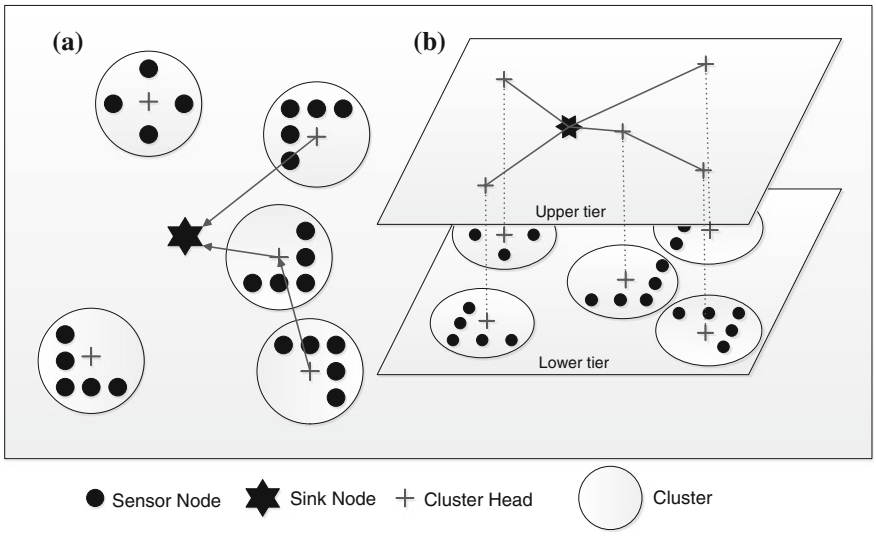


Fig. 6.3 Two-tiered architecture of WSNs. **a** Physical view. **b** Logical view

means application node which is a cluster head, and BS means base station, another name for sink node. The sink node is located at a position which would minimize the maximum distance between cluster heads and the sink node, or minimize the transmission power consumed by the cluster head to transmit the data to the sink node (Pan et al. 2005).

6.3.2 Dynamic Sink Node Positioning

Static sink node positioning does not consider dynamic changes during the network operation and therefore does not move the sink nodes once they are deployed in their original locations. Examples of dynamic changes are: traffic patterns which can change based on the monitored events; load many not be balanced among the nodes, causing bottlenecks; application-level interest can vary over time; and the available network resources may change due to the depletion of energy in some nodes (Akkaya et al. 2007).

Dynamically repositioning the sink nodes while the network is operational can further improve the performance of the network. For example, in a target tracking application, it may be wise for the sink node to keep a certain distance from a harmful target. A safe distance should be maintained. In a disaster management application, sensors can detect fires, collapsing buildings, gas leaks, and so on. Moving too close to these reported events in such scenarios would be risky. Another example is that when many sensor nodes in the vicinity of the sink node become dysfunctional due to the exhaustion of their batteries. It is better for the sink node to reposition itself to become easily and reliably reachable by data sources. Vincze et al. (2006) suggested that the sink node should be repositioned adaptively in an even-driven sensor network based when the events happen. Akkaya et al. (2005) proposed a three-step approach: when would it make sense for the sink node to relocate to, where should it go, and how will the data be routed while the sink node is moving? Akkaya et al. (2007) presented three sample heuristics for dynamic relocation of the sink node that can improve the network performance in terms of energy consumption, data delivery delay and safety of the sink node. They are repositioning for increased network longevity, enhancing timeliness of delay-constrained traffic, and protecting the sink node.

6.3.3 Mobile Sink Node Positioning

Dynamic sink node positioning considers that a sink node can move on-demand or otherwise would stay stationary. A sink node might be designed as a mobile device, which is able to move constantly within a sensor field rather than on-demand. A mobile sink node can be used to collect data from a sparsely populated sensor network by moving itself closer to those sensor nodes where data originated. The movement of sink

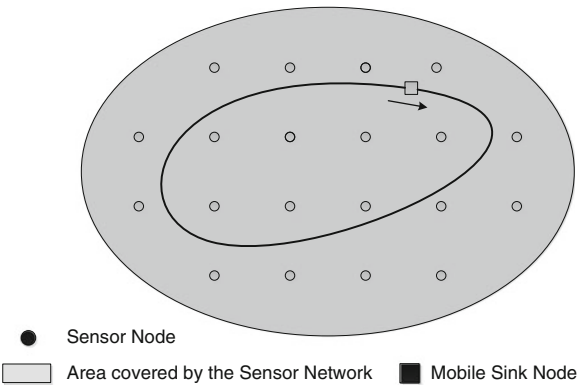


Fig. 6.4 Sensor network with a mobile sink node

nodes can be random (Shah et al. 2003), or pre-defined (Chakrabarti et al. 2003). Randomly moving around within a sensor field is not suitable for time-sensitive applications but can work well for large-scale applications such as forest fire monitoring, provided the sink node is equipped with a flight capability.

Predictable sink node mobility attempts to achieve optimum energy utilization by positioning a sink node at a pre-specified position at a specific time. Chakrabarti et al. (2003) used a mobile sink node that moves along a pre-defined path, and pulls data from the sensors when it arrives close to them, as shown in Fig. 6.4. Such a predictable mobility approach enables the collection of data to be done in a bounded transmission delay. The maximum power saving could be realised by invoking nodes only when they are scheduled to transfer their data.

6.4 Optimizing Locations of Static Multiple Sink Nodes

6.4.1 System Assumption

Here we restrict the WSN to a mesh topology, as shown in Fig. 6.5. For the sake of simplicity the following assumptions about the WSNs are made in establishing the system model:

- All sensor nodes are stationary and located in a bi-dimensional square grid composed of cells of the same size;
- Multiple sink nodes are fixed on the grid;
- Data transmission and reception are the major energy consuming activities;
- All sensor nodes have equal initial energy;
- The transmission range of each sensor node is fixed and equals to the distance between two adjacent nodes in the grid, i.e. a hop is of one grid cell side length.
- Sensor nodes communicate with the sink nodes by sending data via multiple hops along the shortest path;

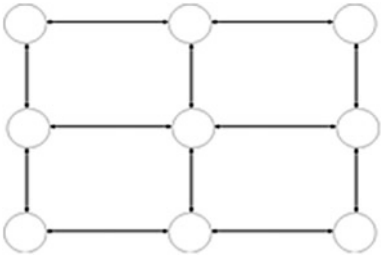


Fig. 6.5 WSN with a mesh topology

- The number of sink nodes is fixed and known in advance;
- Sink nodes can be located only at certain places in the grid, called feasible sites.
- The energy consumed in a senor node when transmitting a bit is constant, and is the same as the energy consumed for receiving one bit.

Formally, a sensor network is represented as a graph *G(V, E)*, where *V* are the vertices representing sensor nodes and *E* edges representing one-hop connectivity between two adjacent nodes *(i, j)*. Considering the mesh topology in the sensor network, a sensor node *i* can communicate directly with its four adjacent nodes (left, right, upper and lower). If the sensor node is not linked with the sink node through one-hop connectivity, then data packages generated at this sensor node have to be relayed through multiple hops in order to reach the sink node.

The notations and the mathematical formulation of power consumption at each sensor node here is adopted from Wang et al. (2005) work, and extend into a multiple static sink node case.

- e: Energy consumption coefficient for transmitting or receiving one bit (Joules/bit);
- e_0: Initial energy (Joules) of each node minus the threshold energy required for node operation;
- r: Rate at which data packets are generated (bits/s); for the homogeneous sensor nodes r is the same for all sensor nodes;
- C_i^k: Power consumption for receiving and transmitting packets at node I when the sink node is located at node k (Joules/s).
- z: Network lifetime (seconds).
- z_{ij}: Lifetime of Node i assigned to sink node k_j (seconds).

6.4.2 Simplified Routing Protocol

In the WSN with the mesh topology a simplified routing protocol is considered here as shown in Fig. 6.6.

- When a sensor node lies on the same horizontal or vertical line of the sink node, a unique shortest path between the sensor node and the sink node is taken.

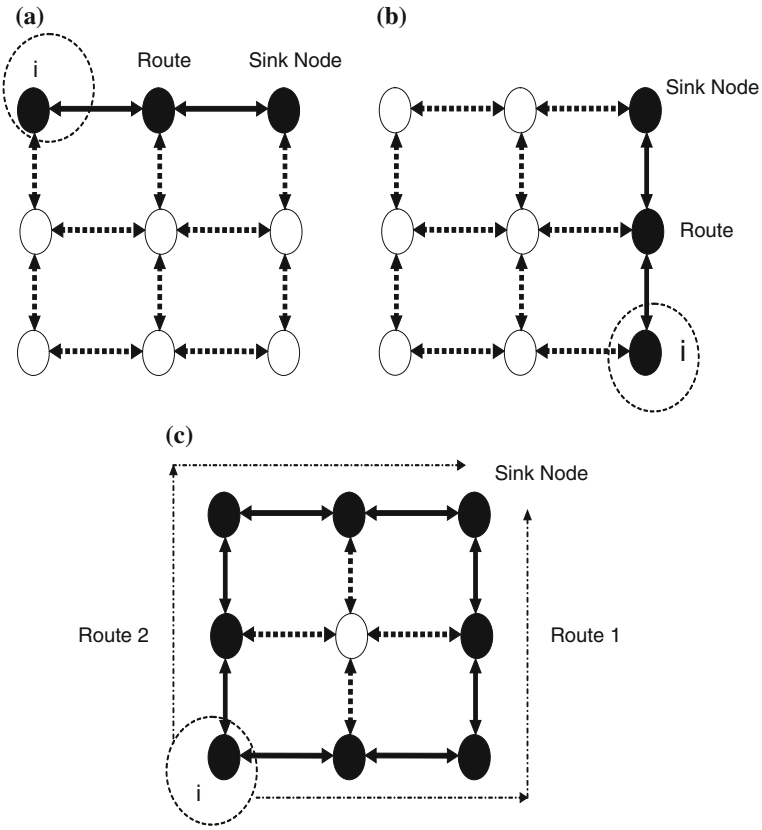


Fig. 6.6 Simplified routing protocol for a WSN with the mesh topology

- Otherwise, the two paths along the perimeter of the rectangle with the sensor node and the sink node as the opposite corners will be taken in equal proportions.

Three cases are illustrated in Fig. 6.6. Two hops in a unique route are required in Fig. 6.6a, b for transmitting a data package from node i to the sink node. Four hops in two symmetric routes are required in Fig. 6.6c.

6.4.3 Energy Consumption Model

Following (Wang et al. 2005) notation, each node's position is represented using the ordered pair of the node's column and row numbers $(x, y), x = 0, 1, \ldots, L-1$; $y = 0, 1, \ldots, L-1$. L is the numbers of column and row in the grid. A pair of

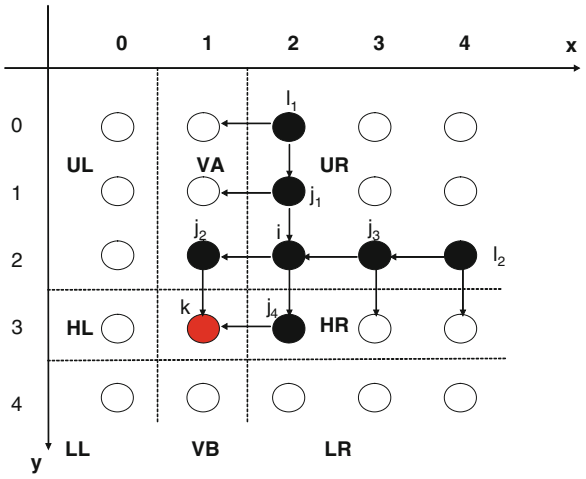


Fig. 6.7 Data flows received and transmitted at node i (Wang et al. 2005)

horizontal and vertical dotted lines is drawn enclosing the nodes associated with the row and the column of the sink node. These lines partition the sensor field into nine subsets as shown in Fig. 6.7: Upper Left (UL), Upper Right (UR), Lower Left (LL), Lower Right (LR), Vertical Above (VA), Vertical Below (VB), Horizontal Left (HL), Horizontal Right (HR), and the node K where the sink node is located.

Wang et al. (2005) gave the formulas shown in Eq. (6.1) to calculate the energy consumption at node i.

$$c_i^k = \begin{cases} er((x+1)(1+L)-1) & i \in HL \\ er((L-x)(1+L)-1) & i \in HR \\ er((y+1)(1+L)-1) & i \in VA \\ er((L-y)(1+L)-1) & i \in VB \\ er(1+x+y) & i \in UL \\ er(L-x+y) & i \in UR \\ er(L+x-y) & i \in LL \\ er(2L-x-y-1) & i \in LR \\ er & i \in k \end{cases} \tag{6.1}$$

Using node i in subset UR as an example, Node i transmits its own generated data packets to node j_2 and j_4, successively. Nodes j_2 and j_4 relay these packets to sink node k. In addition, node i receives half of the packets generated at nodes j_1 and j_3, and half of the packets generated at nodes l_1 and l_2. Then, node i retransmits the packets originated at nodes j_3 and l_2 to node j_2 and those originated at nodes j_1 and l_1 to node j_4. In summary, node i receives data packets at a rate $2r$ and transmits at a rate $3r$, having therefore power consumption $c_i^k = 5\,re$.

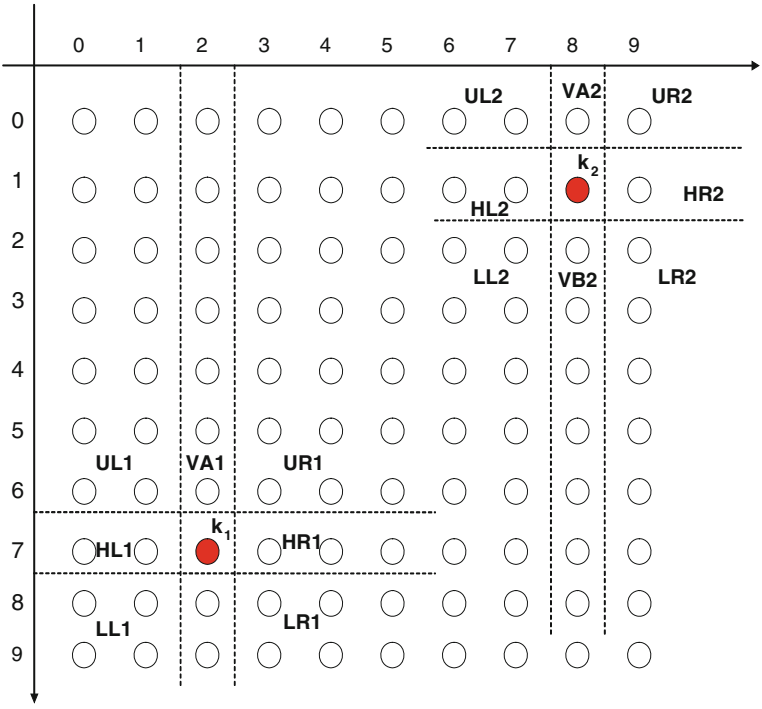


Fig. 6.8 Partition of a sensor field with two sink nodes

If there are multiple sink nodes in the sensor field, the sensor field can be further partitioned. Figure 6.8 illustrates the partition of the sensor field with two sink nodes k_1 and k_2. The subsets are denoted as UL1, UR1, LL1, LR1, VA1, VB1, HL1, HR1, and UL2, UR2, LL2, LR2, VA2, VB2, HL2, HR2. Depending on the locations of the two sink nodes some of the subsets will be overlapped. The formulas of calculating the energy consumption shown in Eq. (6.1) will still be suitable for the multiple sink node cases shown in Fig. 6.8 if all the sensor nodes are assigned to their nearest sink nodes and there is no double assignment i.e. each node is assigned to one and only one sink node.

6.4.4 Optimal Locations of Multiple Sink Nodes

Assume there are N sink nodes $k_1, k_2, \ldots, k_N$ for a sensor network. The lifetime of any sensor node i which is assigned to the sink node k_j is given by z_{ij}

$$z_{ij} = \frac{e_0}{c_i^{k_j}}$$

The lifetime of a sensor node i which is assigned to the nearest sink node is given as

$$\max\left(\frac{e_0}{c_i^{k_j}}\right), \quad j = 1, 2, \ldots, N$$

The lifetime of the sensor network can be defined as the time till the first sensor node in the sensor network runs out of battery capacity, i.e.

$$\min_i\left\{\max_j\left(\frac{e_0}{c_i^{k_j}}\right)\right\}$$

In order to maximize the lifetime of the sensor network the shortest lifetime of all the sensor nodes should be maximized by optimizing the locations of the multiple sink nodes. Therefore the objective of optimal locations of multiple sink nodes is given as (Yang 2006):

$$z = \max_{k_1, k_2, \ldots, k_N}\left\{\min_i\left\{\max_j\left(\frac{e_0}{c_i^{k_j}}\right)\right\}\right\}, \quad i = 0, 1, \ldots, L \tag{6.2}$$

The situation where sink nodes cannot be placed at any obstacle in the sensor field can be formulated into the above optimal problem with a constraint, where ϕ is the set of the obstacles.

$$k_1, k_2, \ldots, k_N \notin \Phi \tag{6.3}$$

6.5 Solving Optimal Location Problems

Many optimization approaches can be used to solve the optimal location problems represented in Eqs. (6.1)–(6.3). Because of the nature of multi-variables in the above optimization, most of the evolutionary computation algorithms can be applied to solve the location problems (Yang 2006; Alageswaran et al. 2012). Yang (2006) represents the optimal location problem as a GA search problem. In detail the following three definitions are implemented:

- A chromosome representation of the problem which is amenable symbolic manipulation;
- A fitness function defined in terms of this representation;
- A set of manipulation operators such as crossover, mutation, and reproduction.

The variables to be optimized in the objective function shown in Eq. (6.2) are the coordinates of N sink nodes, which are N pairs of integer coordinates

$$(i_a, j_a), \quad a = (1, 2, \ldots, N) \tag{6.4}$$

The chromosome is expressed as an integer string with a length $2N$ (the double of the number of the sink nodes) as bellow:

$$i_1 \quad j_1 \quad i_2 \quad j_2 \quad \ldots \quad \ldots \quad i_N \quad j_N \tag{6.5}$$

where

$$(i_a, j_a) \neq (i_b, j_b), \ if \quad a \neq b \tag{6.6}$$

$$(i_a, j_a) \notin \phi, \ a = (1, 2, \ldots N) \tag{6.7}$$

Equation (6.6) represents that two different sink nodes cannot be located in a same location. Equation (6.7) means that sink nodes cannot be located in the same location as the set of the obstacles.

The fitness function in the GA is chosen from the objective function shown in Eq. (6.2).

$$f_{\text{fitness}} = \min_i \left\{ \max_j \left(\frac{e_0}{c_i^{k_j}} \right) \right\} \tag{6.8}$$

The desired value of the fitness function will be the maximum value of Eq. (6.8) for all possible combinations of N sink nodes. The three operators of GA, crossover, mutation, and reproduction are applied to update the chromosome shown in Eq. (6.5) with satisfaction of the constraints represented in Eqs. (6.6) and (6.7).

A simple simulation has been done in this study. Two sink nodes are designed for a 8 × 8 sensor network with three prohibited locations (1, 1), (5, 5), and (6, 6). In the initial stage, these two sink nodes are selected randomly, so GA starts with several chromosomes that describe a number of random solutions to the optimal location problem. The parameters of GA are chosen, as the probability of mutation being 0.08, the probability of crossover 0.6, and the size of the population 20. The values of the parameters of the 8 × 8 sensor network are chosen as $r = 1$ bit/s, $e = 0.62$ μJ/bit, $e_0 = 1.35$ J. The search results illustrate that the upper left corner (0, 0) and the bottom right corner (7, 7) are the optimal locations for the two sink nodes. This result is consistent with the results in Wang et al. (2005) work for mobile sink nodes.

6.6 Conclusions

In this chapter we only consider the optimal problem of the multiple sink node locations with respect to a simplified routing algorithm. If any other routing algorithm is employed in the sensor network the system model is likely to be much more complex. The assumptions we made in Sect. 6.4.1 are also simplified. In particularly, we assume that all sensor nodes are stationary and located in a bi-dimensional square grid composed of the same size-cells and the transmission

range of each sensor node is fixed and equals to the distance between two adjacent nodes in the grid. These two assumptions make the system model the simplest, but are unlikely to be realistic. As the sensor nodes may be randomly distributed and they may have different communication ranges. We also assume that the number of the sink nodes is fixed and known in advance. If this assumption is untrue the number of the sink nodes has to be included as an additional optimal variable in the optimization. Nevertheless, the methodology introduced and the formulizations of the search problem are, in general, applicable to any complex sink node positioning problems.

References

Akkaya, K., Younis, M., Bangad, M.: Sink repositioning for enhanced performance in wireless sensor networks. Comput. Netw. **49**(4), 512–534 (2005)

Akkaya, K., Younis, M., Youssef, W.: Positioning of base stations in wireless sensor networks. IEEE Commun. Mag. **45**(4), 96–102 (2007)

Alageswaran, R., Usha, R., Gayathridevi, R., Kiruthika, G.: Design and implementation of dynamic sink node placement using particle swarm optimization for life time maximization of WSN applications. International conference on advances in engineering, science and management. Nagapattinam, Tamil Nadu, pp. 552–555 (2012)

Chakrabarti, A., Sabharwal, A.,and Aazhang, B.: Using predictable observer mobility for power efficient design of sensor networks. Information Processing in Sensor Networks, Lecture Notes in Computer Science, vol. 2634, pp. 129–145 (2003)

Heinzelman, W.R., Chandrakasan, A., Balakrishnan, H.: Energy-efficient communication protocol for wireless microsensor networks. In: Proceedings of the 33rd annual Hawaii International Conference on System Sciences, p. 10, (2000)

Oyman, E. I., Ersoy, C.: 2004 IEEE international conference on communications, vol. 6, pp. 3663–3667

Pan, J., Cai, L., Hou, Y.T., Shi, Y., Shen, S.X.: Optimal base-station locations in two-tiered wireless sensor networks. IEEE Trans. Mob. Comput. **4**(5), 458–473 (2005)

Shah, R.C., Roy, S., Jain, S., Brunette, W.: Data MULEs: modeling and analysis of a three-tier architecture for sparse sensor networks,". Ad Hoc Netw. **1**(9), 215–233 (2003)

Sohraby, K., Minoli, D., Znati, T.F.: Wireless sensor networks: technology, protocols, and applications. Wiley, New York (2007)

Vincze, Z., Vass, D., Vida, R., and Vidács, A.: Adaptive sink mobility in event-driven clustered single-hop wireless sensor networks. In: Proceedings of the 6th International Network Conference, Plymouth, UK, pp. 315–322 (2006)

Wang, Z. M., Basagni, S., Melachrinoudis, E., Petrioli, C.: Exploiting sink mobility for maximizing sensor networks lifetime. In: Proceedings of the 38th Annual Hawaii International Conference on System Sciences, p. 287a (2005)

Yang, L.: Determining sink node locations in wireless sensor networks. IEEE international conference on systems, man and cybernetics, pp. 3400–3404 (2006)

Chapter 7
Interference of WSNs with IEEE 802.11b Systems

Keywords Interference · Energy detection · Interference mitigation · IEEE 802.15.4 · IEEE 802.11b · WiFi

7.1 Introduction

The rapid development of wireless technology has brought significant changes for many applications by removing the restriction caused by wires. Unfortunately, wireless interference seriously restricts the development of wireless devices; as the communication medium, air, is open to potential wireless interferers and consequently such systems, unlike wired system, have no effective protection against interference. Popular wireless products working in the 2.4 GHz ISM band include Wi-Fi, Bluetooth, ZigBee, microwave ovens, cordless phones, etc. (Thonet et al. 2008). Since these technologies are mainly developed for consumer electronics, it is common for users to have two or more of these products simultaneously in use. As a result, their performance may be affected, if multiple wireless products are working in the same frequency area.

As the standard for low data rate, low cost wireless solutions, the IEEE 802.15.4 specification is widely applied in the construction of wireless personal area network, such as WSNs. The IEEE 802.15.4 standard provides the features necessary for the construction of inexpensive wireless connectivity enabling monitoring and control functions in the areas of residential, commercial and industrial applications. Because of the mobile and ubiquitous deployment of WSNs, there are many scenarios where different wireless systems operate in the same location at the same time. Potentially, the possibility for the wireless links established between IEEE 802.15.4 WSNs to suffer interference will be considerably increased. The characteristics of the 802.15.4 WSNs, includes low transmission power (typically 1 mW) and a relative narrow bandwidth (2 MHz for each channel), making its receiver subject to interference by other powerful wireless system. In many practical situations and scenarios, IEEE 802.15.4 WSNs and IEEE 802.11b WiFi systems are simultaneously

S.-H. Yang, *Wireless Sensor Networks*, Signals and Communication Technology,
DOI: 10.1007/978-1-4471-5505-8_7,

operating in the same vicinity. New development in theoretical analysis and some primary tests of wireless systems identifies the factors that can cause interference with the operations of WSNs. Commonly used interference mitigation strategies include keeping a physical and frequency separations between the victim systems and interferers, employing effective routing protocols, and allowing dynamic frequency agility etc. This chapter explores the definition, causes and effective mitigation of wireless interference in the operations of WSNs.

7.2 Wireless Coexistence and Interference in WSNs

Coexistence is defined as "the ability of one system to perform a task in a given shared environment where other systems may or may not be using the same rules" (IEEE 2000). For example, if a ZigBee based home automation system is to be deployed in a home environment, a main deployment issue will be ensuring the coexistence of the ZigBee system with any home WiFi system. For a large-scale WSN deployed for forest fire detection, environment or traffic monitoring etc., the coexistence of the WSN and other wireless systems will ensure the satisfactory performance of the WSN.

Interference in the concept of wireless communications usually refers to one of the following two definitions: (1) multiple (i.e. more than two) simultaneous packet transmissions causing packets to collide at the receiver, (2) physical factors in the radio propagation channel (Golmie 2006).

If multiple wireless signals simultaneously arrive at the receiver, the receiver will be unable to abstract any useful information since the desired signal and the interfering signal overlap each other.

The physical factor in the radio propagation channel is another challenge to the operations of the wireless communication systems. Various physical impediments, such as multipath propagation, should be taken into consideration when the radio system is being designed. Multipath propagation means that a transmitted signal can reach the receiver via several different paths (e.g. reflections from house, windows, or walls). Figure 7.1 shows an example of multipath propagation.

In Fig. 7.1, the 'signal on direct path' lying between the transmitter and receiver is the desired wireless signal path, also called as 'Line of Sight (LoS) connection'. If the obstacle (e.g. the wall in Fig. 7.1) exists in the vicinity of the transmitter, the

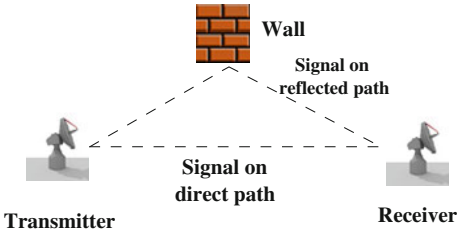


Fig. 7.1 Example of multipath propagation

radio signal could be reflected and reach the receiver via a 'reflected path'. Since a simple receiver cannot distinguish multipath signals, it just adds them up. Consequently, the 'signal on direct path' and 'signal on reflected path' interfere with each other (Molisch 2005).

In the context of this chapter, the interference study is mainly about the discussion and evaluation of the effect on a IEEE 802.15.4 WSNs caused by other wireless systems, particularly by IEEE 802.11b WiFi systems, where simultaneous packet transmissions are the main concern.

7.3 Performance Metrics

In IEEE 802.15.4 WSNs, the performance metric used to evaluate the wireless communication can be separated into two parts: Physical (PHY) layer and Media access control (MAC) layer. These metrics are normally used to measure the level of interference.

7.3.1 PHY Layer Performance Measures

The commonly used metric in the PHY layer of a wireless system is the signal-to-noise ratio (SNR), which denotes the ratio of the average signal power to the average noise power in decibels (dB). A radio system must transmit a modulated signal at a certain frequency. Any successful reception at the receiver side can only be achieved if the receiver keeps listening at the same frequency. If the SNR is less than a defined threshold, which means the level of noise is greater than the useful signal, the receiver will fail to obtain the desired signal (Chandra et al. 2007). Another important metric is the bit error rate (BER), which expresses the number of incorrectly received bits on the receiver side against the total number of bits transferred during a transmission. Because of the use of different modulation schemes, the requirements of SNR and BER for achieving an acceptable level of performance are different in certain wireless systems. Figure 7.2 illustrates the simulation results of BER at different SNR for various wireless standards.

In Fig. 7.2, multiple wireless techniques are simulated with various SNR level. A general tendency is that a low bit error rate can be obtained when SNR increases. For example, if the IEEE 802.15.4 system is required to achieve a bit error rate of 1.0E–09, the corresponding SNR should be greater than 3 dB. In other words, once the noise rises to a high level, the number of correctly recovered bit will accordingly decrease. This usually happens when an IEEE 802.15.4 signal is being interfered with by a powerful IEEE 802.11b (i.e. Wi-Fi) signal.

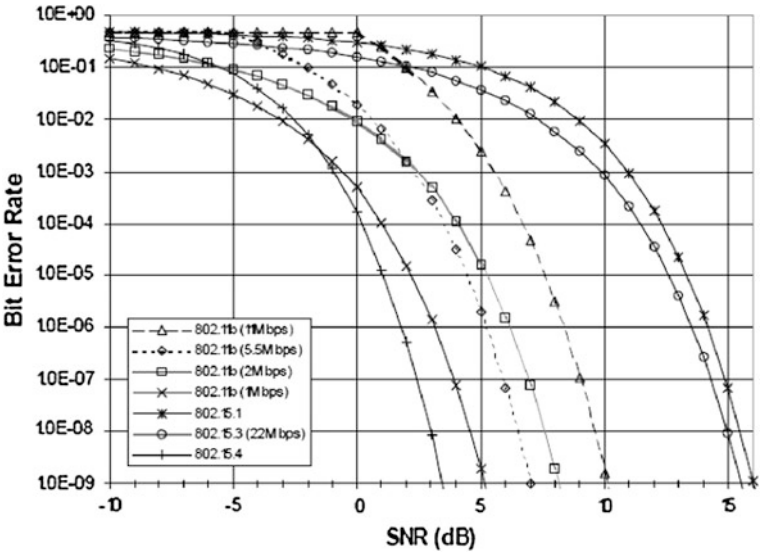


Fig. 7.2 BER via SNR simulation results for IEEE 802.11b and IEEE 802.15 *asterisk* (IEEE Std 802.15.4 2003)

7.3.2 MAC Layer Performance Measures

Although it is important for the developers to understand the wireless communication performance in the PHY layer metrics, such as SNR and BER, when the system is under interference, measurements of these metrics are difficult to obtain without specific devices. More explicit tests are mostly used for evaluation. For example, the Packet Error Rate (PER) is used to describe how sustainable a wireless system can be in certain environments. This type of measurement can be implemented at the MAC layer.

The MAC layer consists of rules that regulate the mechanism of channel access and sharing. It is also responsible for assembling and disassembling data packets passed through the PHY layer. In order to analyse the effect of interference on WSNs at the system level, the metrics of PER, transmission delay and throughput should be included (Shin et al. 2007).

Packet Error Rate: Packet error rate is the percentage of packets lost, expressed as the ratio between the packets, which fail to be received by the receiver against all the packets generated by the source node (Cuomo et al. 2007). One of the consequences caused by interference in WSNs is an increase in the packet error rate. It is also the most important metric, which can be improved by the means of anti-interference design.

Delay and Throughput: The throughput is the amount of data transferred from one station to another station during a specified period of time (Shin et al. 2007). The

occurrence of interference in WSNs will obviously cause an increase in the transmission delay and the reduction of throughput, which could be improved by an effective anti-interference design at the system level.

7.4 Coexistence Mechanism of IEEE 802.15.4

In the design of the IEEE 802.15.4 standard, the 802.15.4 task group cooperated with other Coexistence Task Groups, such as 802.15.2TM to ensure the standard's coexistence capability with other wireless devices (IEEE Std 802.15.4 2003). As a result, the IEEE 802.15.4 standard provides support for coexistence at both of the PHY layer and MAC layer. At the PHY layer, Direct Sequence Spread Spectrum (DSSS) is adopted. Frequency Division Multiple Access (FDMA) and Carrier Sense Multiple Access with Collision Avoidance (CSMA-CA) are adopted at the MAC layer.

7.4.1 Direct Sequence Spread Spectrum

The license-free industrial scientific and medical (ISM) bands are crucial to the burgeoning market for wireless embedded technology. A short list of possible users includes: IEEE 802.11b networks, IEEE 802.11g networks, IEEE 802.11n networks, Bluetooth Pico-Nets, IEEE 802.15.4 networks, cordless phones, home monitoring cameras, microwave ovens and WiMax networks (ZigBee 2007). Accordingly, any of these has the potential to interfere with other systems. The IEEE 802.15.4 standard adopts the technology of direct sequence spread spectrum (DSSS) to increase the opportunities for coexistence with multiple ISM band users.

The 'spread spectrum' modulation technique is designed to promote a radio system's capability of coexistence and robustness in the presence of interference. The idea of spread spectrum is to spread the transmission over a large bandwidth. The spread spectrum approach appeared originally in military applications. It is used because of a number of attractive properties, e.g. anti-jamming performance, low probability of interception and multiple access communications (Fakatselis 1996). In general conditions, even though the center frequencies of narrow band signals (signals that encode and transmit information by using a small bandwidth) are not exactly the same, it is still possible to have signal collision and data packet loss. The frequency allocation is restricted and controlled by regulators such as the U.S Federal Communications Commission. However, there is no compulsory requirement in the ISM bands. Thus the wireless interference could happen to any wireless system operating with narrowband signals (IEEE Std 802.15.4 2003). Figure 7.3 illustrates collisions between two narrowband signals.

In Fig. 7.3, two narrowband signals collide with each other. Since the main bodies of these two signals overlap, the information carried by the overlapping

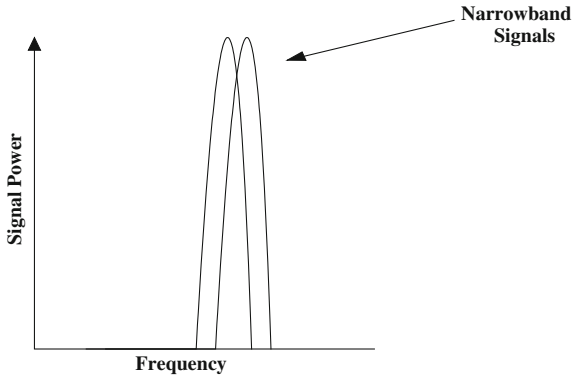


Fig. 7.3 Narrow band signals (ZigBee 2007)

parts could be corrupted due to interference. To avoid uncontrollable interference between narrowband signals, the effective area of the overlapping parts should be limited. The method of 'spread spectrum' was designed to solve this problem. Figure 7.4 illustrates the principle of 'spread spectrum'.

In Fig. 7.4, the two narrowband signals denote narrowband interference signal (with the solid line) and narrowband desired signal (dashed line) respectively. The purpose of the 'spread spectrum' approach is to use more bandwidth to convey the bit information originally carried by the narrowband desired signal. After spreading, only a small part of the desired original narrowband signal is affected by the narrowband interference signal (indicated by the fraction of the grey cube in Fig. 7.4). When the spread signal reaches the receiver, the system will abstract the useful signals from the 'spread spectrum'.

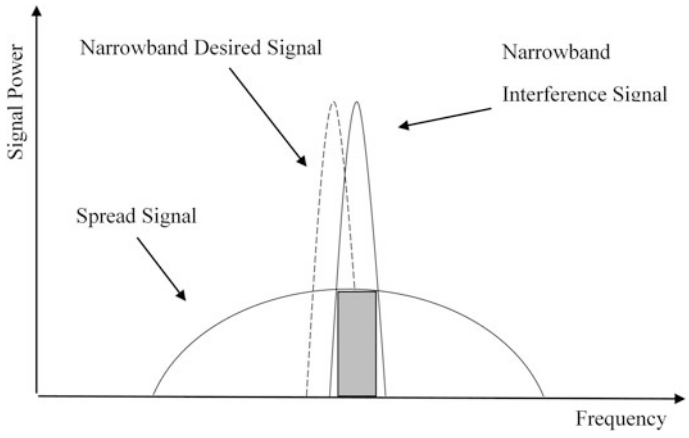


Fig. 7.4 Principle of spread spectrum

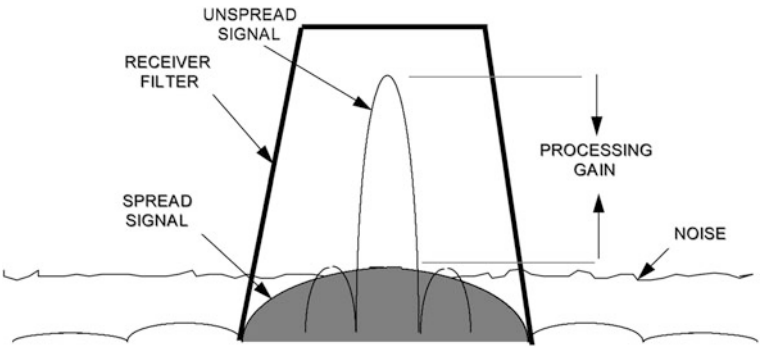


Fig. 7.5 Direct spread spectrum at the receiver (Fakatselis 1998)

In Fig. 7.5, the spread spectrum signal is recovered into the form of an 'unspread signal' after passing through the receiver filter, whose main function is to make the receiver only sensitive to signals on the specified frequency. Although some parts of the narrowband interference signal would also pass through the receiver filter, it is highly probable that the desired narrowband signal is obtained correctly, since only a small portion of the spread signal is affected by the interference. Theoretically, if more bandwidth is used to convey the spread signal, the more interference can be tolerated. A common measure used in spread spectrum is the processing gain G (Golmie 2006):

$$G = 10 \log_{10}(r_c / r_b) \tag{7.1}$$

where r_b and r_c denote the bit rate and chip rate respectively. In a DSSS system, each bit, before transmission, is broken down to a pattern of bits called a 'chip'.

A chip is generated by performing an XOR (Exclusive-OR) operation on each bit with a pseudo random code. The output of the XOR operation, the chip bit, is then modulated and transmitted. The receiver uses the same pseudo random code to decode the original data. The benefit of processing gain is that the pseudo random code spreads the transmitted narrowband desired signal and makes it less susceptible to the narrowband interference signal within the employed bandwidth. The processing gain can be thought of as the ratio of signal to interference at the receiver after the dispreading operation (Fig. 7.5). For example, a wireless system requires 10 dB E_b/N_0 (it is a normalized version of SNR, where E_b denotes the energy per bit, N_0 denotes the noise power spectral density) to achieve a satisfactory performance with an acceptable BER. If the system process gain is 4 dB, the system can maintain the required performance when the desired signal has 6 dB (10–4 dB) over the interference. In an IEEE 802.15.4 system working in the 2.4 GHz band, the chip rate is 2,000 kchip/s, and the bit rate is 250 kb/s. Therefore, the processing gain for the IEEE 802.15.4 device is 9 dB according to Eq. (7.1). The use of DSSS in IEEE 802.15.4 systems adds the capability to effectively coexist with a narrowband wireless communication system (e.g. Bluetooth) whose bandwidth is less than the bandwidth of IEEE 802.15.4 signals (IEEE Std 802.15.4 2003).

Although the radio system design helps the IEEE 802.15.4 device improve the capability of tolerating the interference to a certain extent (i.e. when the interference power is less than the desired signal power), it is impossible to have all the interference overcome, particularly when the interfering signal is far more powerful than the desired signal.

7.4.2 Frequency Division Multiple Access

The use of frequency division multiple access (FDMA) in an IEEE 802.15.4 system divides the 2.4 GHz ISM band into 16 non-overlapping channels and is depicted in Fig. 7.6.

In Fig. 7.6, a total of 16 channels are defined in the 2.4 GHz band, starting from 2.405 MHz and ending at 2.480 MHz. Each IEEE 802.15.4 channel is 2 MHz wide and the channels are 5 MHz apart. The setting of non-overlapping channels allows multiple IEEE 802.15.4 users to operate separately on different frequencies without worrying about overlapping each other. Accordingly, if the interfering radio frequency is close to the IEEE 802.15.4 communication channel currently being used, the IEEE 802.15.4 system can switch to another channel whose center frequency is away from any interfering energy. Some other wireless systems employ the same or similar mechanisms to make use of the radio frequency. For example, IEEE 802.11b/g technique utilizes the same FDMA mechanism to define 14 communication channels in the 2.4 GHz band. Bluetooth (i.e. IEEE 802.15.1) divides the scientific band (S-Band) into 79 channels of 1 MHz each and utilizes Frequency Hopping (FH) to achieve wireless communication. This system constantly hops among the defined channels. The order of the channels or hop sequence used by the transmitter is predefined and has previously been communicated to the receiver. The time a transmitter spends in each channel is less than 400 ms and the maximum power for the transmitter should not exceed 1 W. As a consequence of FH, the Bluetooth devices can easily avoid the effect of interference by regularly switching channels.

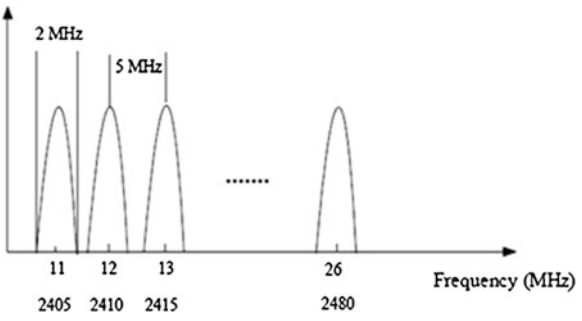


Fig. 7.6 Channel allocation of IEEE 802.15.4 in 2.4 GHz band

7.4.3 Carrier Sense Multiple Access with Collision Avoidance

Since IEEE 802.15.4 devices are likely to coexist with different wireless systems, the IEEE 802.15.4 MAC protocol adopts the use of Carrier Sense Multiple Access with Collision Avoidance (CSMA-CA) to deal with situation when unpredictable interference, or signal collision happens whilst the IEEE 802.15.4 devices are in communication. The CSMA-CA technique has been widely used for other network communications, such as Ethernet and Wi-Fi. It employs a simple 'listen before you talk' strategy. Before initiating a wireless transmission, a device listens on the channel and implements channel assessment. If the channel is idle, the transmission will proceed. If the channel is busy, the device will wait for a random interval before assessing the channel again. With the increment of channel assessment failure, the wait interval increases exponentially in order to avoid interference (ZigBee 2007).

All the mechanisms mentioned above are useful in ensuring the coexistence of the IEEE 802.15.4 based WSNs, however they become effective in different ways. The DSSS technique helps the radio transmission improve the possibility of being successfully processed by the receiver, the FDMA provides the IEEE 802.15.4 system with greater chance to coexist with other wireless system by moving to different radio frequency channel, and CSMA-CA aims to deal with the signal collisions before actually propagating the radio signal. When the WSNs are employed in practical applications, the various situations caused by different scenarios may occur, which therefore require greater consideration in designing interference mitigation strategies.

7.5 Mitigating Interference Between IEEE 802.11b and IEEE 802.15.4

The existing studies suggest that interference occurs only if two conditions are satisfied: a small or zero offset of the radio frequency, and powerful interfering energy. Frequency offset here means the difference between the center frequencies of two associated communication channels.

7.5.1 Frequency Offset

Both IEEE 802.15.4 and IEEE 802.11b devices work in the specified communication channels. Due to the limited range of the 2.4 GHz ISM band, it is possible to have both of the two wireless systems working at a close frequency. The radio transmission power usually gathers around the center frequency of the selected

channel, which therefore can easily cause interference if the frequency offset is small. Figure 7.7 and Table 7.1 show the channel allocations for both IEEE 802.15.4 and IEEE 802.11b in the 2.4 GHz ISM band. The IEEE 802.11b has 14 channels whose center frequency ranges from 2.412 to 2.473 MHz. Each channel is 22 MHz wide and has a 5 MHz separation from the adjacent channels. Due to the wide bandwidth, many IEEE 802.11b communication channels overlap with each other. In order to ensure multiple IEEE 802.11b networks can simultaneously work in the same area, the frequency spacing between any IEEE 802.11b communication channels must be at least 30 MHz (So 2004). Therefore, the IEEE 802.11 standard recommends that if multiple IEEE 802.11b networks are required to run in close vicinity, three non-overlapping channels can be employed. The settings of these three non-overlapping channels are not the same in different geographical regions: channels 1, 6, 11 are recommended in China and North American while channels 1, 7, 13 are selected in European (IEEE Std 802.11 2007), as shown in Fig. 7.7.

On the other hand, the IEEE 802.15.4 has a total of 16 channels in the 2.4 GHz band. Each channel is 5 MHz apart and has a frequency range of 2 MHz. The channel number starts at 11 as the IEEE 802.11b PHY layer offers one channel in

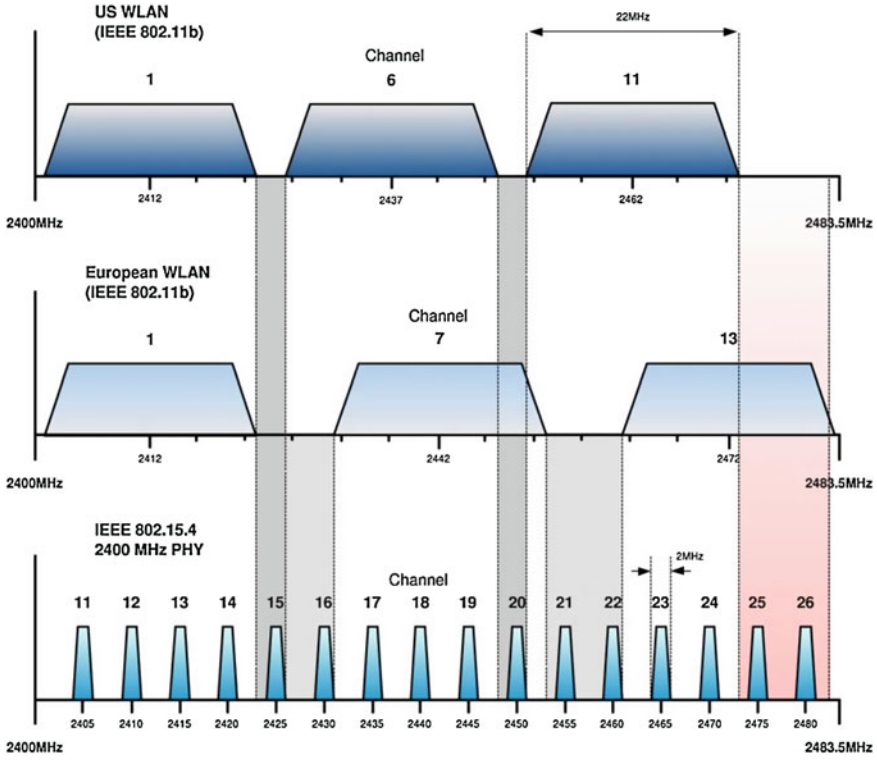


Fig. 7.7 Non-overlapping IEEE 802.11b and IEEE 802.15.4 channel allocation

the 868 MHz band, and ten in the 915 MHz band. The frequency range of each IEEE 802.11b channel is overlapped with the frequency ranges of four different IEEE 802.15.4 channels. For example, IEEE 802.11b channel 1 has the frequency range between 2.401 and 2.423 MHz, which includes the frequency ranges for IEEE 802.15.4 for channel 11 to channel 14. IEEE 802.11b channel 1 can cause radio interference to IEEE 802.15.4 channels 11, 12, 13, and 14 when it is operating in proximity.

As shown in Fig. 7.7 and Table 7.1, most of the IEEE 802.15.4 communication channels overlap with the Wi-Fi communication channels except for a few channels illustrated in grey. These channels provide a simple means for coexistence. For example, the IEEE 802.15.4 channels 15, 20, 25, and 26 use a frequency range outside of the frequency range for 802.11b channels 1, 6 and 11. Therefore, those channels can be used in certain environments, in which interference from 802.11b is expected. However, there are many situations, in which it is necessary to use more channels. These IEEE 802.11.15.4 channels can experience interference.

Even though both IEEE 802.11b and IEEE 802.15.4 employ the technique of DSSS, the advantage of 'spread spectrum' does not have any obvious effect if the center frequencies of the IEEE 802.11b system and the IEEE 802.15.4 system are close to each other. Additionally, the maximum transmitting power of an IEEE 802.11b device can achieve 20 dBm (equivalent to 100 mW), which is much higher than the IEEE 802.15.4 devices (i.e. 1 mW). Once the IEEE 802.11b signals affect the IEEE 802.15.4 receiver, the relative higher output power will contribute to the noise part of the SNR. Figure 7.8 illustrates the transmit spectrum mask of the IEEE 802.11b signal.

Table 7.1 IEEE 802.15.4 channel allocation, and IEEE 802.11b channel allocation in the 2.4 GHz ISM band

IEEE 802.11b		IEEE 802.15.4	
Channel	Frequency (GHz)	Channel	Frequency (GHz)
1	2.401–2.423	11	2.405
2	2.406–2.428	12	2.410
3	2.411–2.433	13	2.415
4	2.416–2.438	14	2.420
5	2.421–2.443	15	2.425
6	2.426–2.448	16	2.430
7	2.431–2.453	17	2.435
8	2.436–2.458	18	2.440
9	2.441–2.463	19	2.445
10	2.446–2.468	20	2.450
11	2.451–2.473	21	2.455
12	2.456–2.478	22	2.460
13	2.461–2.483	23	2.465
14	2.466–2.488	24	2.470
		25	2.475
		26	2.480

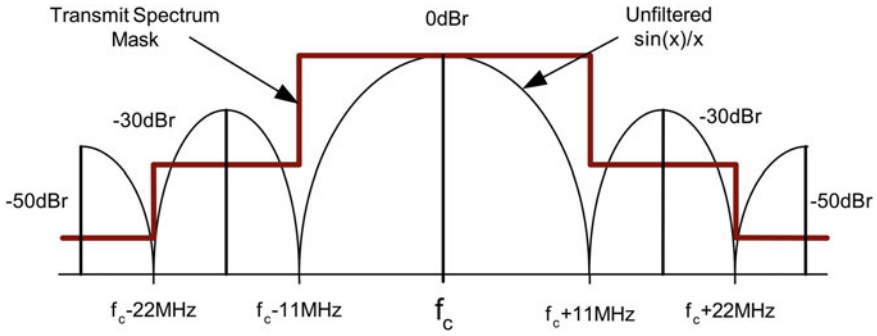


Fig. 7.8 Transmit spectrum mask of IEEE 802.11b (IEEE Std 802.11 2007)

In Fig. 7.8, the power spectrum concentrates on the centre frequency of the selected IEEE 802.11b communication channel. The increment of separation from the centre frequency causes the power contained in the IEEE 802.11b signal to decrease. In (Shin et al. 2007), a simulation was carried out to study the relationship between the interference and frequency offset. The result states that the IEEE 802.15.4 system can achieve an acceptable performance (i.e. PER less than 1 %) when the frequency offset between the centre frequencies of these two systems is larger than 7 MHz. Another similar study was carried out by Petrova et al. (2006). The measurements were made for different frequency offsets between the central frequencies of the IEEE 802.15.4 communication channels and the IEEE 802.11b communication channels. The following conclusions were stated: "Our measurement showed that there should be at least 7 MHz offset between the operational frequencies for a satisfactory performance of the IEEE 802.15.4".

7.5.2 Interfering Energy and Physical Separation

The interfering energy is the major factor that causes the reception failure of the receiver of the wireless node. As mentioned before, powerful interfering energy can easily make the receiver unable to recognize the desired signal, unless the strength of the interfering energy is less than an acceptable level.

The effective range of interference in wireless communication is mainly determined by the physical distance between the interferer's transmitter and the victim's receiver. Two parameters are usually used to describe the performance of a radio system: output power and receiver sensitivity.

- The output power indicates the energy level of the output signal sent from the transmitter.
- The receiver sensitivity denotes the minimum energy level of the radio signal, which is detectable on the receiver.

A receiver can recover the radio signal if the remaining energy level of the output signal when it reaches the receiver is greater than the receiver sensitivity. After propagation, the energy level of the output signal will attenuate with an increase in the distance that the signal travels. When the interferer's transmitter and victim's receiver are separated by a certain physical distance, the interfering signal strength reaching the victim's receiver will be reduced. If the remaining energy level of the interfering signal is less than allowed noise level, the victim's receiver should be able to function normally. This signal strength reduction is classified as path loss. Path loss means the ratio of the total radiated power from a transmitter antenna times the numerical gain of the antenna in the direction of the receiver to the power available at the receiver antenna (Chandra et al. 2007). According to different environment conditions, the path loss can be described by different models. The basic model is when the path loss is applied to the simplest possible scenario: a transmitter and a receiver in a free space. The model (Yilmaz 2002) is given as:

$$L_p = 20\log d + 20\log f + 32.45 \tag{7.2}$$

where:

L_p path loss in dB
d distance between the transmitter and the receiver in km
f frequency of transmission in MHz

In the environment of free space, assuming (a) the output power of interfering signal is 0 dBm (i.e. 1 mW), the sensitivity of the victim receiver is −82 dBm; (b) the interfering signal and victim receiver work on 2.410 and 2.430 MHz respectively; and (c) if the interfering power falling on the victim's receiver is less than −82 dBm, the interference effect can be ignored. Therefore, the allowed path loss on the interfering signal is $0 - (-82\text{dBm}) = 82$ dBm. According to Eq. (7.2), the distance d is obtained as 125 m, which can be thought as a safe distance for the victim to avoid interference (Rodriguez 2005).

In a practical environment, the calculation of path loss will be affected by many factors, e.g. antenna, building structure, and street layout et al. Shin et al. (2007) analysed the interference in the IEEE 802.15.4 system caused by an IEEE 802.11b transmitter using a simple indoor path loss model as follows:

$$L_p(d) = \begin{cases} 20\log_{10}\left(\frac{4\pi d}{\lambda}\right) & d \le d_0 \\ 20\log_{10}\left(\frac{4\pi d}{\lambda}\right) + 10n\log_{10}\left(\frac{d}{d_0}\right) & d > d_0 \end{cases} \tag{7.3}$$

where d denotes the distance between the transmitter and receiver in metres, d_0 denotes the length of line-of-sight in metres, which is normally 8 m. The parameter λ is equal to c/f_c, where c is the velocity of light and f_c is the carrier frequency in MHz; n denotes path loss exponent which is 3.4 in an indoor

environment for a distance over 8 m (Golmie et al. 2005). For both IEEE 802.15.4 and IEEE 802.11b systems, if the output power is fixed, the received power on the receiver is obtained as follows (Shin et al. 2007):

$$P_R = P_T \times 10^{\frac{-L_P(d)}{10}} \tag{7.4}$$

where :

P_T Transmission power measured on the transmitter in mW
P_R Received power measured on the receiver in mW
$L_P(d)$ The path loss of transmission power after travelling distance d in dB

Equation (7.4) also shows that transmission power degrades with an increase in the communication range. The received powers of both the desired signal (IEEE 802.15.4) and the interferer (IEEE 802.11b) can be adjusted by changing the physical separation between the IEEE 802.15.4 receiver and the IEEE 802.11b transmitter. On the one hand, the IEEE 802.11b transmission power would be harmless to an IEEE 802.15.4 receiver if the physical distance between them is kept large enough. On the other hand, if the received power of the IEEE 802.15.4 signal (desired signal) is greater than the receiver's sensitivity, and the signal to interference and noise ratio (SINR) at the receiver is greater than its threshold value, the physical separation at distance d between the IEEE 802.15.4 receiver and the IEEE 802.11b transmitter is safe, otherwise a further physical separation is required in the deployment. SINR is the extension of SNR and can be determined by

$$SINR = 10\log_{10}\left(\frac{P_R^s}{P_R^i + P_R^n}\right) \tag{7.5}$$

where P_R^s, P_R^i, and P_R^n denote the power of the desired signal, the interference power and noise power on the receiver, respectively.

Shin et al. (2007) carried out a simulation with assumptions that the output power of the IEEE 802.11b (interferer) and the IEEE 802.15.4 system (victim) are 30 and 1 mW. The IEEE 802.11b system works at 11 Mbps with a 1,500 bytes payload size. The IEEE 802.15.4 works at 250 kbps with a 105 bytes payload size. The offset between the centre frequencies of the IEEE 802.11b and the IEEE 802.15.4 systems was 2 MHz. Consideration was also given to the non-uniform power spectral density distribution of the IEEE 802.11b signal. The simulation results stated that the packet error rate of the IEEE 802.15.4 was smaller than 10^{-5} when the distance between the IEEE 802.15.4 receiver and the IEEE 802.11b transmitter was greater than 8 m.

7.5.3 Recommendations Made in IEEE 802.15.4

The IEEE 802.15.4 task group has developed general guidance for IEEE 802.15.4 systems to coexist with other wireless devices operating in an unlicensed frequency band. The coexistence mechanisms provided in the IEEE 802.15.4 standard include clear channel assessment (CCA), dynamic channel selection, modulation, energy detection (ED) and link quality indication (LQI), low duty cycle, low transmit power, and channel alignment (IEEE Std 802.15.4 2003).

- CCA: The CCA is part of the CSMA-CA mechanism. There are three CCA methods available for use: energy detection over a certain threshold, detection of a signal with IEEE 802.15.4 characteristics, or a combination of these two methods. The IEEE 802.15.4 PHY can choose one of the CCA methods to implement channel assessment for detecting whether any other device is occupying the channel.
- Dynamic Channel Selection: The IEEE 802.15.4 specification does not support direct frequency hopping. However, users can specify a certain mechanism in applications, to manually switch to a suitable communication channel when interference is detected on the current one.
- Modulation, ED, and LQI: The employed modulation scheme is Offset Quadrature Phase Shift Keying (O-QPSK), which is a power-efficient modulation method that achieves a low signal-to-noise ratio. ED and LQI are two measurement functions. ED is used to detect the energy level within an IEEE 802.15.4 channel. Meanwhile, it can provide useful information for a channel selection algorithm executed by a higher layer. LQI measures the signal strength for each received packet, which is usually used as an indicator of signal quality.
- Low duty cycle is a type of requirement for working style. For a single IEEE 802.15.4 device working as part of a WSN for environment monitoring, it is reasonable for the interval between the sending of a report containing sensor readings (e.g. 1 byte temperature reading) to 1 min or longer. Briefly, assuming an IEEE 802.15.4 packet containing a 22 byte payload is transmitted with a data rate of 250 kbps every 1 min, the required transmission time is $22 \times 8/250\ \text{kbp} = 0.704$ ms. Then the duty-cycle of this IEEE 802.15.4 device is $0.704/(1 \times 60 \times 1,000) = 1.17 \times 10^{-3}$ %. The transmitter is in an inactive state for the rest of the working period. By following the methods of low duty cycle approach, the chance for the IEEE 802.15.4 device to compete with interfering signals can be significantly decreased.
- Low transmit power and channel alignment: Low transmit power is a mechanism mainly for promoting an IEEE 802.15.4 device's capability to coexist with other wireless systems. Although the Federal Communication Commission (FCC) rules allow transmit power up to 1 W in the 2.4 GHz band, IEEE 802.15.4 devices are likely to operate with much lower transmit power (i.e. typically 1 mW) to minimize interference with other wireless devices. Channel alignment requires a proper separation between the IEEE 802.15.4 communication channel and the

potential wireless systems, which can enable multiple wireless systems to work simultaneously without significant mutual interference.

7.6 Advanced Mitigation Strategies

There is significant research being undertaken in interference mitigation. Two main areas are channel switching and retransmission. Swapping the current working channel of the IEEE 802.15.4 based WSNs to a relative free frequency when interference occurs is an easy and effective way to combat interference. Alternatively, consecutive transmission for a certain time can avoid the extra cost in channel swapping. We will only review a few existing typical mitigation strategies in this section.

7.6.1 Adaptive Interference-Aware Multi-Channel Clustering

Kang et al. (2007) proposed an adaptive interference-aware multi-channel clustering algorithm to avoid any IEEE 802.11 interference in a ZigBee network. In the description of this algorithm, a stationary ZigBee network is assumed so that no topology change or mobile node addition is allowed. A typical topology is shown in Fig. 7.9.

In Fig. 7.9, the ZigBee devices are classified into a number of clusters. Except for the PAN Coordinator, each cluster has a cluster head (CLH) responsible for cluster management and a number of ZigBee devices (Dev). A cluster identifier (CID) is used by devices in the same cluster to establish communication. There are two channel settings: An intra-cluster channel for devices in the same cluster and an inter-cluster channel for a cluster header and a bridge device (BRD). A bridge

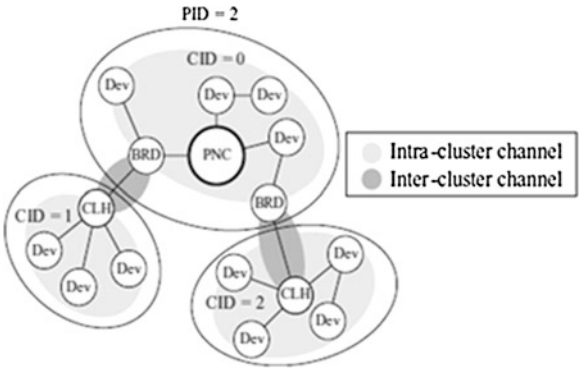


Fig. 7.9 ZigBee network with intra and inter clusters (Kang et al. 2007)

device is a node directly connected to a cluster header of a neighbouring cluster. The use of Inter-cluster is to increase the coverage area of a ZigBee network. The algorithm consists of the following two steps:

- Interference Detection: Once a device in a cluster detects the existence of IEEE 802.11 interference (e.g. loss of beacon synchronization, or loss of acknowledgement), it should broadcast a channel change broadcast message (CCBM) throughout the cluster, allowing the other devices in the same cluster to detect the interference.
- Interference Avoidance: On receipt of the CCBM, devices in the same cluster start to change their channel to a new channel. To ensure each device can move to the same channel without introducing additional cost, a combination of PAN identification, cluster ID, current channel and channel switch counter is used as a key to generate the next channel. Devices sharing the same parameters can obtain the same result. These parameters are inputted in a pseudo random sequence generator (PRSG) as shown in Fig. 7.10.

For the Inter-cluster connection, the cluster head periodically sends a test frame to the bridge node. If a number of acknowledgements are lost, the cluster head assumes that the Inter-cluster channel is experiencing interference. It then sends a CCBM frame and moves to the next channel. For the bridge node, if a number of test frames are not received as scheduled, it also transmits a CCBM to the cluster to which it belongs, and moves to the next channel.

7.6.2 Adaptive Radio Channel Allocation

An adaptive radio channel allocation for supporting coexistence of 802.15.4 and 802.11b was proposed by Won et al. (2005). This algorithm also focuses on an IEEE 802.15.4 based WSN for large-scale deployment. Figure 7.11 illustrates a scenario for the coexistence of 802.15.4 and 802.11b networks.

In Fig. 7.11, a deployed IEEE 802.15.4 sensor network using a mesh topology is being interfered with by IEEE 802.11b signals. The node named 'Src' is set to send data packet to the node 'Dst'. The routing path from the source node to the destination node has been pre-configured by following the solid arrows. The shaded area is the part of the network being affected by the interference. If the source device can reselect a new route to bypass the affected area, the problem of

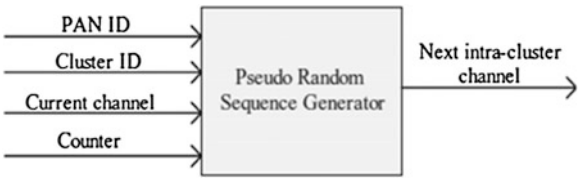


Fig. 7.10 Block diagram for pseudorandom sequence generator (Kang et al. 2007)

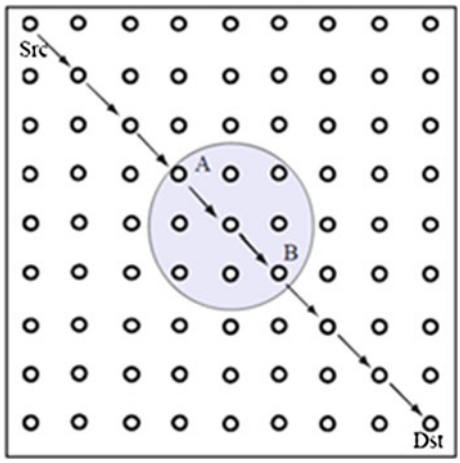


Fig. 7.11 Multi-hop IEEE 802.15.4 network with interference (Won et al. 2005)

interference can be solved. However, additional computation cost will be generated. Won et al. (2005) introduced a strategy to reduce the cost of additional route selection by enabling the nodes within the interfering area to temporarily switch communication channel. The strategy implementation consists of three steps: interference detection, group formation and tear-down.

- Interference Detection. Each IEEE 802.15.4 node in the mesh network keeps monitoring the data throughput and executing interference detection using the standard function of energy detection, or clear channel assessment. Once a sudden degradation of throughput is detected, and the energy detection returns a high level value, the node will enter into a procedure of group formation in order to form a temporary group in a clean channel.
- Group Formation. The node, which starts the procedure of group formation, should provide its immediate neighbours with information about the channel to which it is going to switch. On receipt of this message, the neighbour node will change its role to act as a border node, which establishes a bridge between the original mesh network and the nodes within the interference area. The border node will send a reply message on the new channel to the node from which it received the group formation message. The reply message is to confirm that the border node is aware of the situation change. Next, the border node switches back to the previous channel. If the border node receives new data for the nodes that have joined the temporary group, it quickly switches to the channel used by the temporary group, and sends the data to the desired node. After completion of data sending, the border node returns to the original channel and continues to listen on it.
- Tear-down. The nodes in the temporary group keep checking the previous channel periodically. If the channel is determined to be clear, they will send a tear down message to all immediate neighbours, especially the border nodes.

Consequently, the whole group will be torn down when the interference has completely diminished.

7.6.3 Consecutive Data Transmission

The mitigation strategy can start from the point whether it is possible for an IEEE 802.15.4 system, when it is suffering from interference, to maintain communications by using consecutive data transmission (Yao and Yang 2010). Assuming the success rate for a IEEE 802.15.4 device communication under interference is $R_{Interference}$, and the system is achieving at least one successful communication at a probability of $P_{Success}$ after consecutively sending n packets, the following equation should be satisfied:

$$1 - (1 - R_{Interference})^n \geq P_{Success} \tag{7.6}$$

Then the number of consecutive data packets n can be derived by

$$n \geq \frac{\log(1 - P_{Success})}{\log(1 - R_{Interference})} \tag{7.7}$$

Given a successful rate $R_{Interference}$, which can be obtained from a baseline test, the number of consecutive data packet sending given by Eq. (7.7) can guarantee the success of communication. For example, if the selected rate $R_{Interference}$ is 25 %, and the demanded probability $P_{Success}$ is 90 %, the IEEE 802.15.4 device should send out 8 (i.e. $n = 8$) times of data request packets. If the desired acknowledgement is not received after the consecutive data transmission, the IEEE 802.15.4 assumes that the interference is serious. It should then start energy detection and switch to a clear channel with the least energy activities.

7.6.4 Multi-hop Data Transmission Control

When multi-hop transmission is required in an IEEE 802.15.4 ad hoc network, particularly for large volume data transfers, the setting of the transmission interval is the key point that determines the success of transmission. Compared with the procedures for implementing data transmission between a single pair of devices, multi-hop transmission has a high possibility of transmission failure as an individual communication link failure on the route can interrupt the whole transmission process.

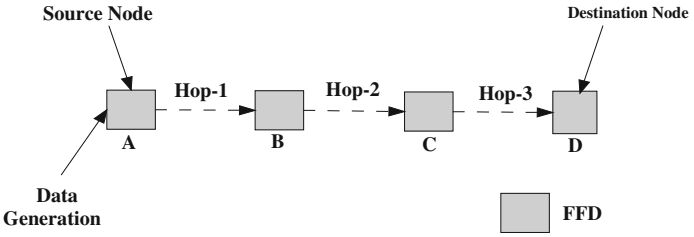


Fig. 7.12 Simplified model for multi-hop transmission

7.6.4.1 Transmission Interval Between Two Consecutive Packets

Figure 7.12 illustrates a simplified model of multi-hop transmission. In Fig. 7.12, the model of multi-hop transmission can be described as a transmission chain. Device A is the source device. Device D is the destination device waiting for data from device A. The metric used to measure the performance of multi-hop transmission is 'Arrival Rate (AR)'. The AR expresses the ratio of data that successfully reaches the destination device to the total data sent from the source device.

According to the IEEE 802.15.4 standard, device A should complete three standard steps to ensure that the transmission is successful.

- Implement CSMA-CA to detect if the channel is clear for data transmission.
- Send data to the next hop.
- Wait for acknowledgement from the next hop.
- Devices B and C acting as intermediate devices on the route require four steps to complete the task of relay.
- Receive data sent from the previous node on the route, sending back acknowledgement if required.
- Implement CSMA-CA to detect if the channel is clear for data transmission.
- Send data to the next hop.
- Wait for acknowledgement from the next hop.

Device D, which is the destination device of the multi-hop transmission, needs to receive the data relayed from device C, and send back acknowledgement if required. A description of multi-hop transmission based on the same time line is illustrated in Fig. 7.13.

Figure 7.13 gives a comparison of actions taken by all devices involved in a multi-hop transmission on the same timeline. Theoretically, if the data have been successfully passed to device C by device B, the source node, device A, can start a new data transmission for the following data packet. However, packet collisions will happen if the transmissions are not well scheduled, or the transmission time interval is too small.

Wireless communication has an important restriction that only one radio transceiver is allowed to emit radio signals within a given period if multiple transceivers exist in the same area (Golmie 2006). It is well known that the conflict

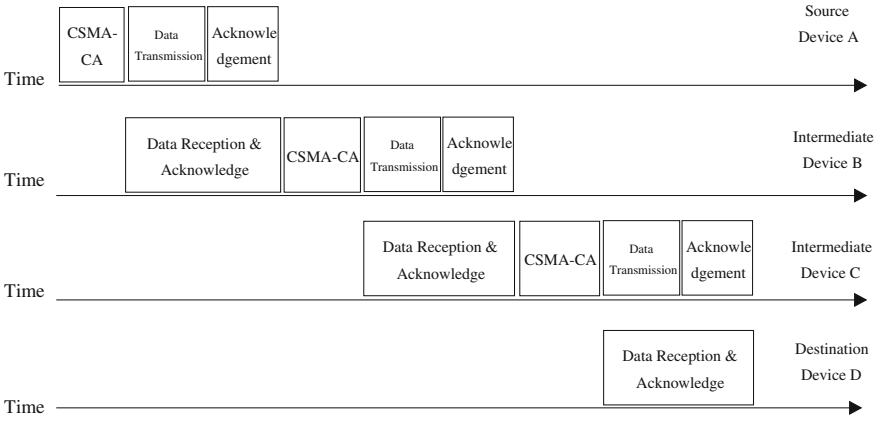


Fig. 7.13 Completed multi-hop transmission based on the same time line

that happens during wireless communications is classified as the issues of 'hidden node' and 'exposed node' (Hwang et al. 2005; Koubaa et al. 2006). The wireless devices are located relatively close to each other in order to ensure the reliable connectivity. Furthermore, routing protocols usually take various factors into consideration in the route selection, e.g. signal strength, device response time delay and distance from the candidate to the destination device. Therefore, it is possible that intermediate devices selected for a route are within the same communication range as each other. For the simplified model shown in Fig. 7.12, we assume that device A and device C are within a 1 hop communication range. By following the same process in Fig. 7.13, if device A starts to send the second packet, whilst the first packet is being transferred from device C to the destination device D, it is possible that both of these two packet transmissions will collide. Figure 7.14 shows the collision occurring under this circumstance.

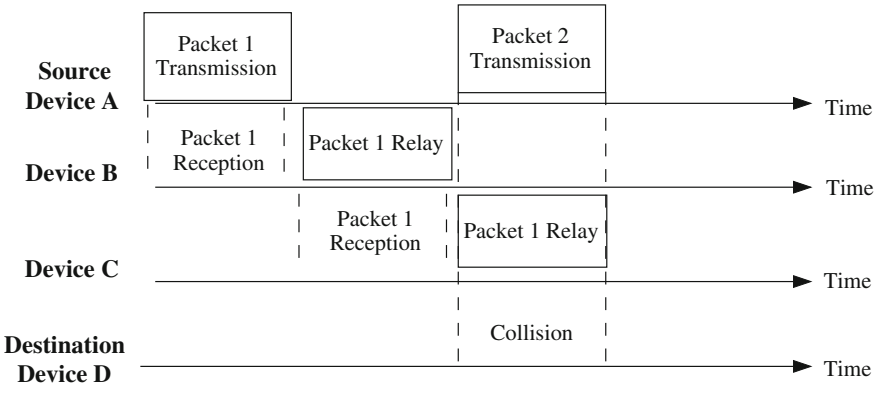


Fig. 7.14 Packet collisions in multi-hop transmission

Another possible situation is that the action 'Packet 1 Relay' starting from device B to device C and the action 'Packet 2 Transmission' will cause channel contention. Then, one of them should defer channel access and wait for a random delay before making the next attempt. If the transmission interval is not controlled properly on the source device A (e.g. the interval is too small), the subsequent packet transmission could cause an even greater delay. The default retransmission mechanism for the IEEE 802.15.4 standard is less effective under such circumstance as it is implemented for the scenario where an expected acknowledgement is not received after data transmission. If more intermediate devices are involved in multi-hop transmission, the collision and channel contention will become more complicated and unpredictable. The uncertainty of multi-hop transmission must be considered in the design of the transmission protocol.

To ensure the success of multi-hop transmission, the source device should set the interval between each packet transmission to a minimum level, which should be equal to the time required for a packet travelling from the source device to the destination device. If a subsequent data transmission starts only after the reception of previous data on the destination device, there will be little chance for collision and channel contention. The minimal interval is defined as follows:

$$T_{Total}(L) \times N_{Hops} \tag{7.8}$$

where $T_{Total}(L)$ denotes the time required to send an L bytes packet from one device to another device in a single hop transmission. L is the size of the packet. N_{Hops} is the number of hops involved in the multi-hop transmission. The minimum interval between each the transmission of two consecutive source packets can be set in the MAC layer in cooperating with the associated upper layers (e.g. the network layer and the application layer).

7.6.4.2 Mitigating Interference in Multi-hop Transmission

Deploying an 802.15.4 networks using a mesh topology will provide the greatest flexibility should interference occur. The routing protocol used can be an alternative to retrying a multi-hop data transmission, if an existing route becomes unavailable. Under certain conditions, when most of the wireless nodes are suffering interference, switching to another clear channel can be effective. However, the problems of implementing channel switching will increase, with the growth in the coverage of the 802.15.4 network since keeping the whole 802.15.4 network synchronized would require considerable cost.

Figures 7.15 and 7.16 illustrate a means of mitigating interference by discovering an alternative route that goes round the interference area (Salvatore and Yang 2012). In order to facilitate the operation, every node will store a list of immediate neighbours. Every neighbour has an associated interference bit that indicates whether that node is considered as suffering from interference or not. In detail, every node keeps track of the missed acknowledgment packets for every

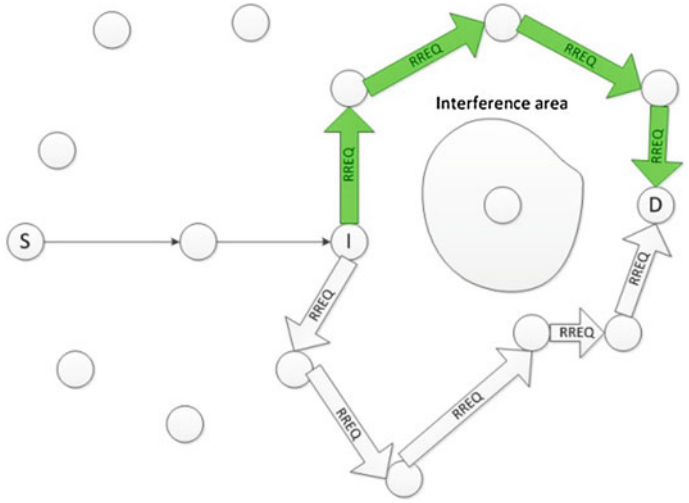


Fig. 7.15 New route discovery phase after interference detection

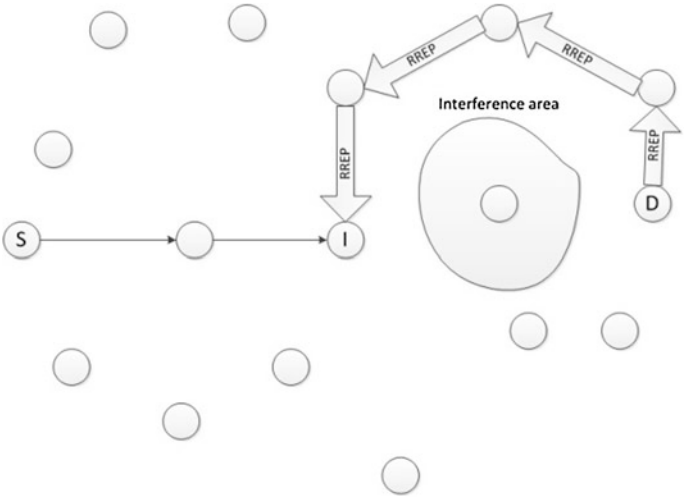


Fig. 7.16 RREP (Route REPly) propagation after interference detection

neighbour. If the number of missing ACKs is over a specified limit, that neighbour is considered inside the interference area and the corresponding interference bit is set to one. The nodes deemed to be within the interference area will not participate in the new route discovery phase.

It is still useful for the intermediate node to send a message back to the source node in order to inform it about the link break. In this way, the source node will

stop sending any data packet and will wait for a specified amount of time. The duration of this time can be adjusted according to the network's size. Only in the case where the intermediate node is not able to find a new route to the destination, will it send a RERR message back to the source node, so that it can start a new route discovery phase from the beginning. If a RERR message is not received, the source node will assume that a new route to the destination is available, and it will restart sending data packets to the previous next hop. The above idea can also be applied to the case where several interference areas are detected.

7.7 Empirical Study

Empirical studies are usually employed by researchers as a more practical approach to investigate the effect of interference (Jennic 2008).

7.7.1 Single Hop Transmission

Petrova et al. (2006) illustrated an interference test to evaluate the coexistence issue of an IEEE 802.11 network and an IEEE 802.15.4 network, as shown in Fig. 7.17.

In Fig. 7.17, the distance between two IEEE 802.15.4 devices and two IEEE 802.11b/g devices is equally set as 5 m. The test was implemented with various frequency offsets between communication channels employed by the two systems, and various lengths of IEEE 802.15.4 packets. The result is shown in Fig. 7.18. If the frequency offset between the IEEE 802.15.4 and the IEEE 802.11 channels is over 7 MHz, the packet error rate of the IEEE 802.15.4 system can be regarded as acceptable (i.e. around 1 %). A noticeable result is that packets with a larger size are more prone to errors.

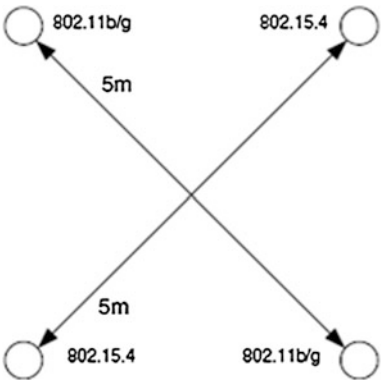


Fig. 7.17 Test bed for single hop transmission

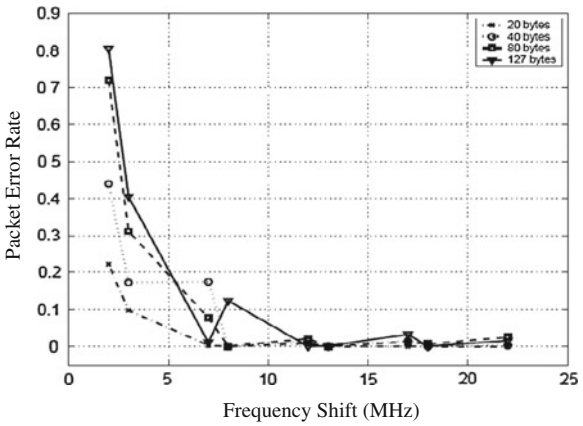


Fig. 7.18 IEEE 802.15.4 PER when interfered by an 802.11 transmission (Petrova et al. 2006)

7.7.2 Multi-hop Transmission

The experimental studies here (Yao 2010) consist of three tests: a baseline test, an interference test and a data recovery test. The baseline test and the interference test were mainly to address the transmit capability of multi-hop transmission in an IEEE 802.15.4 based ad hoc network with and without the presence of Wi-Fi interference. The data recovery test is designed to evaluate if the new route discovery can improve the performance of the multi-hop transmission. All experiments were carried out on the Jennic JN5139R1 platform (Jennic 2009).

7.7.2.1 Baseline Test: Transmission Control on Multi-hop Communications

In Sect. 7.6.4.1 and Eq. (7.8) we stated that a minimal transmission interval between two consecutive transmission packages is needed to avoid overloading on any intermediate devices. The baseline test is designed to illustrate how seriously the setting of the transmission interval impacts the performance of a multi-hop transmission. The deployment of the test devices is illustrated in Fig. 7.19.

In Fig. 7.19, device A is the destination device. Device B is the source device, which sends data to device A. The devices located between device A and B form intermediate devices for relaying the data. The transmission route has been manually scheduled and programmed into these intermediate devices. The source device B will send data by following the sequence of hops indicated in Fig. 7.19 (Hop 1- > Hop 2- > …- > Hop n) until it reaches device A. In this test, the number of hops ranges from 2 to 6 hops, and was previously specified in the intermediate devices. The software running on device A records the amount of data received.

To make it convenient for comparison, the payload length of each IEEE 802.15.4 packet was fixed at 50 bytes, which is half of the maximum MAC layer data payload length defined in the IEEE 802.15.4 standard. The total amount of

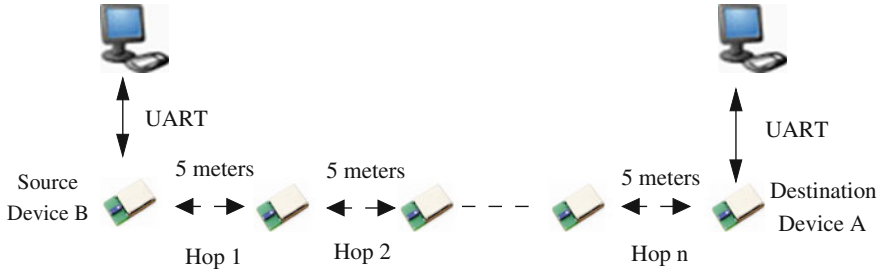


Fig. 7.19 Hardware deployment in the baseline test

data payload sent from device B was fixed at 1 MBytes, i.e. 20,000 IEEE 802.15.4 data packets will be sent from device B to device A. Figure 7.20 illustrates the baseline test results.

In Fig. 7.20, the horizontal axis expresses the transmission interval between two consecutive transmission packages set on the source device B. The vertical axis expresses the corresponding data arrival rate measured on the destination device A. The test result verifies that the arrival rate is obviously related to the number of hops involved in the multi-hop transmission. For example, to achieve a satisfied arrival rate (over 99 %) for a 2 hops transmission, the minimal time interval is approximately 10 ms, whereas a multi-hop transmission with 4 hops requires a value of at least 19 ms. It is clear that if more hops are employed in multi-hop data transmission, longer transmission interval should be specified.

7.7.2.2 Interference Test

The interference test is designed to investigate the effect of interference on the arrival rate of multi-hop transmissions. In the interference test, an IEEE 802.11b router was located close to one of the intermediate devices, and broadcast IEEE 802.11b signals using a fixed packet rate (e.g. 10 packet/second, 100 packet/second). During the period of an IEEE 802.15.4 multi-hop transmission, every intermediate device recorded the number of packets they successfully received. By

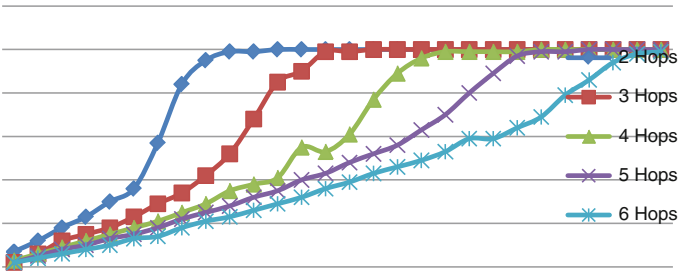


Fig. 7.20 Results of the baseline test

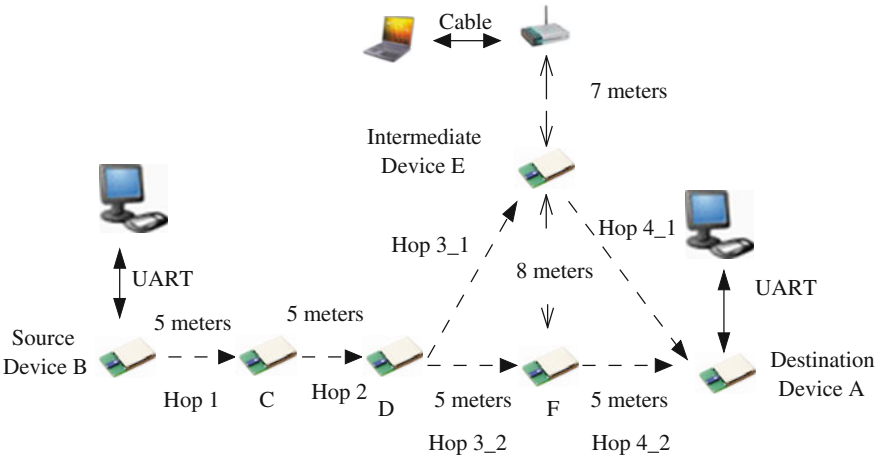


Fig. 7.21 Hardware deployment in an interference test

comparing with the total number of packets sent from the source device, it will be clear how the interference affects the multi-hop transmission. Figure 7.21 illustrates the hardware deployment in the interference test.

In Fig. 7.21, an IEEE 802.11b Wi-Fi router working on channel 13 (2.472 MHz) is connected to a laptop, and placed 7 m away from the intermediate device E. A dedicated software based packet generator is set to run on the laptop, and constantly interferes with device E. Four hops data transmission is pre-specified in the IEEE 802.15.4 ad hoc network, following the sequence of Hop1- > Hop2- > Hop3_1- > Hop4_1. The transmission interval is set at 22 ms, which can achieve around a 99 % arrival rate according to the baseline test, as shown in Fig. 7.20. The IEEE 802.15.4 ad hoc network works on channel 23 (2.465 MHz). The centre frequency is 7 MHz away from the centre frequency of the communication channel used by the IEEE 802.11b router.

During the interference test, the IEEE 802.11b wireless router broadcasts signals, with a packet rate increasing from 10 to 600 packet/second. The 802.15.4 packet contains 50 bytes data payload. The total number of IEEE 802.15.4 packets sent from the source device is 20,000. The test result from the interference test is shown in Table 7.2 illustrates that interference coming from the IEEE 802.11b wireless router can cause a considerable decrease in the IEEE 802.15.4 ad hoc network arrival rate when the IEEE 802.11b signal is working with a heavy duty-cycle. The arrival rate is the percentage of the accepted 802.15.4 packets comparing with the total packets sent from the source device B.

It is obvious that the significant decrease in the arrival rate starts from device E, which is surrounded by the 'Interference Area'. The worst situation is observed in the condition when the wireless router works at 600 packet/second, and the corresponding arrival rate measured on the destination device A is only 42.58 %.

Table 7.2 Summary of packet received on each device involved in IEEE 802.15.4 multi-hop transmission

802.11b packet rate (packet/second)	Device C arrival rate (%)	Device D arrival rate (%)	Device E arrival rate (%)	Device A arrival rate (%)
10	99.75	99.75	98.45	98.10
100	99.99	99.87	98.99	96.91
200	97.53	97.30	93.03	91.29
300	98.04	97.76	91.66	89.35
400	98.80	98.28	72.06	68.79
500	97.39	96.51	49.05	44.06
600	99.78	99.12	47.49	42.58

7.7.2.3 Data Recovery Test

The data recovery test is implemented by employing an alternative route in which the intermediate device E is replaced by device F once interference is detected. The new route used in the data recovery test is Hop1- > Hop2- > Hop3_2- > Hop4_2. The distance between the IEEE 802.11b wireless router and device F is $8 + 7 = 15$ m. Because of the physical separation, the interference from the IEEE 802.11b wireless router on device F is expected to be much weaker. The arrival rate of the destination device A is 95, 86, and 82 % under the IEEE 802.11b packet rate 400, 500 and 600 packet/second respectively.

7.8 Summary

This chapter introduces the definition of interference, the cause of interference, and some mitigation strategies. The emphasis is focused on interference between IEEE 802.11b and IEEE 802.15.4. In general, interference occurs only if two conditions are satisfied: a small or zero offset of the radio frequency, and powerful interfering energy. The mitigation strategies can be categorized as follows:

- *Channel Selection*: It is recommended to use channels 25 and 26 to avoid most of the IEEE 802.11b/g interference. If the system is deployed within an environment where the pre-configuration of wireless systems is controllable, a channel centre-frequency offset of 7 MHz should be reserved to ensure acceptable coexistence with IEEE 802.11 systems.
- *Physical Separation*: Ensuring a physical separation of at least 8 m from an IEEE 802.11 access point is useful for coexistence.
- *Mesh Networking*: If applicable, an IEEE 802.15.4 network can be constructed on the basis of a mesh topology which provides the additional benefits of a self-organizing and self-healing capability.
- *Network Layer Frequency Agility*: By switching to a clean channel when interference occurs, an IEEE 802.15.4 network can effectively avoid

performance degradation. The channel hopping is normally supported by high level protocols (e.g. network layer). The decision of dynamic channel selection should be made in terms of the results of channel assessments (e.g. energy detection, link quality indicator).

- *Network Planning*: Before deploying an IEEE 802.15.4 network, initial assessment such as a site survey can be performed to evaluate the radio frequency environment. The results provide important guidance for the physical installation. During the period of system operation, a radio frequency environment evaluation can be periodically performed to monitor any changes in possibility interference.

References

Chandra, P., Dobkin, D.M., Bensky, D., Olexa, R., Lide, D., Dowla, F.: Wireless Networking, Know It All. Newnes (2007)

Cuomo, F., Luna, S.D., Monaco, U., Melodia, T.: Routing in ZigBee: Benefits from exploiting the IEEE 802.15.4 association tree. IEEE Int. Conf. Commun., 3271–3276 (2007)

Fakatselis, J.: Processing gain for direct sequence spread spectrum communication systems and PRISM. In: Harris Semiconductor Application Note (1996)

Fakatselis, J.: Processing gain in spread spectrum signals. In: Harris Semiconductor Application Note (1998)

Golmie, N.: Coexistence in Wireless Networks Challenges and System-Level Solutions in the Unlicensed Bands. Cambridge University Press, Cambridge (2006)

Golmie, N., Cypher, D., Rebala, O.: Performance analysis of low rate wireless technologies for medical applications. Comput. Commun. **28**(7), 1266–1275 (2005)

Hwang, L.J., Sheu, S.T., Shih, Y.Y., Cheng, Y.C.: Grouping strategy for solving hidden node problem in IEEE 802.15.4 LR-WPAN. In: Proceedings of First IEEE International Conference on Wireless Internet (WICON), pp. 26–32 (2005)

IEEE Standard 802.11.: IEEE standard for information technology: Telecommunications and information exchange between systems: Local and metropolitan area networks: Specific requirements: Part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications (2007)

IEEE Std 802.15.4.: IEEE standard for information technology: Telecommunications and information exchange between systems: Local and metropolitan area networks: Specific requirements part 15.4: Wireless medium access control (MAC) and physical layer (PHY) specifications for low-rate wireless personal area networks (LRWPANs) (2003)

IEEE.: 802.15.2 definition of coexistence. Available at http://grouper.ieee.org/groups/802/15/pub/2000/Sep00/99134r2P802-15_TG2-coexistenceinteroperabilityandotherterms.ppt (2000)

Jennic.: Co-existence of IEEE 802.15.4 at 2.4 GHz. In: Jennic Application Note (2008)

Jennic.: JN5139 data sheet. Available at http://www.jennic.com/download_file.php?supportFile=JN-DS-JN5139MO-1v5.pdf (2009)

Kang, M.S., Chong, J.W., Hyun, H., Kim, S.M., Jung, B.H., Sung D.K.: Adaptive interference: aware multi-channel clustering algorithm in a ZigBee network in the presence of WLAN interference. In: IEEE International Symposium on Wireless Pervasive Computing (2007)

Koubaa, A., Alves, M., Tovar, E.: IEEE 802.15.4: A wireless communication technology for large-scale ubiquitous computing applications. In: Proceeding of Conference on Mobile and Ubiquitous Systems. Guimarães (2006)

Molisch, A.F.: Wireless Communications. Wiley (2005)
Petrova, M., Riihijarvi, J., Mahonen, P., Labella, S.: Performance study of IEEE 802.15.4 using measurements and simulations. In: Wireless Communications and Networking Conference (WCNC), pp. 487–492 (2006)
Rodriguez, R.: MC1319x coexistence. In: Freescale Semiconductor Application Note, AN2935 (2005)
Salvatore, D., Yang, S.H.: Routing algorithm of WSN under interference environment. In: Wireless Sensor Systems: IET Conference, London (2012)
Shin, S.Y., Park, H.S., Kwon, W.H.: Mutual interference analysis of IEEE 802.15.4 and IEEE 802.11b. Comput. Netw. **51**(12), 3338–3353 (2007)
So, J., Vaidya, N.: Multi-channel MAC for ad hoc networks: Handling multi-channel hidden terminals using a single transceiver. In: Proceedings of the 5th ACM International Symposium on Mobile Ad hoc Networking and Computing, pp. 222–233 (2004)
Thone, G., Allard-Jacquin, P., Colle, P.: ZigBee-WiFi coexistence, white paper and test report (2008)
Won, C., Youn, J.H., Ali, H., Sharif, H., Deogun, J.: Adaptive radio channel allocation for supporting coexistence of 802.15.4 and 802.11b. IEEE Veh. Technol. Conf. **4**, 2522–2526 (2005)
Yao, F.: Interference mitigation strategy design and applications for wireless sensor networks. PhD thesis, Loughborough University, Loughborough (2010)
Yao, F., Yang, S.H.: Mitigating interference caused by IEEE 802.11b in the IEEE 802.15.4 WSN within the environment of smart house. In: Proceedings of the 2010 IEEE International Conference on System, Man, Cybernetics. IEEE, Istanbul, Turkey, pp. 2800–2807 (2010)
Yilmaz, O.: Propagation simulation for outdoor wireless communications in Urban areas. Available at www.ee.bilkent.edu.tr/grad/ms-thesis/yilmaz-ms.pdf (2002). Accessed Aug 2011
ZigBee, A.: ZigBee and wireless radio frequency coexistence. Available at http://www.zigbee.org (2007). Accessed June 2012

Chapter 8
Sensor Data Fusion and Event Detection

Keywords Data fusion · Pattern extraction · Data mining · In-network database

8.1 Introduction

Sensor data often contains large amount of redundancy, which would be hard for even a professional data analyst to interpret. Sensor data also frequently contains 'noise', which can be difficult to separate out from 'real' data. In addition, sensor data is usually meaningless unless it is associated with the time and location of the information. This chapter focuses mainly on the steps and technologies required to make sensor data usable.

8.1.1 Features of Sensor Data

In comparison with traditional data, data from WSNs has distinctive special features, which bring challenges in managing and processing such sensor data.

8.1.1.1 The Streaming Nature of Data

Sensor data is best modelled as continuously arriving data streams rather than persistent relations. Data streams differ from traditional data in the following ways (Yang et al. 2010):

- Sensor data is automatically generated and arrives in a multiple, continuous, time-varying manner. Therefore, the volume of sensor data increases with time, and the total volume of data are potentially unlimited. In contrast to this, traditional data is typically entries input by human, permanently or persistently stored in a database, and has a limited volume.

S.-H. Yang, *Wireless Sensor Networks*, Signals and Communication Technology,
DOI: 10.1007/978-1-4471-5505-8_8,

- "Data stream is time ordered data, either explicit with a time-stamp or implicit based on arrival order" (Kim et al. 2005). However, traditional data is usually not time-ordered unless specified explicitly.

As a result, the streaming feature of sensor data presents challenges in sensor data processing such as storing data with unbounded increasing volume, continuous loading and continuous queries.

8.1.1.2 Existence of High Tempo-Spatial Correlations

Sensors are usually deployed at a certain density so that they can cover the entire monitoring field. As a result, "most sensor networks likely exhibit temporal and spatial correlations among node readings" (Silberstein et al. 2007). More specifically, the high temporal and spatial correlations that exist in sensor data causes them to exhibit the following characteristic: "the readings observed at one time instant are highly indicative of the readings observed at the next time instant, as are readings at nearby devices" (Jeffery et al. 2006).

However, the high tempo-spatial correlation can provide potential benefits. They can be used to estimate missing or corrupted data (Chok and Gruenwald 2009), detect outliers and improve the quality of sensor data, provide data suppression (Silberstein et al. 2007), reduce data transmission in the network and thus reduce energy consumption. However, challenges exist in identifying correlations and modelling correlations, and keeping correlation information updated, etc.

8.1.1.3 Generation of Redundant Data

Significant data redundancy in a database can result from the strong spatial and temporal correlations typically present in sensor data. However, redundancy can be used to predict missing values and to detect outliers and a certain level of redundancy can improve the accuracy of database query results. Resolving redundancy in sensor data is not simply a case of removing redundant data, but rather maintaining it at a level that provides confidence in the data without producing unnecessary storage demands.

8.1.1.4 Sensor Data Contains 'Noise'

Sensors are designed to be low power consumption and low cost. However, this design focus can result in the accuracy of the sensors being limited. As well as design issues, sensors are normally deployed in harsh environments with the potential for background interference, and consequently sensors can experience internal faults or damage during emergencies such as fires. Research has shown that sensor data often contains errors (due to sensor function) and noise (due to other

environmental interference) (Elnahrawy and Nath 2003). These characteristics indicate that sensor data should be cleaned before being stored in any database.

8.2 Sensor Data Fusion Techniques

The process of sensor data fusion consists of three stages: pre-processing, data mining and post-processing. Figure 8.1 shows the overall process of information extraction from raw data (Tan 2006).

8.2.1 Sensor Data Pre-processing

The data quality problem in sensor networks is an issue that has recently been receiving increased interest. Sensor data often contains noise (Elnahrawy and Nath 2003), outliers (Basu and Meckesheimer 2007), and missing values (Allison 2001). The causes of such data quality problems include (1) sensors' internal errors, (2) the harsh environment in which sensors are deployed, and (3) damage or loss during wireless transmission, as shown in Fig. 8.2. Data pre-processing includes data cleaning, outlier detection, missing values recovery, data reduction, dimension reduction, and data prediction, etc.

8.2.1.1 Sensor Data Cleaning

Various approaches have been used for sensor data cleaning, including Bayesian Theory, Neural Network, Wavelets, Kalman Filter and Weighted Moving Average. Because of the limits imposed by the sensor's computation capability it is hard to implement the Bayesian Theory, Neural Network, and Wavelets methods. Only Kalman Filter and Weighted Moving Average are considered in this chapter.

Zhuang et al. (2007) proposed a smart weighted moving average based sensor data cleaning approach, which consists of three steps.

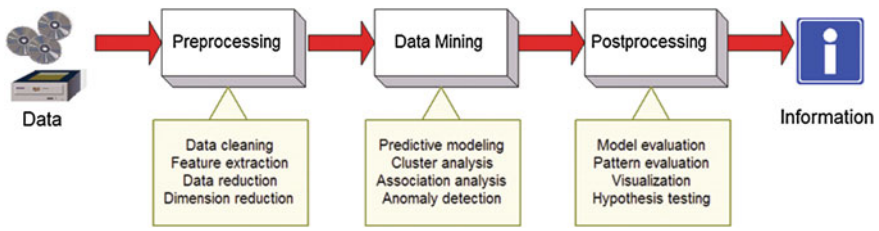


Fig. 8.1 Overall process of information extraction from data (Tan 2006)

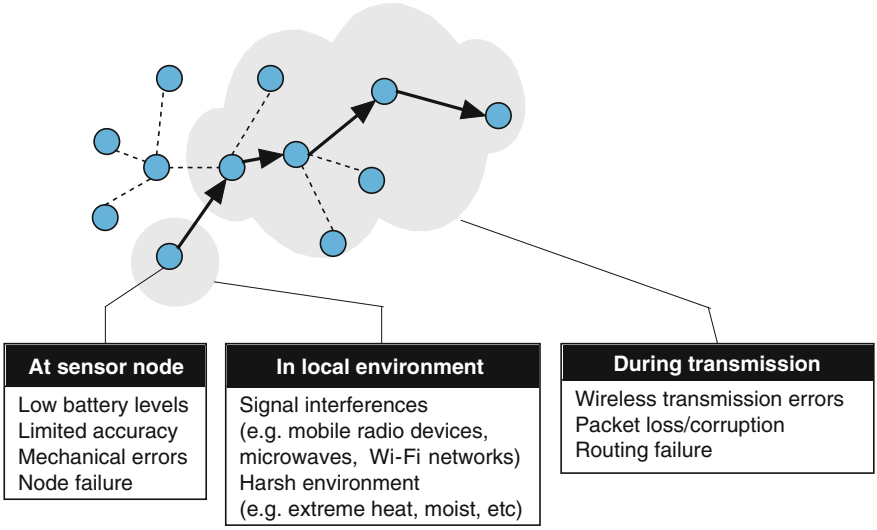


Fig. 8.2 Possible causes of data quality problems in sensor networks

- Step 1: Locate important values by range prediction.
- Step 2: Increase confidence for important values by sensor testing and neighbour testing at individual sensors.
- Step 3: Perform weighted moving average at the sink.

This approach used Kalman Filter and Linear Regression for range prediction. Values outside the predicted range would be considered as "important" values, and their confidences would be calculated in Step 2. Finally, the weighted moving average at the sink node is combined with the temporal average and the spatial average together.

8.2.1.2 Missing Values Recovery

To handle the problem of network data loss, the traditional way is to wait for a predefined period of time before the receiver sends a retransmission request to the sender, or the sender automatically retransmits if no acknowledgement has been received from the receiver. However, there are two major drawbacks in applying this approach in sensor networks: (a) "increased power consumptions by the sensors" and (b) "increased latency of the produced result by the query" (Halatchev and Gruenwald 2005). Therefore, the existing research on handling missing sensor data focused on estimating or recovering missing values using available values from those sensors related to the sensor that has the missing values.

A variety of estimation methods were proposed, such as Expectation Maximization (Moon 1996), Association Rules (Halatchev and Gruenwald 2005)

and Belief Propagation (Chu et al. 2005). Expectation Maximization (EM) is a common approach to converge to local maxima of the complete data likelihood (i.e., the likelihood of both the observed and missing data). The E-step calculates the expectation (or possibilities) of the missing nodes values ($P(Y|X,\theta)$), where X represents the observed data, Y represents the missing values, and θ represents the statistical model parameter. Given the expectations for the missing values, the M-step calculates the value for θ that maximizes the expected complete data likelihood. The mathematical detail is not given here and can be found in most statistic textbooks.

8.2.1.3 Sensor Data In-network Aggregation

"Having a large amount of redundant data may slow down or confuse the knowledge discovery process" (Han and Kamber 2006). In-network aggregation of redundant data can reduce the total data flow over the sensor network and thus can extract the most representative data using minimum resources (Akcan and Brönnimann 2007), which effectively reduce power consumption (Santini and Römer 2006). Therefore, a branch of sensor data pre-processing research focuses on sensor data reduction in WSNs.

Getting an average of the raw data and reporting the average when it is greater than a predefined threshold is the simplest case. Table 8.1 shows the query in Structured Query Language (SQL) syntax, where an average temperature "AVG" is taken from a sensor. If the average is greater than 'threshold', the average is sent out "HAVING AVG". The sampling period is 30 s.

Akcan and Brönnimann (2007) proposed a weighted in-network sampling algorithm to obtain a deterministic (much smaller but representative) sample instead of raw redundant data. Compared with random sampling, the advantage of weighted sampling algorithm is "it can guarantee that each node's data has the same chance to belong to the final sample, independent from its provenance in the network".

Instead of selectively sampling the network nodes, a prediction-based data reduction strategy (Santini and Römer 2006) is to have prediction methods deployed both at the sensor and sink-level, so that sensors only need to send data that deviates from the predicted value. In detail,

(a) at both the sink node and the sensor node, applying a prediction model G to get an estimate of the sensor reading at the new instant $\bar{X}^{t+1} = G(X^t)$.

Table 8.1 Average aggregation in SQL

```
SELECT AVG(temperature), FROM Sensors
WHERE floor=6
HAVING AVG(temperature) > threshold
SAMPLE PERIOD 30 s
```

(b) at the sensor node, if $\left|X^{t+1} - \bar{X}^{t+1}\right| > \varepsilon$, where $X^{t+1}$is the actual sensor reading at the new instant, and ε is the tolerant error, sending the actual sensor reading to the sink node,
(c) otherwise, the sink node uses the estimate of the sensor reading.

8.2.2 Sensor Data Mining

Data mining aims to extract patterns from data. Traditional data mining technologies (Han et al. 2011), often referred to as called data miners, include Decision Trees, Rule-based Classifiers, Artificial Neural Networks, Nearest Neighbour, Naive Bayes, Support Vector Machines, Logistic regression, etc. Most of these were initially developed to be applied in central data warehouse.

Sensor data mining mainly focuses on distributed in-network data mining. Most authors suggested a form of hierarchical network topology for sensor data mining. Bontempi and Borgne (2005) proposed a two-level architecture for sensor data mining, as shown in Fig. 8.3.

The lower level consists of aggregator nodes, denoted as dotted circles, which perform modular aggregation of the neighbouring sensor nodes represented as black dots. The aggregated signals are then sent up to the upper level data miner, where the required sensing tasks such as classification, regression or prediction, are performed. This architecture introduced a layer of aggregator nodes in the WSN topology, each of which acts as a cluster head to a number of sensor nodes.

8.2.3 Sensor Data Post-processing

Data post-processing includes pattern evaluation, model evaluation, data visualization/presentation, etc. This step can link the result of sensor data mining to specific applications. Data visualization can be based on computer graphics,

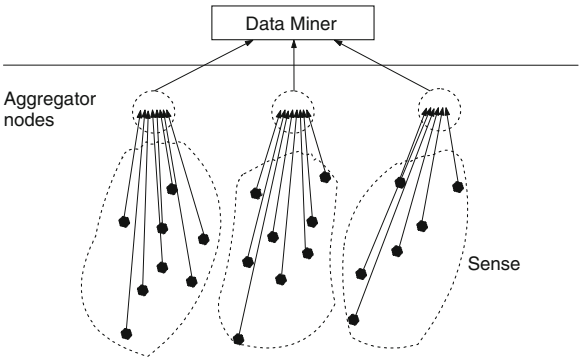


Fig. 8.3 An adaptive modular architecture of sensor data mining (Bontempi and Borgne 2005)

statistical methods, or user interaction techniques. This topic is slightly out of the core area of interest for this text and is omitted here.

8.3 Event Detection

Event detection can be formally described as follows:

Given a set of measured data arriving over time, denoted as $D = \{z_t | t = 1, 2, 3, \ldots n\}$, event detection is to find the time t when an event of interest occurs, i.e. where the data is different from the normal pattern of behaviour.

Therefore, the common goals of event detection are:

- To identify whether an event of interest has occurred;
- To characterize the event (e.g., the time, the affected area, the type and the severity of the event).

Two categories of event detection approaches have been identified in sensor network applications (Yang et al. 2012): threshold-based event detection and tempo-spatial pattern based event detection.

8.3.1 Threshold-based Event Detection

The threshold-based event detection method is based on the underlying intuition that an event occurring will result in changes in the sensor readings, e.g., an object moving will result in an increased acceleration reading, a fire will result in an increased temperature reading. Therefore, normal behaviour can be defined as a threshold (e.g., maximum values, rates of increase and combination thereof from multiple sensors) based on statistics from historical data (or domain knowledge). Alarms can be raised if the predefined threshold is exceeded. The advantage of threshold-based event detection is the simplicity of its implementation and low computation complexity. However, crossing the threshold is highly dependable on the specific detection problem and the environment that sensors are monitoring, and some events cannot be fully captured by discrete threshold values. Thus, the accuracy of detection is limited.

Figure 8.4 shows a multi-threshold based fire detection algorithm. Four types of sensor readings are used, which are the smoke density, the carbon monoxide (CO) concentration, temperature, and flame. A binary value (1 or 0) is introduced to represent the state of each sensor reading. A weighted average is used to combine these four binary values into a single value. If this single value is greater than 50 % a fire alarm is alerted.

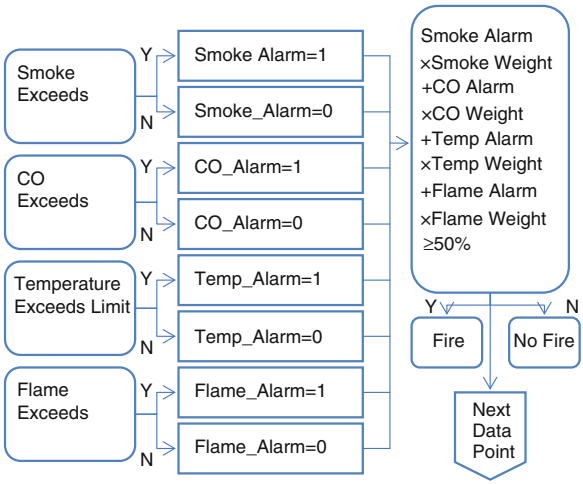


Fig. 8.4 Multiple threshold based fire detection

8.3.2 Tempo-Spatial Pattern Based Event Detection

In contrast to threshold-based event detection, the underlying intuition of tempo-spatial pattern based event detection is that an event occurring in the monitoring field usually results in some tempo-spatial patterns in the sensor readings of network nodes. For instance, a gas leakage event can be characterized as a spatial distribution of sensor readings following a gradual decreasing trend from the source of the leakage to the surrounding area nearby. The event of interest can be defined as temporal (Mukherji et al. 2008), spatial (Xue et al. 2006), or tempo-spatial patterns, and then the event detection problem is converted into a pattern-matching problem.

An example of spatial pattern matching is shown in Fig. 8.5. The pattern extracted from live sensor data in the left column is compared with the pre-defined pattern for an event in the middle column. If they don't match with each other, the detection result is normal; otherwise, an event has been detected. A contour map (Xue et al. 2006) is often used in representing a tempo-spatial pattern, which is a map showing elevations and surface configuration by means of contour lines, along which a sensor reading has a constant value.

The advantage of tempo-spatial pattern based event detection is that it takes context information into account, and incorporates tempo-spatial correlations, which typically exist in sensor data, to improve the accuracy of event detection. However, the disadvantages are:

- Increased complexity, because it incorporates data from the whole network in the pattern matching.

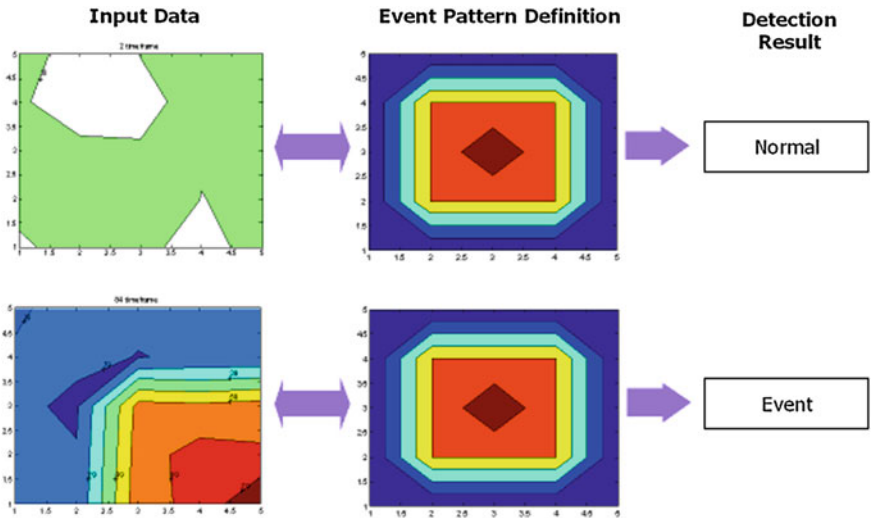


Fig. 8.5 Intuition of spatial based event detection

- Difficulty in defining suitable patterns to represent the events of interest. If a static pattern is defined, its flexibility is low; if the pattern is defined with user-specified factors, tuning the factors results in similar difficulties to those encountered when tuning thresholds.

8.4 Generic Sensor State Model with Neighbourhood Support

8.4.1 Generic Sensor State Model

No matter what phenomenon a sensor is measuring, the sensor can be in one of the four possible states: normal, suspicious, abnormal, and false alarm. The transition among these four states are shown in Fig. 8.6 and described as follows:

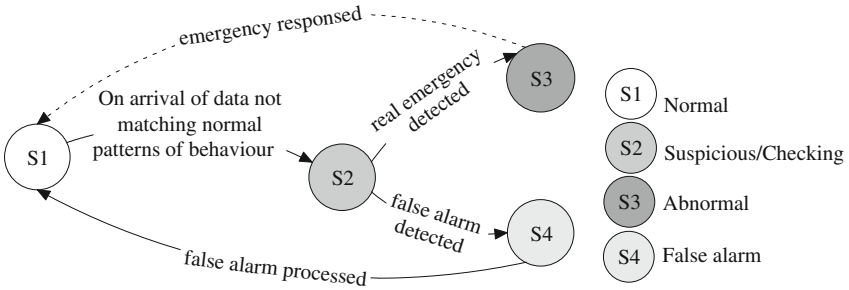


Fig. 8.6 Generic sensor state model

S1 (**Normal**): A state s is considered to be normal when the pattern extracted from the live sensor data P_L matches the predefined normal pattern of behaviour P_N, denoted as

$$s \in S_1 \Leftrightarrow P_L \approx P_N \tag{8.1}$$

S2 (**Suspicious/Checking**): A suspicious state appears when P_L does not match the normal pattern of behaviour P_N, denoted as

$$s \in S_2 \Leftrightarrow P_L \neq P_N \tag{8.2}$$

When a suspicious state occurs, a level of support from its temporal or spatial correlations is required to further confirm the sensor state.

S3 (**Abnormal**): If the level of support l_s is above a threshold th, it indicates that an incident is detected, denoted as

$$s \in S_3 \Leftrightarrow s \in S_2 \wedge l_s > th \tag{8.3}$$

S4 (**False alarm**): If the level of support is not large enough, it is considered to be a false alarm, denoted as

$$s \in S_4 \Leftrightarrow s \in S_2 \wedge l_s < th \tag{8.4}$$

False alarms will be logged, and the sensor reading will go back to its previous state.

8.4.2 Neighbourhood Support

The concept of Neighbourhood Support is more formally defined here as an example of spatial support checking.

Neighbourhood: in an area where a density of sensor nodes are deployed, the neighbourhood of sensor node $s_i$consists of all the sensor nodes that are deployed within the radius r from s_i, denoted as $N_{s_i} = \{s_j | \mathrm{dis}(s_j, s_i) \leq r\}$, where $\mathrm{dis}(s_j, s_i)$ is the distance from sensor node s_j to s_i, assuming that sensor nodes know their geographical location information, either during the deployment stage or through RF-based beacons.

Neighbourhood Support: is defined as the level of support a sensor node gets from its neighbourhood region when it detects a suspicious state, denoted by *NS*. Intuitively, sensor nodes in the neighbourhood can be considered as witnesses, able to confirm or deny the suspicious state that a sensor node detects. It can be implemented either as thresholds or as a contour map.

8.5 Sensor State Model Based Event Detection

The state transition of the sensor state model can be both threshold-based and tempo-spatial pattern based and therefore used in both types of event detection.

8.5.1 Threshold-based Event Detection

The generic sensor state model can be represented as follows: if the state transition is based on a threshold comparison.

S1 (**Normal**): At each node with m types of sensors, the state of each sensor z_i is normal when its rising rate RR_i is within its threshold TH_i, denoted as $z_i \in S_1 \Leftrightarrow RR_i \leq TH_i, i = 1, 2, \ldots, m$. The overall state of a node s is considered to be normal when all m types of sensors are normal, denoted as

$$s \in S_1 \Leftrightarrow z_1 \in S_1 \wedge z_2 \in S_1 \wedge \cdots \wedge z_m \in S_1 \tag{8.5}$$

S2 (**Suspicious/Checking**): The state of each sensor is suspicious when its rising rate exceeds its threshold TH_i,$z_i \in S_1 \Leftrightarrow RR_i > TH_i, i = 1, 2, \ldots, m$. A suspicious state of a node appears when the state of any of the m types of sensors is suspicious, denoted as

$$s \in S_2 \Leftrightarrow z_1 \in S_2 \vee z_2 \in S_2 \vee \cdots \vee z_m \in S_2 \tag{8.6}$$

When a suspicious state occurs, the system will check the Neighbourhood Support, which can be defined as the ratio of the number of neighbours who are in a suspicious state N_{S_2} to the total number of neighbours N_n, denoted as $NS = \frac{N_{S_2}}{N_n}$.

S3 (**Abnormal**): If the Neighbourhood Support is above a threshold th, it indicates that an incident is detected, denoted as

$$s \in S_3 \Leftrightarrow s \in S_2 \wedge NS > th \tag{8.7}$$

S4 (**False alarm**): If the Neighbourhood Support is not large enough, it is considered to be an outlier. Outliers will be logged, and the sensor will go back to its previous state, denoted as

$$s \in S_4 \Leftrightarrow s \in S_2 \wedge NS < th \tag{8.8}$$

8.5.2 Tempo-Spatial Pattern Based Event Detection

In the generic sensor state model shown in Eqs. (8.1) and (8.2), current and normal patterns of sensor behaviour, P_L and P_N, can be presented in the format of a contour map (Xue et al. 2006), which is the distribution of sensor readings over the

network. Renaming P_L and P_N as the contour map extracted from the live sensor reading C_L and the pre-defined contour map event patterns C_E, respectively, the generic sensor state model can be represented as follows if the state transition is based on contour map matching.

S1 (**Normal**): A state s is considered to be normal when the contour map extracted from the live sensor reading C_L does not match any of the predefined contour map event patterns C_E, denoted as

$$s \in S_1 \Leftrightarrow C_L \neq C_E \tag{8.9}$$

S2 (**Suspicious/Checking**): A suspicious state appears when C_L matches the contour map event pattern C_E, denoted as

$$s \in S_2 \Leftrightarrow C_L = C_E \tag{8.10}$$

When a suspicious state occurs, the system will check whether the match continues for a consecutive period of time T.

S3 (**Abnormal**): If the suspicious state lasts for a consecutive period of time T and a neighbourhood support is obtained, an incident is detected, denoted as

$$s(t) \in S_3 \Leftrightarrow s(t - T + 1, \ldots, t) \in S_2 \tag{8.11}$$

where t is the detection time.

S4 (**False alarm**): If the suspicious state does not last for a consecutive period of time T and/or there is not enough neighbourhood support for the suspicious state, it is considered to be a false alarm, i.e. an outlier, denoted as

$$s(t) \in S_4 \Leftrightarrow s(t) \in S_2 \wedge \exists s(t) \in S_1, t = \{t - T + 1, \quad t - T + 2, \ldots, t\} \tag{8.12}$$

8.6 Sensor Network as a Database

A sensor-net database (Govindan et al. 2002) allows users to issue a query to the sensor network as if it is a database system and obtain a response to that query. A sensor-net database is a distributed database as the data is generated and stored in individual sensor nodes. The fundamental difference between a sensor-net database and a traditional distributed database is that the data in a sensor-net database is generated only on demand, i.e. after a request for the data is made by the users. This feature is called an *in-network* implementation of database operators. When a user makes a query to the network, that query is disseminated across the sensor network and transmitted to the relevant sensor nodes. In response to that query, the relevant nodes generate data records that match the query and send the data records back to the users through the network. Chap. 12 of this book will introduce the details of connecting the Internet with WSNs and acquiring data from this particular type of distributed database.

8.7 Summary

Sensor data has a set of special features. This chapter summarizes them as the streaming nature of data, high tempo-spatial correlations, significant redundancy, and errors and noise. Challenges in managing and processing sensor data have to be addressed properly. This chapter briefly introduces sensor data fusion techniques and divides them into pre-processing, data mining and post-processing. Event detection is one of the main purposes of sensor data fusion. This chapter categorizes event detection approaches into threshold-based and tempo-spatial pattern based and presents the sensor state model based implementation of these two approaches. Sensor-net database is closely relevant to the sensor data fusion. The concept of the sensor-net database is introduced at the end of this chapter. This chapter has omitted the details of sensor data mining and sensor data query as they could be developed based on traditional data mining and data query technologies.

References

Allison, P.D.: Missing Data Thousand Oaks. Sage Publications, CA (2001)

Akcan, H., Brönnimann, H.: A new deterministic data aggregation method for wireless sensor networks. Elsevier J. Sig. Process. **87**(12), 2965–2977 (2007)

Basu, S., Meckesheimer, M.: Automatic outlier detection for time series: an application to sensor data. Knowl. Inf. Syst. **11**(2), 137–154 (2007)

Bontempi, G., Borgne, Y. L.: An adaptive modular approach to the mining of sensor network data. In: Proceedings of 1st International Workshop on Data Mining in Sensor Networks as part of the SIAM International Conference on Data Mining (Newport Beach, CA, 21–23 April 2005), pp. 3–9. SIAM Press (2005)

Chok, H., Gruenwald, L.: An online spatio-temporal association rule mining framework for analysing and estimating sensor data. In: Proceedings of the 2009 International Database Engineering and Applications Symposium, pp. 217–226. Cetraro, Calabria, Italy (2009)

Chu, F., Wang, Y., Parker, D.S., Zaniolo, C.: Data cleaning using belief propagation. In: Proceedings of the 2nd international workshop on Information quality in information systems, pp. 99–104. Baltimore, Maryland, (2005)

Elnahrawy, E., Nath, B.: Cleaning and querying noisy sensors. In: Proceedings of 2nd ACM International Conference on Wireless Sensor Networks and Applications, pp. 78–87. San Diego, CA, USA, (2003)

Govindan, R., Hellerstein, J., Hong, W., Madden, S., Franklin, M., Shenker, S.: The sensor network as a database. Technical Report 02-771, Computer Science Department, University of Southern California (2002)

Halatchev, M., Gruenwald, L.: Estimating missing values in related sensor data streams. In: Proceedings of the International Conference on Management of Data, pp. 83–94. Goa, India (2005)

Han, J., Kamber, M., Pei, J.: Data mining concepts and techniques. Morgan Kaufmann, MA, USA (2011)

Jeffery, S. R., Alonso, G., Franklin, M. J., Hong, W., Widom, J.: A pipelined framework for online cleaning of sensor data streams. In: Proceedings of the 22nd International Conference on Data Engineering, pp. 140–143. Atlanta, GA (2006)

Kim, C.H., Park, K., Fu, J., Elmasri, R.: Architectures for streaming data processing in sensor networks. In: Proceedings of the 3rd ACS/IEEE International Conference on Computer Systems and Applications, p. 59. Washington, DC (2005)

Moon, T.K.: The expectation maximization algorithm. IEEE Sig. Process. Mag. **13**, 47–60 (1996)

Mukherji, A., Rundensteiner, E.A., Brown, D.C., Raghavan, V.: SNIF TOOL: Sniffing for patterns in continuous streams. In: Proceedings of the 17th ACM Conference on Information and Knowledge Management, pp. 369–378. Napa Valley, California, USA (2008)

Santini, S., Römer, K.: An adaptive strategy for quality-based data reduction in wireless sensor networks. In: Proceedings of the 3rd International Conference on Networked Sensing Systems, pp. 29–36. Chicago (2006)

Silberstein, A., Braynard, R., Filpus, G., Puggioni, G., Gelfand, A., Munagala, K., Yang, J.: Data-driven processing in sensor networks. In: Proceedings of 3rd Biennial Conference on Innovative Data Systems Research (CIDR), pp. 10–21. Asilomar, California (2007)

Tan, P.: Knowledge discovery from sensor data, available online at: http://www.sensorsmag.com/sensors/article/articleDetail.jsp?id=317466 (2006)

Xue, W., Luo, Q., Chen, L., Liu, Y.: Contour map matching for event detection in sensor networks. In: Proceedings of the ACM SIGMOD international Conference on Management of Data, pp. 145–156. Chicago, USA, (2006)

Yang, Y., May, A., Yang, S.H.: Sensor data processing for emergency response. Int. J. Emergency Manage. **7**(3/4), 233–248 (2010)

Yang, Y., May, A., Yang, S.H.: A generic state model with neighbourhood support from wireless sensor networks for emergency event detection. Int. J. Emergency Manage. **8**(2), 135–152 (2012)

Zhuang, Y., Chen, L., Wang, X.S., Lian, X.: A weighted moving average-based approach for cleaning sensor data. In: Proceedings of 27th International Conference on Distributed Computing Systems (ICDC'07), pp. 38–45. Toronto (2007)

Chapter 9
WSN Security

Keywords Security · Denial of service (DoS) attack

9.1 Basic Concepts of OSI Security

The ITU-T (International Telecommunication Union-Telecommunication standardization sector) Recommendation X.800 gives a security architecture for open systems interconnections (OSI), including the requirements for security and the approaches required to satisfying those requirements. It uses the following definitions:

- Security attack: any actions that compromise the security of information owned by an organisation or an individual.
- Security mechanism: a mechanism that is designed to detect, prevent, or recover from a security attack.
- Security service: a service that enhances the security of the information systems and the information transfers of an organisation. The security services make use of one or more security mechanisms to provide the service.

Security attacks can be passive attack or active attack. Passive attack aims to eavesdrop or make use of information from the information system without affecting system performance. Active attacks attempt to alter system resources, parameters or disrupt their operation or at least degrade their performance.

Security mechanisms recommended in X.800 (ITU-T 1991) include:

- Cryptography—methods for the transformation of data in order to hide its information content and prevent its undetected modification or its unauthorized use.
- Encipherment—The cryptographic transformation of data to produce ciphertext.

S.-H. Yang, *Wireless Sensor Networks*, Signals and Communication Technology,
DOI: 10.1007/978-1-4471-5505-8_9,

- Digital signature—Data appended to a data unit that allows a recipient of the data unit to prove the source and integrity of the data unit and hence protect against forgery.
- Access control mechanisms—A mechanism intended to prevent the unauthorized use of a resource, including the prevention of use of the resource in an unauthorized manner.
- Data integrity mechanisms—a mechanism intended to prevent the data being altered or destroyed in an unauthorized manner.
- Authentication exchange—A mechanism intended to ensure the identity of an entity by means of information exchange.
- Traffic padding—The generation of spurious instances of communication, spurious data units and/or spurious data within data units.
- Routing control—The application of rules during the process of routing in order to choose or avoid specific networks, links or relays.
- Notarisation—The registration of data with a trusted third party that allows the later assurance of the accuracy of its characteristics such as content, origin, time and delivery.

Security mechanisms are used to implement security services that ensure adequate security of the systems or data transfers. The X.800 Recommendation divides security services into the following categories (ITU-T 1991):

- *Authentication*—is concerning with assuring that a communication is authentic, i.e. is concerned with the verification of the identity of an entity or the verification of the source of a received message. For example, using an online banking service, both the use and the bank should be sure that the message came from the source that it claims to be from and each should be assured about the identities of the other. The most common approach for providing data authentication is to use a secret key, known only by the sender and the receiver, to encrypt data. The recipient of the data checks that the received data has been encrypted with the correct secret key and only if the data has been encrypted with the correct secret key is the data considered as being authentic.
- *Access control*—are controls that specify access to a resource, detailing the level and conditions of access and stating, if access is provided, what the users are allowing to do. In the online banking example, a user may be allowed to view his/her balance, but may not allowed to make any change or carry out any transactions for some of their accounts. Access control involves the assignment of different levels of access to entities once they have been authenticated. User name and password control is the primary approach for providing an access control service.
- *Confidentiality*—is concerned with keeping data secret from third parties who are not authorized to view the contents of any communication. A system, which implements confidentiality, must protect data from direct and indirect access by unauthorized entities. Direct access involves unauthorized entities intercepting and viewing data. Indirect access involves unauthorized entities viewing communications traffic patterns and deriving the contents of the communications,

through traffic analysis. The primary approach for providing confidentiality is through the use of encryption.

- *Integrity*—is concerned with preventing the unauthorized modification, insertion, deletion, replay of data, without detection by a legitimate user. Message encryption, message authentication codes and hash functions are three established approaches for verifying that a message has not been tampered with.
- *Non-repudiation*—is a protection against denial by one of the entities involved in a communication that they having participated in the communication. Non-repudiation can be related to both the message's origin and destination and prove that the message was sent or received by the specified party. In the online banking example, non-repudiation protects against the situation where a user has made a transaction, but later denies it. Digital signature and notarization are two primary approaches used to provide non-repudiation services.
- Service availability—protects a system to ensure its availability and particularly focuses on denial-of-service (DoS) attacks. Service availability focuses on the protection of service providing systems from attackers that attempt to overwhelm the resources of these systems in an attempt to either permanently remove the availability of the service, or to intermittently to degrade it to a sufficient level, thus removing it from availability. There are numerous methods employed by malicious users to adversely affect the availability of services. However, such methods are collectively termed DoS attacks.

Table 9.1 presents the attacks and the security services required to defending against those attacks. It can be seen that at least one security service is designed to defending any particular attack.

9.2 Unique Challenges in WSN Security

Wireless sensor networks (WSNs) have many unique features that differ from wired or wireless ad hoc networks and other types of wired and wireless networks. When considering security in sensor networks, security techniques used in

Table 9.1 Attacks and security services

Security service	Attack					
	Release of message contents	Denial of participation	Masquerade	Replay	Modification of message	Denial of service
Authentication			√			
Access control			√			
Confidentiality	√					
Integrity				√	√	
Non-repudiation		√				
Service availability						√

traditional networks cannot be directly applied (Perrig et al. 2004). Firstly, a large-scale sensor network may consist of thousands of sensor nodes. Also, typical sensor nodes are limited in their energy, computation, and communication capabilities in order to make sensor networks economically feasible. Secondly, wireless sensor nodes may be dispersed over a wide geographical area, presenting the additional risk of physical attack. And thirdly, sensor networks interact with people and environments, posing new security risks.

End-to-end securities such as message authenticity, integrity, and confidentiality, are usually achieved in traditional networks by an end-to-end security mechanism such as the secure socket layer (SSL). This approach requires a robust key exchange and distribution scheme. As a resource-limited network, the key exchange and distribution scheme for WSNs must be simple to execute and feasible for the limited sensor node memory and computation capability. The large number of communicating nodes makes end-to-encryption impractical, since sensor node hardware cannot normally store a large number of unique encryption keys. Furthermore, the dominant traffic pattern in WSNs is many-to-one, i.e. many sensor nodes to a single sink node or base station. Data aggregation, duplicate elimination, or data compression, as described in Chap. 8, requires intermediate nodes to access, modify, and possibly suppress the contents of messages, in order to reduce communication cost. It is unlikely therefore, that an end-to-end security mechanism between a sensor node and a base station can be used to guarantee message security (Du and Chen 2008). For a large-scale wireless sensor network, it is impractical to protect each individual sensor node from either physical or logical attack. Being dispersed over a large area furthermore exposes sensor nodes to adversaries who can capture and reprogram individual sensor nodes. Adversaries can also induce the WSN to accept their own sensor nodes as legitimate nodes. Once having deploying a small number of malicious nodes inside the WSN, the attacker could then launch attacks from inside the WSN (Chan and Perrig 2003). Interaction with people and environments make direct site surveillance possible. For example, a few wireless receivers placed outside a house or a sensor field might be able to monitor the situation inside the house or sensor field, thus revealing detailed information. An attacker can gain access to sensitive information either by accessing stored sensor data or by querying or eavesdropping on the WSN. Remote access to the WSN also allows an attacker to potentially access large volumes of information without being physically present to maintain surveillance.

In summary, because WSNs pose the above unique security challenges, specially designed security techniques are required.

9.3 Classifications of Security Attacks on WSNs

There are various ways to classify attacks on WSNs such as by attacker locations, attacker capabilities, protocol layers, and the purpose of the attacks.

Attacks on WSNs can be classified into outside and inside attacks. An outside attacker has no access to most cryptographic materials in the sensor networks, while an inside attacker may have partial key materials and the trust of some other sensor nodes. Insider attackers can introduce to the network their own sensor nodes and induce the WSN to accept them as legitimate nodes, or they can claim multiple identities for an altered node once they have captured and reprogramed these nodes. Once in control of a few nodes inside the WSN, the insider attacker can launch a variety of attacks from inside the WSN. The typical inside attacks are falsification of sensor data, extraction of private sensed information from sensor network reading, and denial of service. Inside attacks are much harder to detect and defend against (Du and Chen 2008). Outside attackers need to penetrate a gateway before approaching the WSN. A mains powered gateway is much more reliable and capable of defending against outside attacks.

Attacks can also be classified based on the capability of the attacker, such as being at sensor-level or laptop-level. A powerful laptop-level attacker can do much more harm to a WSN than a malicious sensor node, since it has a much larger power supply and computation and communication capabilities than a battery driven sensor node. More frequently attacks on WSNs are classified according to the network layers. This classifies events as attacks on physical, link, network, transportation, or the applications layers. Wood and Stankovic (2002) classified various DoS attacks on WSNs according to network layers and gave a table that lists the layers of a typical sensor network and describes each layer's vulnerabilities and defence. Each layer is vulnerable to different types of DoS attacks and has different options available for its defence. Some attacks cut across multiple layers or exploit the interactions between them. Du and Chen (2008) summarized typical attacks on sensor networks and possible defence techniques. Table 9.2 presents the name of the attack, the corresponding network layer, and possible defence techniques.

In Table 9.2, jamming, tampering, collision, exhaustion, unfairness, neglect and greed, homing, sinkhole, flooding, and de-synchronization can be grouped as Denial of Service attacks as all of them diminish or eliminate a network's capacity to perform its expected function. Key management is the main part of any security service. The next section will introduce key management in ZigBee WSNs. Then, in order to keep the size of this chapter within limits, the rest of the chapter will concentrate on outside DoS attacks and in-door applications of WSNs,.

9.4 ZigBee Security Services

ZigBee security services include methods for key establishment, key transport, frame protection, and device management. It can be in either the standard mode for ZigBee and ZigBee PRO or in a high-security mode for ZigBee PRO. ZigBee security design follows the principle that 'the layer that originates a frame is responsible for initially securing it'. For example, if a NWK command frame

Table 9.2 Sensor network layers and DoS defences

Network layer	Attacks	Defences
Physical	Jamming	Spread-spectrum, priority message, lower duty cycle, region mapping, mode change
	Tampering	Tamper-proofing, hiding
Link	Collision	Error-correcting code
	Exhaustion	Rate limitation
	Unfairness	Small frames
Network	Neglect and greed	Redundancy, probing
	Manipulating routing information	Authentication, encryption
	Sybil attack	Authentication
	Wormhole attack	Monitoring, flexible route selection
	Selective forwarding attack	Redundancy, probing
	Homing	Encryption
	Misdirection	Egress filtering, authorization, monitoring
	Sinkholes	Authorization, monitoring, redundancy
Transport	Flooding	Limiting connection numbers, client puzzles
	De-synchronization	Authentication
Applications	Clone attack	Unique pair-wise keys

needs protection, then the NWK layer's security should be used. ZigBee uses Counter Mode (CTR) encryption plus Cipher Block Chaining (CBC) message integrity code (MIC) with the 128-bit advanced Encryption Standard (AES) algorithm (Elahi and Gschwender 2009).

9.4.1 Cryptography Used in ZigBee Security

Cryptograph is the term for keeping information secure by encoding and decoding information, often termed encryption and decryption. The Original message is referred to as plaintext, and the encrypted text is called the ciphertext. The plaintext is encrypted by an encryption algorithm and an encryption key. The ciphertext is then transmitted over the communication channel. At the receiver's end, the ciphertext is decrypted by a decryption algorithm and a decryption key. Figure 9.1 shows the concept of the cryptography.

There are two types of cryptography, one is symmetric key cryptography, and the other one is asymmetric key cryptography. In symmetric key cryptography the transmitter and the receiver of a message share the same key for both the encryption and decryption of the data. In asymmetric key cryptography, unlike symmetric key cryptography the transmitter and the receiver of a message use two

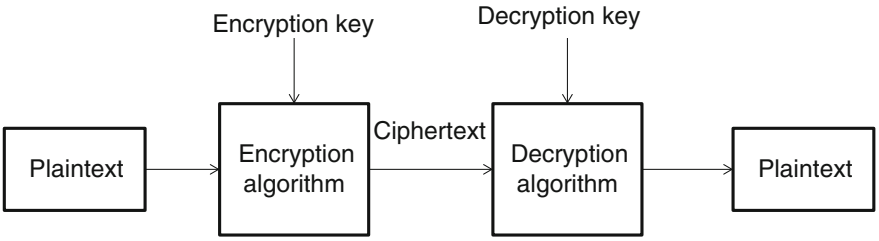


Fig. 9.1 Cryptography process

separate keys for the encryption and decryption of data. A block cipher is a group of bits that are encrypted simultaneously; a stream cypher is a bit by bit encrypted.

9.4.1.1 Advanced Encryption Standard (AES)

Advanced Encryption standard (AES) is a block cipher that uses a 128, 192, or 256-bit key for encryption. The ZigBee security uses a 128-bit (4×4 array of bytes $= 4 \times 4 \times 8 = 128$ bits) encryption key. Figure 9.2 shows the 128-bit key and cipher state.

The AES encryption process consists of 4 steps—substitution, shift row, mix column and add round key, as shown in Fig. 9.3. The process is repeated 10 times before completion.

Step 1: Each byte in the cipher state is substituted by another byte using a conversion table, called S-Box table.

Step 2: perform the following operations of shifting row on each row of the resulting state matrix from Step 1, as shown in Fig. 9.4.

- No shift on first row
- Circulate shift left 1 byte on the second row
- Circulate shift 2 bytes on the third row
- Circulate shift 3 bytes on the fourth row

Step 3: The matrix resulting from Step 2 is multiplied by a given constant matrix, as shown in Fig. 9.5.

Fig. 9.2 128-bit key and cipher state

$K_{0,0}$	$K_{0,1}$	$K_{0,2}$	$K_{0,3}$
$K_{1,0}$	$K_{1,1}$	$K_{1,2}$	$K_{1,3}$
$K_{2,0}$	$K_{2,1}$	$K_{2,2}$	$K_{2,3}$
$K_{3,0}$	$K_{3,1}$	$K_{3,2}$	$K_{3,3}$

$S_{0,0}$	$S_{0,1}$	$S_{0,2}$	$S_{0,3}$
$S_{1,0}$	$S_{1,1}$	$S_{1,2}$	$S_{1,3}$
$S_{2,0}$	$S_{2,1}$	$S_{2,2}$	$S_{2,3}$
$S_{3,0}$	$S_{3,1}$	$S_{3,2}$	$S_{3,3}$

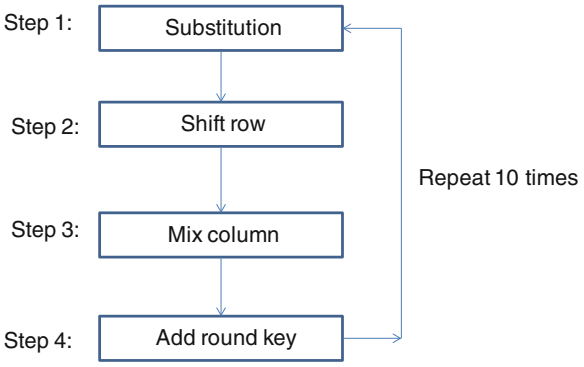


Fig. 9.3 AES encryption process

Fig. 9.4 Row shift operation

$S_{0,0}$	$S_{0,1}$	$S_{0,2}$	$S_{0,3}$
$S_{1,1}$	$S_{1,2}$	$S_{1,3}$	$S_{1,0}$
$S_{2,2}$	$S_{2,3}$	$S_{2,0}$	$S_{2,1}$
$S_{3,3}$	$S_{3,0}$	$S_{3,1}$	$S_{3,2}$

$$\begin{vmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{vmatrix} \times \begin{vmatrix} S_{0,0} & S_{0,1} & S_{0,2} & S_{0,3} \\ S_{1,1} & S_{1,2} & S_{1,3} & S_{1,0} \\ S_{2,2} & S_{2,3} & S_{2,0} & S_{2,1} \\ S_{3,3} & S_{3,0} & S_{3,1} & S_{3,2} \end{vmatrix} = \begin{vmatrix} A_{0,0} & A_{0,1} & A_{0,2} & A_{0,3} \\ A_{1,0} & A_{1,1} & A_{1,2} & A_{1,3} \\ A_{2,0} & A_{2,1} & A_{2,2} & A_{2,3} \\ A_{3,0} & A_{3,1} & A_{3,2} & A_{3,3} \end{vmatrix}$$

Fig. 9.5 Mix column operation

Step 4: Each element of the resulting matrix from Step 3 is XORed with its corresponding element in the 128-bit key matrix, as shown in Fig. 9.6.

9.4.1.2 Counter Mode Encryption

Counter mode encryption follows the following operations, as shown in Fig. 9.7. T_1 is a value generated by a counter, $T_2 = T_1 + 1$, $T_3 = T_2 + 1, \ldots$, and $T_n = T_{n+1} + 1$. EK is a 128-bit AES encryption key.

- The message is divided into blocks $D_1, D_2, \ldots$, and D_n;
- $T_1, T_2, \ldots$, and T_n are encrypted using a 128-bit AES encryption key (EK), and
- The results $I_1, I_2, \ldots$, and I_n are XORed with $D_1, D_2, \ldots$, and D_n respectively.
- The outputs $P_1, P_2, \ldots$, and P_n are the encrypted message.

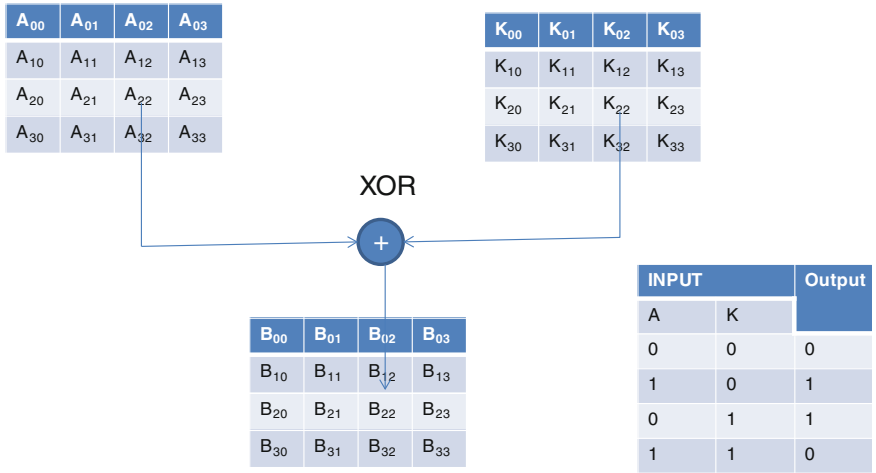


Fig. 9.6 XOR operation

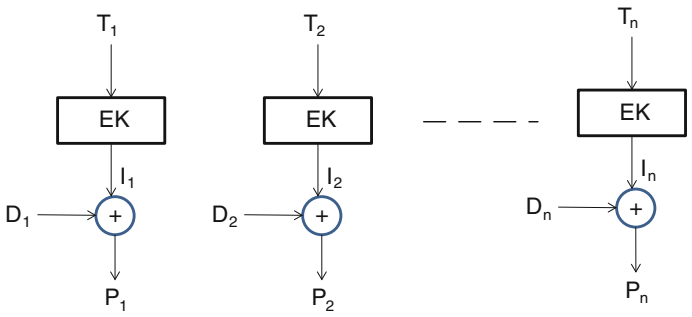


Fig. 9.7 AES counter mode encryptions

9.4.1.3 Cipher Block Chaining Mode Encryption

CBC mode encryption is used for generating a message integrity code (MIC) in the ZigBee security services for maintaining data consistency. As shown in Fig. 9.8, the message is divided into 128-bit blocks. The first block D_1 is encrypted using the 128-bit AES encryption key, the cipher text I_1 is XORed with the next block of data D_2. This process is repeated until the message is fully encrypted. I_n is the MIC and added when necessary, to the ZigBee frame where n is the number of 128-bit blocks of the message before CBC encryption.

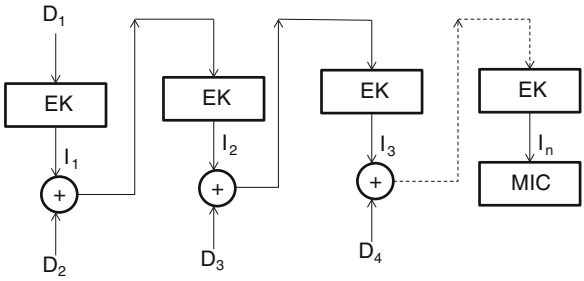


Fig. 9.8 AES CBC mode encryptions

9.4.2 ZigBee Security Keys and Trust Centre

Security amongst a ZigBee WSN is based on 'link' keys, a 'network' key, and a 'master' key. The ZigBee trust centre, called the security service provider, is designed to manage network security and key distribution. The ZigBee trust centre is located at the network layer and the application support sub-layer in the ZigBee stack, as shown in Fig. 9.9.

- Link keys: A link key is used by the application protocol sub-layer to secure Unicast communication between two ZigBee devices. There are two types of link keys:
  - Application link key: this key is used for the security of application data between two devices and it is shared only between two devices. A device acquires link keys either via key-transport, key-establishment, or pre-installation.
  - Trust centre link key: this key is used by the trust centre and devices on the network for securing communication between the trust centre and devices. This key is preconfigured in the devices.
- Network key: The network key is used for broadcast communications by the network layer. All devices in the network share a common network key because all the devices must be able to decipher these broadcast messages. This key is installed by the manufacturer or transported by the trust centre.
- Master key: The master key is used for the generation of the link key. This key may be installed by the manufacturer of by a trust centre.
- Trust centre: The trust centre is the device trusted by all devices within a network to distribute keys for the purpose of network and end-to-end application configuration management. All members of the network shall recognize exactly one trust centre, and there shall be exactly one trust centre in each secure network. The ZigBee specification defined the roles performed by the trust centre into three sub-roles: trust manager, network manager, and configuration manager. 'A device trusts its trust manager to identify the device(s) that take on the role of its network and configuration manager. A network manager is responsible for the network and distributes and maintains the Network key to the devices it manages. A configuration manager is responsible for binding two

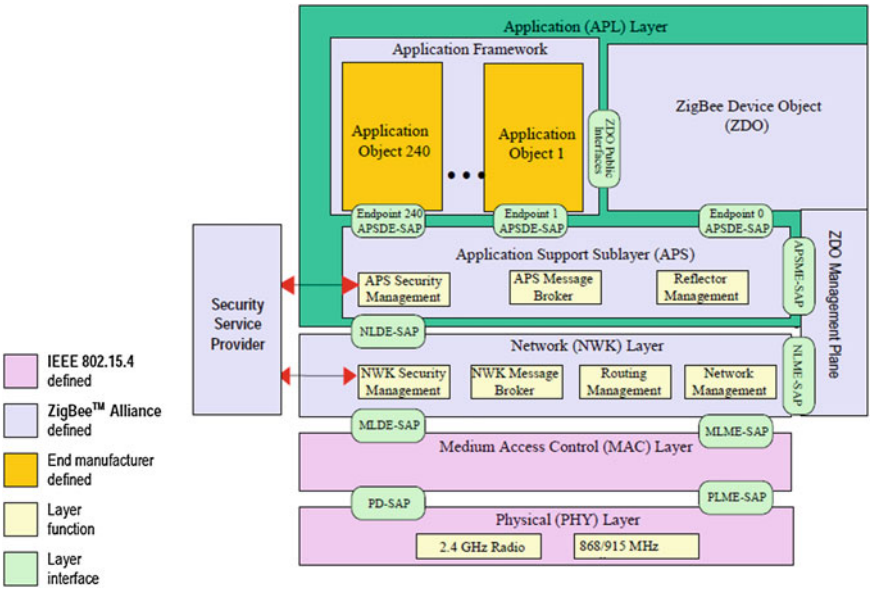


Fig. 9.9 ZigBee stack (ZigBee Alliance 2005)

applications and enabling end-to-end security between devices it manages (e.g., by distributing master keys or link keys). To simplify trust management, these three sub-roles are contained within a single device—the trust centre' (ZigBee Alliance 2005).

9.4.3 Key-Transport and Key-Establishment

Key-transport can be done in secured or unsecured modes. In the secured key-transport mode, the trust centre handles the transport of the link, network, and master keys to the various devices. In the unsecured key-transport mode, the devices are loaded with the keys. The trust centre performs the following functions in the secured key-transport mode:

- Network key transport

The trust centre and devices in the WSN use the preconfigured trust centre link key to transmit a new or active network key from the trust centre to the devices, as follows:

– When a new device joins the WSN, the trust centre encrypts an active network key and transmits it to the device;

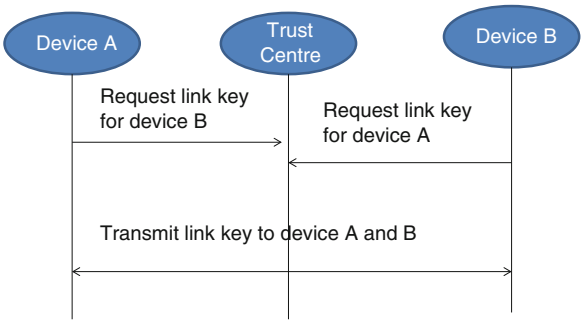


Fig. 9.10 Trust centre generating and transmitting the application link key

  - If a device requests an update of the network key, the trust centre encrypts a new network link key with the trust centre link key and transmits it to the device.

- Application link key transport

The trust centre can generate the application link key for secured communication of application data between two devices after receiving a request from each device. The generated application link key is encrypted with the preconfigured trust centre link key and then the trust centre transmits the encrypted application link key to the devices. The process is shown in Fig. 9.10.

Key-establishment is used to generate link keys for two devices using their master keys. The high-security mode of ZigBee uses the Symmetric-Key Key Establishment (SKKE) protocol for generating the link key. However, the two devices must have already installed a master key in order to use the SKKE protocol. Key establishment involves two application support sub-layer (APS) management entities, an initiator device and a responder device, and is preceded by a trust-provisioning step. Trust information encrypted with a master key provides a starting point for establishing a link key and can be provisioned in-band (i.e. using the normal communication channels) or out-band (i.e. using a different channel than the normal communication channels used by the network). Once trust information is provisioned, a key-establishment protocol involves three conceptual steps: the exchange of ephemeral data, the use of this ephemeral data to derive the link key, and the confirmation that this link key was correctly computed.

9.5 Typical Existing Approaches for DoS Defences

There are numerous approaches for protecting resource rich servers or gateways from DoS attacks (Ricciato et al. 2010), which can be classified as victim based approaches, attacker based approaches, and hybrid approaches based on the location of the defence approach (Mirkovic 2003). Victim based defence

approaches tackle DoS attacks and reduce the impact from the victim-end through resource multiplication (Chiba et al. 2006) or normally by moving the connections from the victim connection buffer to a larger system resource available to established connections (Schuba et al. 1997). However, the potential for all of the servers to have their resources exhausted by a large DoS attack remains. Attacker based defence approaches protect the victims from the attacker-end. Typical examples in this category are Multi-Level Tree for Online Packet Statistics (MULTOPS (Thomer and Massimiliano 2001), DoS netWork Attack Recognition and Defence (D-WARD) (Mirkovic et al. 2003), Message-specific puzzle (Ning et al. 2008), and X-TESLA- an extendable broadcast authentication scheme (Kwon and Hong 2010). For example, D-WARD operates on a router between a private network and the Internet, preventing machines on the private network from sending DoS attack packets. D-WARD analyses incoming traffic and detects DoS attacks through the detection of "nonresponsive hosts", where all packets for a particular IP address are outgoing. Additionally, ingress filtering is used to ensure all outgoing packets have valid subnet addresses, and prevent IP spoofing. Hybrid defence approaches tackle the attacks from both the victim-end and source-end of attacks such as aggregate congestion control (ACC) (Ratul et al. 2002).

Analysing the details of the above existing approaches for DoS defence, two abstract functions are found. Firstly, the existing approaches for DoS defence detect the onset of DoS attacks and attempt to distinguish between legitimate network traffic and network traffic arising from a DoS attack. Secondly, the existing approaches attempt to mitigate any detected DoS attacks. The primary difficulty in dealing with DoS attacks arises from the difficulty in distinguishing between legitimate network traffic and network traffic from DoS attacks. Due to this difficulty, a percentage of attack traffic is often misclassified by the existing approaches for DoS defence as legitimate traffic and allowed to reach the victim, i.e. the existing approaches do not effectively filter out all attack traffic during a DoS attack. The effectiveness of three commonly used existing approaches for DoS defence mentioned above is summarized in Table 9.3 from (Mirkovic 2003; Thomer and Massimiliano 2001; Ratul et al. 2002). As highlighted in Table 9.1, the D-WARD attack tool is the most effective approach for DoS defence identified from a literature review, removing 99.4 % of attack traffic. However, it is almost unavoidable to prevent a low-level stream of misclassified attack traffic from reaching the victim.

In terms of DoS defence for relatively resource rich computers, the existing approaches for DoS defence provide a satisfactory level of protection. The small amount of attack traffic reaching the victim is insufficient to cause a disruption to services for legitimate users. However, low-level DoS attacks targeting the resources at the point of ingress between a sensor network and the Internet, which here is named the Gateway, are sufficient to prevent remote users from communicating with the WSN. Moreover, the low-level DoS attack traffic penetrating the Gateway, if it occurs, is sufficient to flood the sensor network's limited bandwidth resources and quickly exhaust the scarce, non-renewable, power sources of the intermediate wireless sensor nodes. The objective of the rest of this chapter is to design,

Table 9.3 Comparison of typical DoS defence tools

DoS defence tool	Percentage of DoS attack packets removed (%)
D-WARD	99.4
ACC	64
MULTOPS	93

implement and evaluate a third party based approach to work alongside any existing DoS counter measures for mitigating low-level DoS attacks targeted at WSNs. We use a WSN based home automation system as a case study.

9.6 Preventing Low-Level Denial of Service Attacks on WSN Based Home Automation Systems

Home automation systems (HAS) are the introduction of technology within the home to enhance the quality of life for its occupants, through the provision of different services such as tele-health, multimedia entertainment, and energy conservation. Home automation systems also enable the monitoring and control of domestic appliances for home users' comfort and efficient home management. The recent trend in research and industry is towards the development of WSN based HASs. Existing resource-rich approaches for defending against DoS attack remove the majority of attack traffic. However, because of their limited resources, WSN based HASs are still vulnerable to low-level DoS attacks.

This section introduces the design of a defence method to supplement existing approaches for DoS defence and to provide improved protection for resource limited, WSN based, HASs from low-level DoS attacks. The approach for DoS defence consists of three entities, the 'Virtual Home' (VH), 'Remote Home Server' (RHS), and 'DoS Defence Server' (DDS), as depicted in Fig. 9.11. There are two connection approaches between remote users, including attackers, and the HAS. The connection approach indicated by interconnections ① and ② in Fig. 9.11 is termed RHS-1. The one indicated by interconnection ③ is termed RHS-2. RHS-1 is an encrypted third-party connection, but RHS-2 is an encrypted direct connection. As shown in Table 9.3, there remains a low-level stream of misclassified DoS attack traffic able to reach the victim through secure channels. The main objective of the VH is to detect and filter out any incoming DoS attacks, preventing them from reaching the real HAS. Moreover, the VH is responsible for insuring effective communications with remote users during any DoS attacks. The virtual home incorporates a DoS attack detection mechanism and a DoS attack mitigation mechanism for these purposes. The RHS is designed as a resource rich trusted third party and used to build a mediated communication channel when a DoS occurs. The DOS defence server is hosted alongside the RHS and provides an analysis of latency for the HAS.

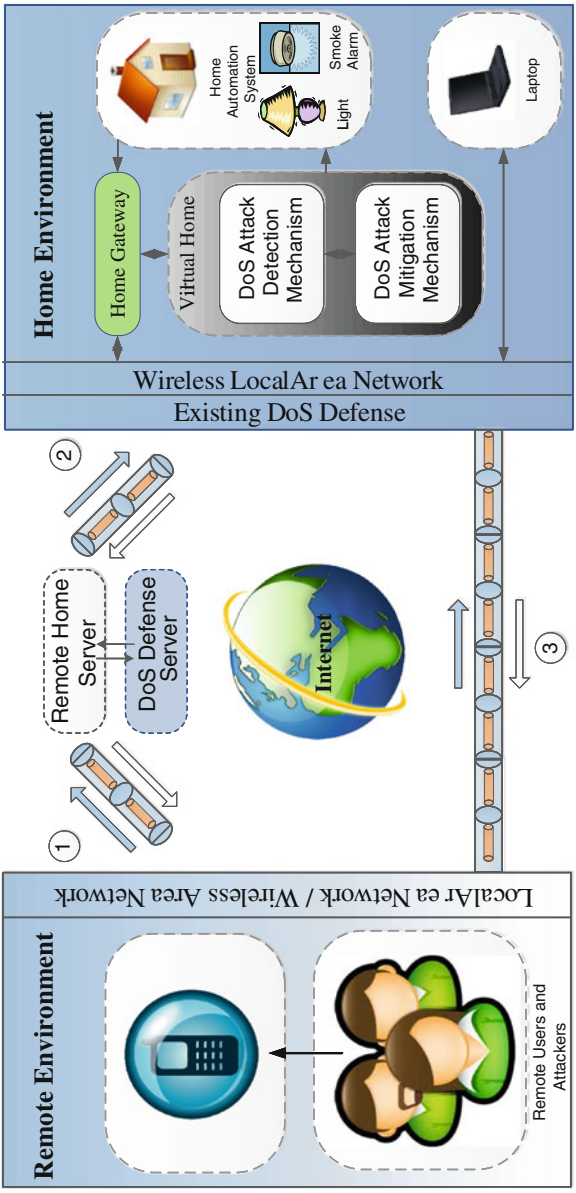


Fig. 9.11 Virtual home based approach for low-level DoS defence in WSN based HASs

9.6.1 Virtual Home: DoS Attack Monitor and Trigger

The remote communications architecture employed in the system shown in Fig. 9.11 encrypts all communications between the remote user and the HAS. End-to-end encryption is employed to protect the homeowner's privacy from malicious users. This has the advantage, in terms of DoS prevention, that all incoming attacks

Table 9.4 Key system parameters

Parameter	Definition	Explanation
DE	Number of decryption errors in a given minute	Unencrypted incoming attack packets generate decryption errors
DF	Number of decryption failures in a given minute	Encrypted packets encoded with an incorrect encryption key fail to decrypt successfully
RM	Number of replayed messages in a given minute	Replayed packets before and after a session has expired, fail to decrypt or generate freshness error

can be detected with 100 % accuracy, unless the attacker has access to the secret key used for encryption. This allows the VH to filter out the low-level DoS attack traffic before it reaches the HAS. This also prevents the low-level attack traffic from exhausting the Home Gateway's resources and eventually occupying the entire WSN's bandwidth. The DoS attack detection mechanism monitors both encrypted and unencrypted incoming packets whenever there is data in the systems input buffer and records the three parameters shown in Table 9.4 every minute. The initial values of these parameters can be set to 0 for the sake of simplicity. The DoS attack detection algorithm updates these parameters every minute. If the rule below is satisfied, a DoS attack is assumed to have started and a flag is set. The predefined threshold value of 300 is set based on experiments which show that a significant increase of access latency into the Home Gateway occurs only after the number of attacks per minute is above this value.

Rule: $If(DE + DF + RM) > 300 : FLAG$

Once the flag has been set no immediate action is taken to mitigate a potential attack. Instead, a message is sent to the RHS described below to initiate an analysis of the home gateway. If, on receipt of the analysis, results from the RHS, the analysis indicates that an attack is significantly effecting communication by legitimate users, the DoS attack response mechanism is activated. If the RHS fails to respond to an analysis request this may indicate that there is an active DoS attack of sufficient strength to block the RHS from communicating with the home gateway. Alternatively, the RHS may be offline due to a technical fault. In either case, the DoS Attack Response Mechanism is activated. The objective of the DoS Attack Response Mechanism is to overcome low-level DoS attacks that target the point of ingress between the Internet and HAS attempting to block remote users from accessing the HAS.

9.6.2 Remote Home Server and DoS Defence Server

The RHS is designed as a trusted third party. Once the system switches from the RHS-2 mode to the RHS-1 mode, i.e. the direct connection between a legitimate user, including an attacker, and the HAS (RHS-2) is disabled, the connection

between the legitimate user and the HAS is maintained via the RHS, indicated by interconnections ① and ② in Fig. 9.11. The DoS analysis requests received by the RHS are routed to a dedicated DDS. A dedicated DDS is employed to prevent any service impact on legitimate users using the RHS for communications purposes. The role of the DDS is to emulate a legitimate home automation user wishing to remotely access a HAS and calculate the average connection latency. The DDS calculates the latency, as a value of service degradation, of the relevant home gateway through repeatedly connecting to the HAS from a simulated mobile device. An average latency value is calculated for ten connection attempts, and if the service degradation exceeds a predefined threshold value set by the respective homeowners a message is sent to the respective virtual home to mitigate to the attack. Otherwise, a message is sent to the virtual home to take no action. The predefined number of connection attempts

9.6.3 Virtual Home: DoS Attack Mitigation Mechanism

The purpose of the DoS attack response mechanism is to act on the connection latency analysis performed by the DDS to mitigate detected DoS attacks and maintain effective communications between remote users and the HAS, overcoming any low-rate DoS attacks against the home gateway. As described earlier, the proposed DoS attack response mechanism makes use of two connection approaches: RHS-1 and RHS-2. RHS-2 is used under normal conditions and directly establishes a secure connection between a remote user with the HAS, as depicted in Fig. 9.11 interconnection ③. This connection approach provides the optimal communications performance. RHS-1 is used under DoS attack conditions and establishes a mediated secure connection between a remote user and the HAS via a resource rich trusted third party, denoted as the remote home server, as shown in Fig. 9.11 interconnections ① and ②. The DoS attack mitigation mechanism switches from the RHS-2 connection approach to RHS-1 under DoS attack conditions. A message from the DoS defence server triggers the virtual home to disable all support for incoming connections to the HAS and create an outgoing connection to the RHS. Remote users connect to the RHS, which creates a secure channel between the remote users and HAS. The RHS-1 approach has been shown to be 93 % slower than the RHS-2 approach (Gill and Yang 2008). However, during a DoS attack the effect of the DoS attack mitigation mechanism disabling support for all incoming connections is to terminate all incoming DoS attack connections. Only legitimate users are able to switch to the RHS-1 connection approach, authenticate with the RHS and establish a mediated secure connection with the HAS. All communications to the home automation network must traverse the outgoing connection from the HAS to the RHS. Any further direct connection requests from an attacker to the HAS are immediately dropped, shifting the focus and bottleneck of the attack away from the WSN based HAS. Therefore, the

attacker has to launch a considerably larger DoS attack against the relatively resource rich trusted RHS to be effective.

The DoS attack mitigation mechanism switches back to the RHS-2 approach, after a user defined period or the DoS attack is over. If the DoS attack has not ended the system remains at the RHS-1 communication approach. The DoS attacks check is performed periodically to ascertain when the DoS attack has ended and the RHS-1 approach can be resumed. In case the home gateway cannot establish a connection to the RHS (RHS-1 mode) a technical fault with the RHS is assumed and RHS-2 mode of communications is resumed.

9.6.4 Virtual Home Placement

As discussed in the literature review section, existing systems designed to handle DoS attacks are predominantly located at the edges of the victim's network. Recently, new systems have been proposed that reside at distributed locations across the Internet or at the edge of the victim's network. These approaches take a cart-blanche approach to filtering network traffic. Existing approaches are designed to filter all Internet traffic passing between two points. The proposed virtual home is positioned at the edge of the home automation network. As depicted in Fig. 9.11, the virtual home is installed on the home gateway, which is located between the edge of the home automation network and the homes local network used to provide access to the Internet. This point is the crucial bridge between other networks and home automation network. All the inbound or outbound home automation data traverses this connection. The location of the virtual home allows for the precise monitoring and filtering of home automation data, whilst allowing data destined for other networks to be monitored by existing DoS defences or unmonitored. This is depicted in Fig. 9.11, where a laptop user can directly connect to the Internet, without any relation to the virtual home. This approach allows for the encryption of all home automation communications and the advantages offered by encryption for DoS prevention as discussed before, whilst saving considerable computational resources spent processing and guessing attack data from normal data, at the edge of the local network at the point of ingress between the Internet and the homes local network.

This VH placement allows for the encryption of all home automation communications and provides total protection for the HAS. Unless the home's secret key is known by an attacker, no attack traffic is sent across the resource constrained WSN based HAS. The pseudo code for the virtual home is depicted in Table 9.5.

Table 9.5 Pseudo code for the virtual home

```
Step 1. Call the function to analyse the decryption results every minute for the whole time the
    application is active
Timer(60000);
Function Timer(Delay){
                analysis();
                Timer(60000);
}
Step 2. The decryption function is called whenever there is data in the systems input buffer. All
    replayed messages, decryption failures and decryption errors will be recorded during this
    process
Function decrypt(){
              rawMessage = (inputBuffer);
              message = decrypt(rawMessage);
              if(message.nonce not correctly incremented){
                        nonceReplay = nonceReplay + 1;
              } else if(message.decryption = fail){
                        decryptionFailure = decryptionFailure + 1;
              }else if(message.decryption = error){
                        decryptionError = decryptionError + 1;
              }else if(message.DDSAnalysisResult){
                        switch(message.DDSAnalysisResult);
              }
}
Step 3. This function is executed every 60,000 ms, if a potential DoS attack is detected a message
    is sent to the DDS to perform an analysis of the WSN based HAS connection latency
Function analysis(){
              int Flag = 0;
              if(nonceReplay > 0){
                        Flag = 1;
                        nonceReplay = 0;
              }if(decryptionFailure > 0){
                        Flag = 1;
                        decryptionFailure = 0;
              }if(decryptionError > 0){
                        Flag = 1;
                        decryptionError = 0;
              }if(Flag == 1){
                        Send(WSN based HAS analysis request to DDS);
              }
}
Step 4. This function chooses which communication approach the WSN based HAS is currently
    using
Function Switch(message.DDSAnalysisResult){
              if(message.DDSAnalysisResult equals "switch"){
```

(continued)

Table 9.5 (continued)

```
                    Switch to using the third party RHS-1 Approach(); //Stop support
    for incoming connections
                }else if (message.DDSAnalysisResult equals "do not switch"){
                    Do not switch communications approach(); //Continue using
    direct RHS-2 approach
                }
}
```

9.7 Implementation of Virtual Home Based Approach for Defencing DoS Attacks on WSN Based HASs

This section describes the implementation of the low-level DoS attack defence approach, discussed in the previous section, and the development of an attack tool designed to launch low-level DoS attacks and test the VH based defence approach. As depicted in Fig. 9.12 the DoS defence strategy consists of four components.

Firstly, the RHS client is installed on a mobile phone and provides the users with a graphical user interface and handles the connections with the HAS. Secondly, the Remote Home Server is installed on a resource rich trusted third party and is a crucial component of the RHS-1 and RHS-2 communications approaches. Thirdly, the DDS is hosted alongside the RHS and provides an analysis of connection latency for HAS. Fourthly, there is the home gateway, which mainly hosts the virtual home and provides the DoS detection and mitigation mechanisms.

9.7.1 RHS Client

The RHS client was implemented on a standard mobile phone as a J2ME midlet. The midlet was used to provide the TCP connectivity and user interface for the HAS. The security functions were implemented by integrating the free source Bouncy Castle API with the midlet. The use of the Bouncy Castle API allowed the use of a well-established and tested security API.

9.7.2 Remote Home Server

The RHS was implemented on a standard laptop. The RHS was coded in C# using the.NET Framework 2.0. Although the.NET Framework provides its own security libraries, resource limitations posed by the HAS, as discussed later, meant that the.NET framework libraries could not be used because they did not provide the encryption algorithms required. As a result, the C# implementation of the Bouncy Castle API was used to provide the required security functionality. A crucial

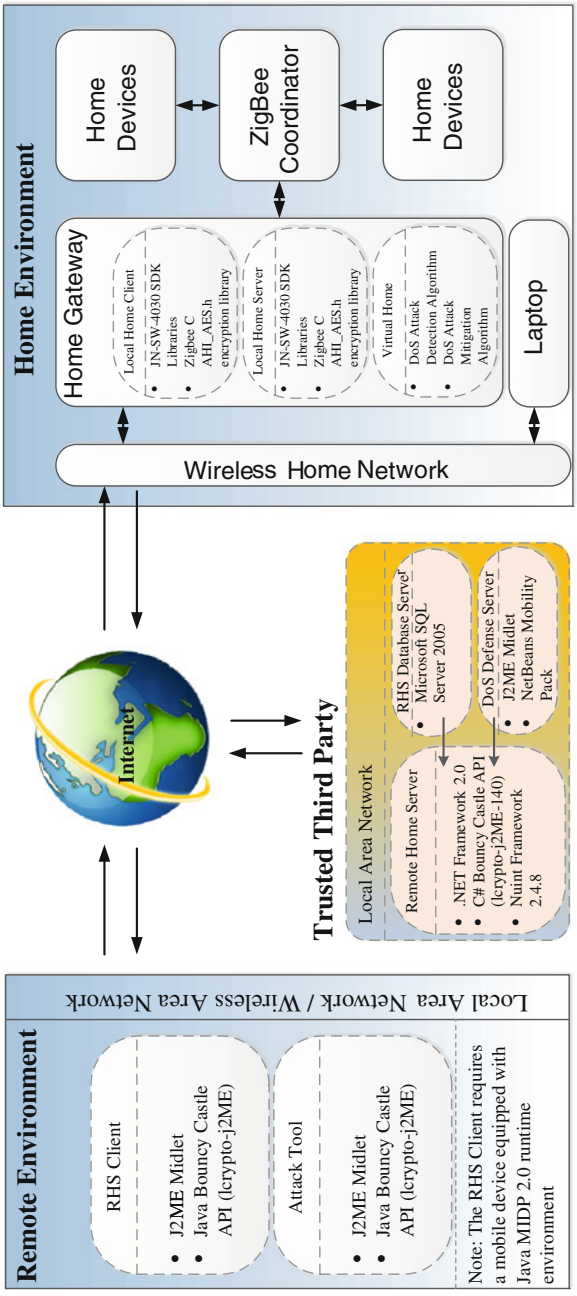


Fig. 9.12 Remote home server system architecture

component of the RHS is the RHS database server. The database maintains information on all the connected homes. The database was implemented using Microsoft SQL server 2005.

9.7.3 DoS Defence Server

A standard laptop hosts the DDS. The DDS consists of a Java midlet and a simulated mobile phone. NetBeans mobility pack 5.5 was used to emulate the mobile phone on which the midlet runs. The midlet attempts to initiate a predefined number of connections to the home gateway and calculate the average connection latency. Additionally, the midlet creates a TCP connection to the RHS to communicate the average connection latency of the respective HAS. The pseudo code for DDS is depicted in Table 9.6.

Table 9.6 DoS defence server pseudo code

```
Step 1. Call the function to analyse the respective homes connection latency
Function latencyCheck(){
        long averagelatency = 0;
        long sumLatency = 0;
        int NumberofChecks = 50;
        int counter = 0;
Step 2. Emulate a mobile user by connecting to the respective WSN based HAS and summing the
  connection latency
        while(counter < NumberofChecks){
            Thread.sleep(1,000); //adds a 1,000 ms delay between connection
  attempts
              long startTest = System.currentTimeMillis();
              long endTest = send(message); //The send message returns the system
  time when the
                                                          //send completes,
  including connection times.
                sumLatency = sumLatency + (endTest-startTest);
              counter++
}
Step 3. Calculate the average connection latency and compare the result with the predefined
  latency threshold value and inform the WSN based HAS of the findings
          averageLatency = (sumLatecy/NumberofChecks);
        If(averageLatency ≥ 3,000){
              Send(switch message to WSN based HAS);
          }else{
              Send(latency below threshold message to WSN based HAS);
          }
}
```

9.7.4 Home Gateway

The home gateway was developed in (Gill et al. 2009) to provide a low cost, mains powered, and stand-alone gateway that provides routing between the Internet and a WSN based HAS bridged with an IEEE 802.11 g compliant LAN. The Home Gateway accommodated a JN5139 ZigBee wireless communication module (one with an antenna pointing to the up-left corner in Fig. 9.13) and a WiMe web server (one with an antenna pointing to the up-right corner in Fig. 9.13). In the WSN based HAS, the Home Gateway acts as a router which allows it to communicate with the ZigBee coordinator and/or other routers within the communication rage. The low cost nature of the Home Gateway and the requirement to be integrated into the Virtual Home to guard the WSN based HAS posed significant resource limitations on the design of the home gateway. This offers the potential to evaluate the proposed DoS defence strategy with a resource limited HAS.

The Home Gateway hosted a local home client, home server and virtual home. The Home Gateway application was implemented on a single ZigBee micro-controller. The micro- controller used the ZigBee AHI_AES.h library to provide the required security functionality. Due to the resource limitations the choice of encryption algorithms and mode of operation were limited. From the available selection, the Advanced Encryption Standard (AES) encryption algorithm was chosen, operating in Counter with CBC-MAC (CCM) mode. CCM is a block cipher mode and as such provides both message confidentiality and integrity. Due to memory constraints, it was not possible to implement a separate encryption and integrity checking algorithm. Moreover, AES CCM is supported by the Bouncy Castle API for J2ME and C# used on the mobile client and RHS respectively.

Fig. 9.13 Home Gateway

The Home Gateway is responsible for switching between communications modes RHS-1 and RHS-2. The Home Gateway has the ability to start and stop services for incoming TCP connections and form an outgoing TCP connection to a predefined IP address. The home gateway does not accept UDP connections as it is not required in a resource-limited network such as WSN based HASs.

9.8 Evaluation

The evaluation of the proposed approach for low-level DoS defence was conducted in a real home environment and consisted of two stages. Firstly, the damage to a WSN based HAS caused by low-level DoS attacks penetrating the D-WARD, an existing defence approach, was investigated. Secondly, the effectiveness of the proposed approach for protecting the Home Gateway from low-level DoS attacks was investigated.

9.8.1 Attack Tool

An attack tool has been developed to launch low-level DoS attacks against the Home Gateway and the associated HAS to validate the effectiveness of such attacks against the WSN-based HAS and to evaluate the implemented defence approach.

To test the effectiveness of low-level attacks targeted at the WSN based HAS with the existing DoS defence approaches, the attack tool sends unencrypted data to the Home Gateway, which is subsequently forwarded across the WSN to the victim device. Additionally, to test the effectiveness of the proposed defence approaches for protecting the WSN based HAS from attacks targeting the Home Gateway in an attempt to prevent remote users from effectively accessing the system; the attack tool launches an application level TCP attack against the WSN. The attack tool relies on the fact that the Home Gateway allows a temporary TCP connection to be formed to receive authentication data. If the connection remains idle after the initial connection and no authentication data is received the connection is rejected. However, the attack tool attempts to exhaust the Home Gateway's TCP connection authorization mechanism, by initiating TCP connections and sending message replays of legitimate data, data encrypted with a random key and unencrypted data. The Home Gateway has to validate this authentication information before the active connection is dropped. Due to the resource limited nature of the Home Gateway legitimate users may not be able to connect while the Home Gateway is authenticating a large number of connections. The Bouncy Castle API provides the attack tool with the necessary encryption functionality to launch the attacks. The pseudo code for the attack tool is depicted in Table 9.7.

Table 9.7 Pseudo code for the attack tool

```
Step 1. Call the attack function to begin the attack
        Function Attack (String attackType)
        {
Step 2. Based on the attack type requested, generate and initialise the attack data
        String attackMessage = "";
                If(attackType == "UnencryptedDataAttack"){
                        attackMessage = UnencryptedDATA;
                }
                If(attackType == "IncorrectlyEncryptedDataAttack"){
                        attackMessage = IncorrectlyEncryptedData;
                }
                If(attackType == "CapturedDataReplayAttack"){
                        attackMessage = ReplayedEncryptedData;
                }
Step 3. Commence the attack until the attacker decides to halt the attack
                Boolean attackStatus = true;
                while (attackStatus) {
                        Open connection to victim (IP Address, Port)
                        Send(attackMessage);
                        Close connection to victim(IP Address, Port)
                        If(stopAttack){
                                attackStatus = false;
                        }
                }
        }
```

9.8.2 *Analysis of Low Level DoS Attacks on WSN Based HASs*

The WSN based HAS was connected with a local wireless WiFi network via the Home Gateway. A local laptop was also connected with the local wireless WiFi network. The WSN based HAS was converted into a star topology configuration with a ZigBee coordinator sitting in the logical centre at a home environment, as shown in Fig. 9.12. The attack tool discussed above was used to send attack data over the Internet to the WSN based HAS through the local wireless WiFi network and then to the Home Gateway, and finally to the victim device, which is the ZigBee coordinator. A simulation of the D-WARD attack tool removed 99.4 % of received attack traffic at the local wireless WiFi network before reaching the Home Gateway. Only 0.6 % attack data penetrated the Home Gateway and reached the victim ZigBee coordinator. As depicted in Fig. 9.14, at the start of the experiment, no attack data was sent to the WSN based HAS, and the packet loss rate for normal communication between the local laptop and the victim ZigBee coordinator was measured at 2.5 %. The attack rate was then increased. For each attack rate, the experiment was repeated ten times and the average packet loss rate was calculated.

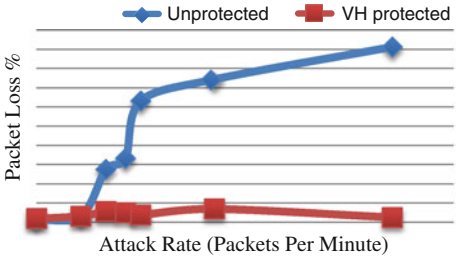


Fig. 9.14 Average percentage packet loss under different levels of DoS attacks in a ZigBee network using a star topology

It was found that for an attack rate of less than 32 packets per minute (ppm) there was little difference from the normal packet loss rate. However, when the attack rate was increased to 50 ppm, on average 26.9 % of packets between the local laptop and the victim ZigBee coordinator were lost. As the attack rate was further increased, the packet loss rate also increased. At an attack rate of 128 ppm, 73.5 % of packets were lost, rising to 86.25 % for an attack rate of 256 ppm.

The experiment was repeated with the addition of the proposed approach for DoS defence in the Home Gateway. Figure 9.14 shows that the VH effectively prevented the attack data from penetrating the Home Gateway and reaching the victim ZigBee coordinator. Figure 9.14 also shows no noticeable difference in packet loss rate in the communication between the local laptop and the victim ZigBee coordinator during the remote low-level DoS attack, with the average packet loss rate staying between 1.9 and 6 %.

The packet size during the attack was 133 Bytes. The maximum bandwidth of the ZigBee WSN was 31.25 K Byte per second (KB/s), i.e. 250 kbps (bit rate per second) (ZigBee Alliance 2005). The above experiment suggest that an attack of approximately 567 (256/60 × 133) Bytes per second caused 86.25 % of packet loss. The attack traffic only occupied 1.8 % of the ZigBee bandwidth in this case. The attack size required to allow 567 Bytes per second to bypass the existing D-WARD defence and reach the victim ZigBee coordinator at the attacker end was 94.5 (567/0.6 %/1,000) KB/s. This implies that a small low-level DoS attack (94.5 KB/s) and a low percentage of the bandwidth occupation (1.8 %) can have a greater effect on the ZigBee WSN (86.25 % of packet loss). More importantly, due to the low cost and battery driven requirements of individual WSN nodes, their memory sizes (RAM) are often around 100 KB (for example, 96 KB for the JN5139 used in this study) and the input buffer size (i.e. the residual RAM size) are often less than 50 KB after uploading the required embedded software. This limited input buffer size of the WSN nodes are insufficient to handle such large data rates leading to significant amount of packets being dropped.

9.8.3 Analysis of Low Level DoS Attacks on the Home Gateway

The previously discussed attack tool was used to generate TCP attack packets (115 bytes per connection attempt), targeted at the home gateway. The attack tool targeted an unprotected home gateway at varying rates between 0 and 1,200 attacks per minute. The results showed an average latency of 530 ms for the successful creation of a TCP connection during the presence of no attack traffic, as depicted in Fig. 9.15. A noticeable change in average latency can be seen at an attack rate of 429 attacks per minute reaching 1,802 ms. Figure 9.15 shows that at higher attack rates the average connection latency is higher. At 1,090 attacks per minute, a relatively small attack rate, there is an average delay of 7,431 ms. This level of attack, although relatively low, would lead to a substantial degradation in service for a system using the RHS-2 communications approach.

From the analysis of connection latency during the attacks against an unprotected home gateway, 3 s was chosen as the threshold value that "connection latency" would have to reach before the DDS should trigger the mitigation mechanism. The proposed virtual home and DDS were integrated into the previous experiment setup. An attack of 799 attacks per minute was implemented against the HAS, the results from which are illustrated in Fig. 9.16.

Figure 9.16 shows that from the onset of an attack (13:33) the average latency escalates quickly from 680 to 3,500 ms. At this point, the virtual home detects

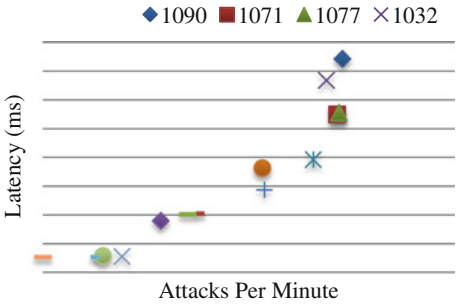


Fig. 9.15 Connection latency during differing rates of attack against an unprotected home gateway

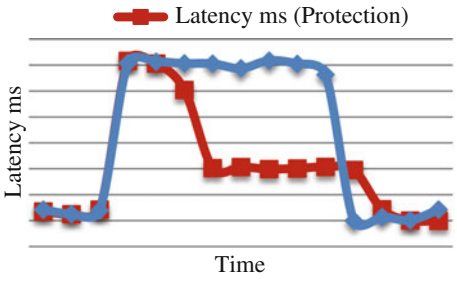


Fig. 9.16 Connection latency while VH in operation during 799 attacks per minute DoS attack

encryption errors and message replays indicating a potential attack and requests the DDS performs a check as described in Section III.B. The DDS performs the check and requests the Virtual Home switch from RHS-2 to RHS-1. This all occurs within a three-minute interval (13:33–13:35 inclusive). Once the switch from mode RHS-2 to RHS-1 has occurred (13:36) the protected HAS connection latency falls to 1,515 ms. The average connection latency is higher when routing via RHS-1 than RHS-2. Hence, for the protected VH, the latency is higher when using the RHS-1 mode to counter the attack than when using the RHS-2 mode before the attack had commenced. After a period of 5 min (13:36–13:40 inclusive) in RHS-1 mode the attack was stopped. For the experiment 5 min was chosen as the user defined period the VH would switch back approaches (RHS-1 to RHS-2) and check if the DoS has ended. Consequently, as depicted in Fig. 9.16, after 5 min the VH switches the connection from RHS-1 to RHS-2 mode and requests the DDS performs a check to see if the DoS attack has ended. In the experiment, the attack had ended so the VH continues to operate using the RHS-2 communication approach. After the attack, the average latency returned to values between 498 and 722 ms, which are similar to those measured before the attack had commenced.

9.9 Summary

There are a variety of security attacks on WSNs, which can be classified into active attacks and passive attacks, inside attacks and outside attacks, sensor-level attacks and laptop-level attacks, and attacks on different network layers. This chapter gives a brief review of possible security attacks and corresponding security defence mechanisms, and security services including ZigBee security services. Due to the resource limited nature of WSNs, there remain significant challenges in protecting WSNs from Denial of Services attacks. DoS attacks are increasing dramatically, and individual attacks are stronger and more sophisticated. It has been shown that existing approaches for DoS defence do not provide sufficient protection for WSNs, against DoS attacks that originate from relatively resource rich coexisting networks. A small amount of attack traffic penetrating existing defences may cause serious disruption to a WSN, exhaust their communication bandwidth, and prevent remote access to the WSN.

Using a WSN based HAS as an application, a defence strategy incorporating a virtual home environment, remote home server, and DoS defence server has been introduced, implemented, and evaluated. The experimental results have shown that the virtual home based defence strategy prevents low-level DoS attacks. In detail, the experiments have shown that using the virtual home based low-level DoS defence approach alongside the existing defence D-WARD, a low-level DoS attack on the WSN based HAS can be effectively detected in the virtual home environment and blocked.

In summary, the severe constraints and demanding deployment environments of WSNs make security for WSNs more challenging than other traditional networks. We have a long way from a good and general defence solution. Application

specific solutions such as the virtual home based defence strategy for HAS might take dominant roles for a decade.

References

Chan, H., Perrig, A.: Security and privacy in sensor networks. Computer **36**(10), 103–105 (2003)
Chiba, T., Katoh, T., Bista, B.B., Takata, T.: DoS packet filter using DNS information. Proceedings of 20th International Conference on Advanced Information Networking and Applications, Vienna, Austria, April 2006, pp. 6–11
Du, X., Chen, H.: Security in wireless sensor networks. IEEE Wirel. Commun. **15**(4), 60–66 (2008)
Elahi, A., Gschwender, A.: ZigBee Wireless Sensor and Control Network. Prentice Hall, NJ (2009)
Gill, K., Yang, S.H.: Secure Remote Access for Home Automation Systems', Measurement + Control, vol. 41(10), pp. 305–309 (2008)
Gill, K., Yang, S.H., Yao, F., Lu, X.: A Zigbee-based home automation system. IEEE Trans. Consum. Electron. **55**(2), 422–430 (2009)
ITU-T.: Security architecture for open systems interconnection for CCITT applications—recommendation X.800. Available at http://www.itu.int/rec/T-REC-X.800-199103-I/en (1991)
Kwon, T., Hong, J.: Secure and efficient broadcast authentication in wireless sensor networks. IEEE Trans. Comput. **59**(8), 1120–1133 (2010)
Mirkovic, J. Prier, G., Reiher, P.: Source-end DDoS defense. Proceedings of the Second IEEE International Symposium on Network Computing and Applications, Cambridge, MA, USA, April 2003, pp. 171–178
Mirkovic, J.: D-WARD: Source-end defense against distributed denial-of-service attacks. PhD Thesis in University of California (2003)
Ning, P., Liu, A., Du, W.L.: Mitigation DoS attacks against broadcast authentication in wireless sensor networks. ACM Trans. Sens. Netw. **4**(1), 1–35 (2008)
Perrig, A., Stankovic, J., Wagner, D.: Security in wireless sensor networks. Commun. ACM **47**(6), 53–57 (2004)
Ratul, M., Steven, M.B., Sally, F., John, I., Vern, P., Shenker, S.: Controlling high bandwidth aggregates in the network. ACM SIGCOMM Comput. Commun. Rev. **32**(3), 62–73 (2002)
Ricciato, F., Coluccia, A., D'Alconzo, A.: A review of DoS attack models for 3G cellular networks from a system-design perspective. Comput. Commun. **33**(5), 551–558 (2010)
Schuba, C.L., Krsul, I.V., Kuhn, M.G., Spafford, E.H., Sundaram, A., Zamboni, D.: Analysis of a denial of service attack on TCP. Proceedings of the IEEE Symposium on Security and Privacy, Oakland, CA, USA, May 1997, pp. 208–303
Thomer, M.G., Massimiliano, P.: MULTOPS: A data-structure for bandwidth attack detection. Proceedings of 10th Usenix Security Symposium, Washington, D.C., USA, August 2001, pp. 23–29
Wood, A.D., Stankovic, J.A.: Denial of service in sensor networks. IEEE Comput. **35**(10), 54–62 (2002)
ZigBee Alliance.: ZigBee specification. Available at http://www.zigbee.org. Last Accessed June 2012

Chapter 10
Mobile Target Localization and Tracking

Keywords Mobile target · Sensor node localization · Location tracking · Triangulation · Fingerprint · Centroid localization

10.1 Introduction

The inherent characteristics of WSNs make a node's location an important part of their state. For WSNs, the term location is being used to identify the physical position at which sensor readings originate. The location information is essential for most WSNs applications, where the measurement data is meaningless without knowledge of precise location from where it is obtained (Liu et al. 2012).

WSNs localization and tracking algorithms are used to estimate the location of sensor nodes in the network by using the known location of few specific nodes called beacon nodes or reference points, which can obtain their absolute positions by their placement at points with known location or by using a global positioning system (GPS). The distance between a sensor node and a beacon node can be obtained by measuring the received signal strength Indicator (RSSI) and Link Quality Indicator (LQI) which are two radio hardware link quality metrics and offer the possibility to determine distance with minimal computational effort. Measuring the time of arrival (ToA) or the time difference of arrival (TDoA) are other options in determining distance. Difficulties concerning time measurement arise from synchronization of the involved devices as well as the significant mathematical effort required to calculate the position (Blumenthal et al. 2007). Section 10.2 will give the detail of distance determination. Many localization algorithms have been proposed to provide location information of sensor nodes. Section 10.3 will introduce triangulation, fingerprinting, centroid localization, and hop counting techniques. More can be founded in the relevant literature.

S.-H. Yang, *Wireless Sensor Networks*, Signals and Communication Technology,
DOI: 10.1007/978-1-4471-5505-8_10,

Section 10.4 will introduce two possible ways to improve localization accuracy. One is to improve the distance measurement quality and another is to find a best solution in terms of the available data set. A ZigBee based multiple targets tracking method is given in Sect. 10.5 to illustrate the way in which multiple targets tracking problem can be transformed into multiple single target tracking. An underground tunnel tracking is provided in Sect. 10.6 as a case study.

Finally, there are a number of issues that should be taken into consideration in the WSNs localization and tracking algorithm design:

- **Tracking accuracy**—Different applications have different requirements on localization and tracking accuracy. It is important to develop a tracking system that meets the required accuracy.
- **Energy constraints**—Most sensor devices are battery powered and have limited energy resources. All processing, communication, and sensing actions involved in localization and tracking must be energy efficient, otherwise it will reduce the lifespan of the sensor devices.
- **Signal interference**—Interference between nodes in the same network results from collisions between packets transmitted by different nodes at the same time. This will degrade the information transmission required for localization and tracking.
- **Environmental obstacles and terrain irregularities**—These lead to inaccurate position estimation. Objects such as large rocks can obstruct the line of sight, preventing Time Difference of Arrival (TDOA) and GPS ranging, or can cause interference with radio signals.
- **Node density**—It is important to identify the algorithm's implicit density assumptions, because high node density might be prohibitively expensive or even totally unfeasible.

10.2 Distance Determination

Localization algorithms require a distance to estimate the position of unknown devices. This section introduces four possible ways to acquire a distance: received signal strength indicator (RSSI), link quality indicator (LQI), time of arrival (ToA), and time difference of arrival (TDoA).

10.2.1 Received Signal Strength Indicator

One possibility to acquire a distance is measuring the received signal strength (RSS) of the incoming radio signal. The idea behind RSS is that the configured transmission power at the transmitting device (P_{TX}) directly affects the receiving power at the receiving device (P_{RX}). According to Friis' free space transmission

equation (Rappaport 1996), the detected signal strength decreases quadratically with the distance to the sender.

$$P_{RX} = P_{TX} \times G_{TX} \times G_{RX} \left(\frac{\lambda}{4\pi d} \right)^2 \tag{10.1}$$

P_{TX} Transmission power of sender
P_{RX} Remaining power of wave at receiver
G_{TX} Gain of transmitter
G_{RX} Gain of receiver
λ Wave length
d Distance between transmitter and receiver

In embedded devices, the received signal strength is converted to a received signal strength indicator (RSSI) which is expressed as the ratio of the received power to reference power (P_{Ref}). Typically, the reference power represents the absolute value of $P_{Ref} = 1$ mW.

$$RSSI = 10 \times \log \frac{P_{RX}}{P_{\text{Ref}}}, \quad [RSSI] = \text{dBm} \tag{10.2}$$

An increasing received power results a rising the RSSI. Thus, distance d is indirectly proportional to the RSSI. An ideal distribution of P_{RX} is not applicable in practice, because the propagation of radio signals could be interfered with by a number of influencing effects, e.g.

- reflections from metallic objects;
- other Electro-magnetic fields;
- refraction by the media with different propagation velocity;
- obstacles;
- inapplicable receiving circuits.

These effects degrade the quality of the RSSI significantly. Hence in many applications the RSSI has a high variance and this significantly reduces the quality of the RSSI.

According to Eqs. (10.1) and (10.2), there exists an exponential relation between the strength of a signal sent out by a radio and the distance the signal travels. In reality, this correlation has proven to be less perfect, but it still exists. The reason is that the effective radio-signal propagation properties differ from the perfect theoretical relation that is assumed in the algorithm. The effects noted above such as reflections, fading and multipath effects can significantly influence the effective signal propagation.

10.2.2 Link Quality Indicator

Another method to determine the distance is based on the link quality indicator (LQI) of the transmission. According to IEEE 802.15.4, LQI is a characterization of the strength and quality of a received packet. It measures each successfully received packet and the resulting integer ranges from 0 to 255 (0 × 00–0 × ff), indicating the lowest and highest quality signals detectable by the receiver (−100 and 0 dBm).

Note that, LQI is only made available by IEEE 802.15.4 compliant devices and scaling the link quality to a LQI must be done by software. Signal strength and link quality values are not necessarily linked. But if the LQI is low, it is more likely that the RSSI will also be low. It therefore makes sense to make the most of the RSSI.

10.2.3 Time of Arrival

Time of arrival (ToA) is based on signal travelling time to estimate distance between two nodes. Usually, ultrasound signals are deployed in the ToA based localization systems, as shown in Fig. 10.1. ToA systems require a high accurate clock in the communication system. The distance d between transmitter and receiver can be calculated using Eq. (10.3).

$$d = \frac{((T_3 - T_0) - (T_2 - T_1)) \times v}{2} \tag{10.3}$$

where T_0, T_1, T_2, T_3, and v are the time instances and velocity of ultrasound signals respectively.

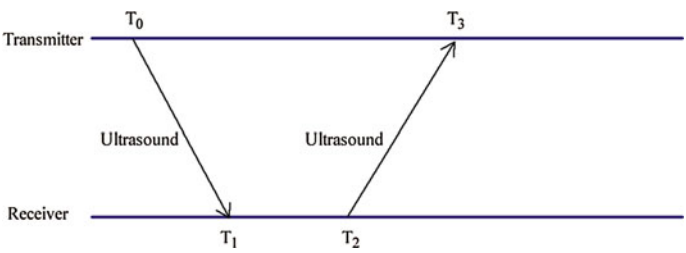


Fig. 10.1 ToA approach

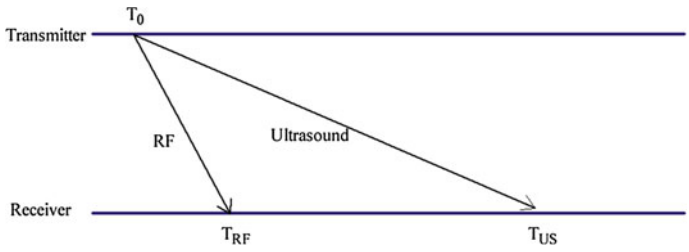


Fig. 10.2 TDOA approach

10.2.4 Time Difference of Arrival

Time different of arrival (TDoA) estimates the distance based on two radio signals travelling at different speeds such as radio frequency (RF) and ultrasound. The distance between two nodes can be estimated by measuring the difference between the transmitting time and the receiving time, as depicted in Fig. 10.2. The distance d between the transmitter and receiver can be calculated using Eq. (10.4).

$$d = (T_{US} - T_{RF}) \times \left(\frac{v_{RF} \times v_{US}}{v_{RF} - v_{US}} \right) \tag{10.4}$$

where v_{RF} and v_{US} are the travelling speeds of the RF and ultrasound signals, T_{US} and T_{RF} are the travelling times spent by the ultrasound and radio frequency respectively.

10.3 Localization Methods

Many localization algorithms have been proposed to provide location information of nodes. With regard to the mechanisms used for estimating location, these localization protocols are classified into two categories: range-based methods and range-free methods. The former is defined by protocols that use absolute point-to-point distance estimates or angle estimates for calculating the location. The latter makes no assumption about the availability or validity of such information. Because of the hardware limitations of WSNs devices, solutions in range-free localization are being pursued as a cost-effective alternative to more expensive range-based approaches. With regard to where localization computation actually occurs, these localization protocols can be classified as centralized and decentralized methods. The centralized localization involves transmitting the measurement information to a central node which then computes the target's locations. On the other hand, the decentralized localizations distribute the location computation to corresponding nodes and only require transmitting the latest coordinators of the targets to the central node. Figure 10.3 depicts the main concept for both systems.

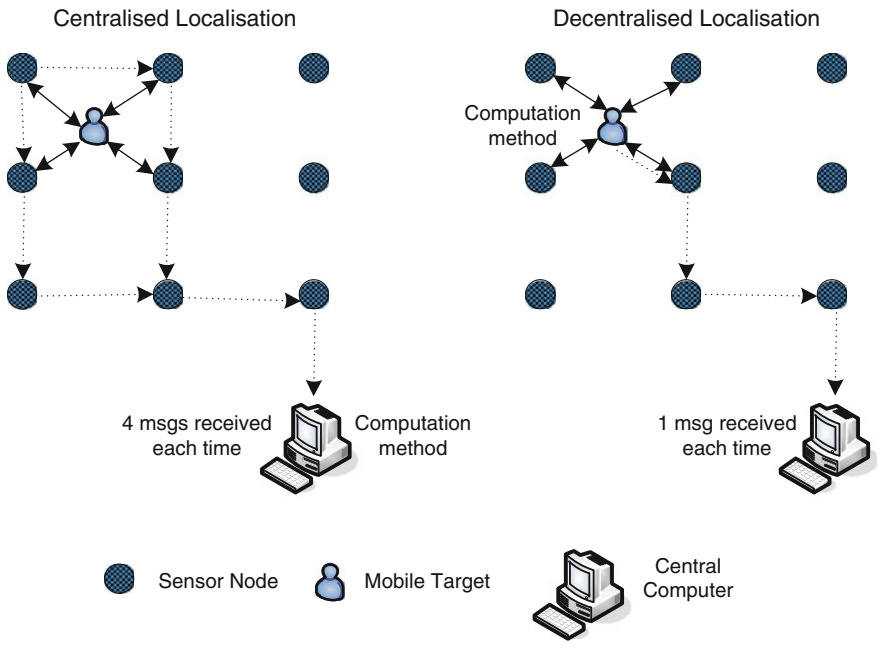


Fig. 10.3 Centralized versus decentralized localization approaches

The centralized approaches require the transmission of the localization information to a central node in order to calculate the location of the target node. Transmitting targets' localization information to a central computer is quite expensive because the power supply for each node is limited, and the long-range multi-hop data transmission is costly and usually inefficient. Consequently, lthe limited power supply available at each sensor node means that any communication with a centralized computing facility is expensive. Furthermore, sending time series data within the network introduces latency, as well as consuming energy and network bandwidth.

Decentralized localization approaches require less communication between nodes and hence reduce the power-consumption of the WSN. However, decentralized localization systems require hardware to be attached to each mobile target in order to gather the localization information from beacon nodes, compute its location, and transmit its current position to a central computer.

10.3.1 Triangulation

Triangulation is the process of determining the location of a target point by measuring distances to it from three different known points. If the distance d can be measured between 3 beacon nodes (nodes with known coordinate locations) and

the target node (the node with unknown coordinate location), then a circle with radius d can be drawn, as depicted in Fig. 10.4. The one and only one intersection point of these three circles is the location of the target node. The triangulation algorithm includes the followings:

Step 1: distribute the beacon nodes in the area of interest;
Step 2: determine the distance between each beacon node and the target node d_1, d_2, and d_3 based on the RSSI, LQI, ToA, or TDoA values;
Step 3: calculate the intersection point (the target node) between the three beacon nodes with radiuses (d_1, d_2, d_3) as follows.

We have the following three Eqs. (10.5)–(10.7) if (d_1, d_2, d_3) are known accurately.

$$(x_1 - x)^2 + (y_1 - y)^2 = d_1^2 \tag{10.5}$$

$$(x_2 - x)^2 + (y_2 - y)^2 = d_2^2 \tag{10.6}$$

$$(x_3 - x)^2 + (y_3 - y)^2 = d_3^2 \tag{10.7}$$

Combining any two of the above equations we will obtain two sets of (x, y). Using the third equation the one and only one set of (x, y) can be determined. Unfortunately, the situation will never be as simple as this. The three radiuses

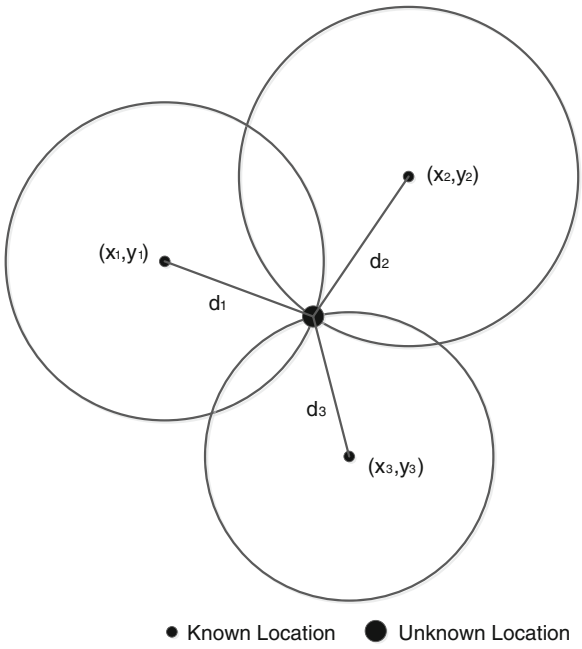


Fig. 10.4 Triangulation approach

(d_1, d_2, d_3) will never be sufficiently accurate. Therefore more effort is required to determine the location of the target node (x, y).

10.3.2 Fingerprint

The fingerprint localization method is based on the behaviour of signal propagation and information through the geometry of the tracking field, which is divided into a number of smaller grids. Location fingerprint works by determining how the signals will behave at every grid point, i.e. every corner of each grid.

Deployment of the fingerprint localization method is usually divided into two phases, as shown in Fig. 10.5. First, the offline phase: this includes measuring the location for a mobile object in different coordinates, and storing the collected information in a Database (DB). The offline algorithm includes the followings:

Step 1: distribute the beacon nodes b_1, b_2, b_n in the area of tracking;
Step 2: divide the area of tracking into several small grids and use the grid points as reference points $(x, y)_1$, $(x, y)_2$,. $(x, y)_3$, … in the tracking area;

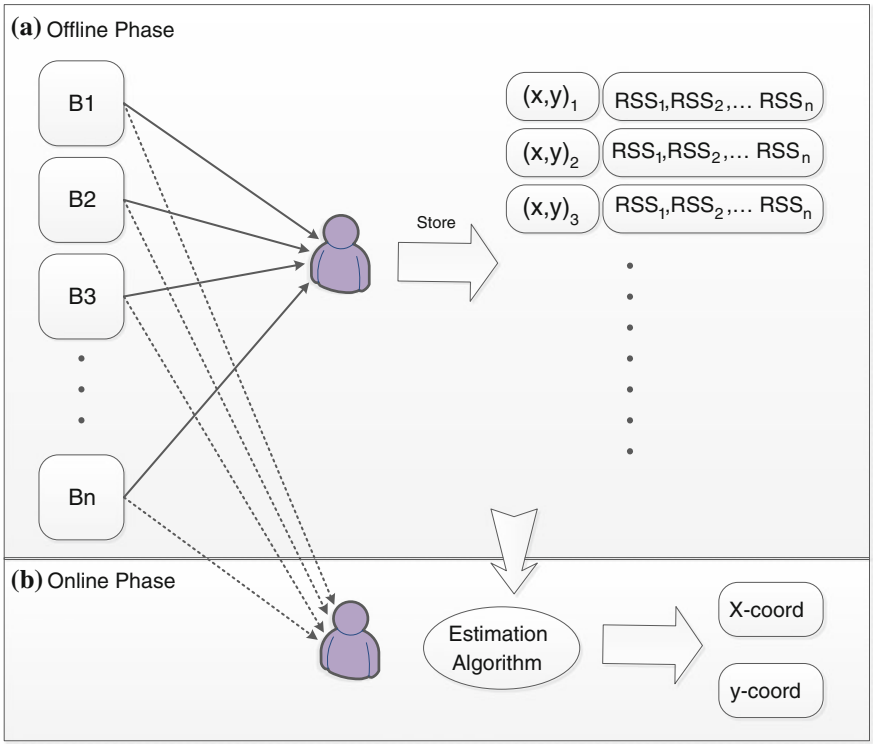


Fig. 10.5 Fingerprint approach (offline and online phase)

Step 3: get the RSS values at each reference point from beacon nodes and store them in the DB with the corresponding locations coordinates.

Next, the online phase: the mobile target collects several RSS values from different beacon nodes in its range and sends it to a server. The server applies an on-line searching algorithm to estimate the mobile object's location (Alhmiedat and Yang 2011). The online algorithm includes the followings:

Step 1: the mobile target enters the tracking area, and then collects the RSS values from each beacon node;
Step 2: compares the collected RSS values with the stored values in the DB;
Step 3: retrieve the position from the DB with the closest RSS values.

For a large tracking area the size of each grid cannot be sufficiently small because of the cost occurring in the offline phase. Therefore some error in determining the correct location of the mobile object will constantly exist.

10.3.3 Centroid Localization

Centroid localization relies on a high density of beacons, so that every target sensor node can hear from several beacons. Based on the spherical radio propagation assumption, each target node estimates its location by measuring the centre of the location of all nodes it hears as shown in Fig. 10.6. The main advantage of the centroid localization approach is that there is no need for any coordination between beacon nodes. This approach offers reasonable localization accuracy.

In the first phase, all beacons send their position $B_j(x, y)$ $(j = 1\ldots., n)$ to all target sensor nodes within their transmission range. In the second phase, all target sensor nodes calculate their own position M_i (x, y) by averaging the coordinates of all n positions of the beacons in range as shown in Eq. (10.8).

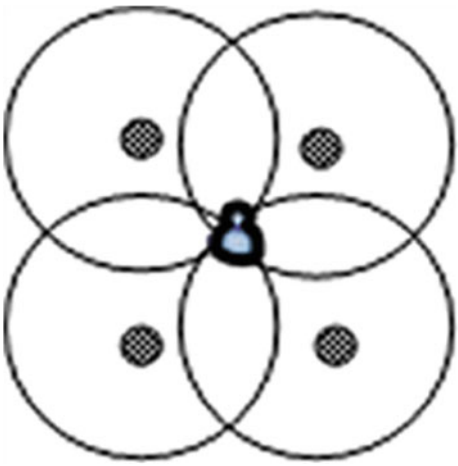


Fig. 10.6 Centroid localization approach

$$M_i(x,y) = \frac{1}{n}\sum_{j=1}^{n} B_j(x,y) \tag{10.8}$$

Weighted centroid localization approach (Blumenthal et al. 2007) introduces weight functions w_{ij} in Eq. (10.8) in order to improve the accuracy of localization. The final formula to determine the position is given in Eq. (10.9).

$$M_i(x,y) = \frac{1}{\sum_{j=1}^{n} w_{ij}}\sum_{j=1}^{n} (w_{ij} \cdot B_j(x,y)) \tag{10.9}$$

The weight functions w_{ij} depend on the distance and the characteristics of the target node receivers. Every application scenario requires a different weight function due to changed environment conditions. This is reasonable that shorter distances are weighted more than higher distances. Thus, w_{ij} and the distance d_{ij} between the target node $M_i(x,y)$ and the beacon $B_j(x,y)$ are inversely proportional. As an approximation, we can obtain Eq. (10.10).

$$w_{ij} = \frac{1}{d_{ij}} \tag{10.10}$$

To weight longer distances marginally lower, the distance is raised to a higher power of g $(g \geq 1)$. Therefore Eq. (10.10) becomes Eq. (10.11). The optimal g can be determined specifically in terms of the application scenarios.

$$w_{ij} = \frac{1}{d_{ij}^{g}} \tag{10.11}$$

10.4 Improving Tracking Accuracy

The challenges in applying the above localization methods mainly focus on dealing with the uncertainty of their distance determinations due to the changed application circumstance and the nature of radio signal propagation. This section introduces an environment factor to illustrate the dynamic change in the application environment, pre-processing RSSI approach to eliminate the outliers of RSSI values, and evolutionary optimization to best consider all the distance determinations for accuracy improvement.

10.4.1 Environment Factor

The tracking environment in which a target is located is, in most cases, dynamic. For example, in an indoor environment people walk around and furniture becomes

rearranged, while in an outdoor environment, the weather will be different from day to day. The known distance between each pair of beacons and real-time RSS measurement is defined in Eq. (10.12) to represent the environment and is called the environment factor ef_{ij} (Alhmiedat and Yang 2008).

$$ef_{ij} = \frac{RSS(B_i, B_j)}{d(B_i, B_j)} \tag{10.12}$$

where $RSS(B_i, B_j)$ represents the RSS values (RSSI or LQI) between beacon nodes B_i and B_j, and $d(B_i, B_j)$ refers to the distance between them, which is known in advance. The simplest way to obtain a unique environment factor, denoted as μ_{ef} is to average all possible individual environment factors, i.e.

$$\mu_{ef} = \frac{\sum_{j=1, i \neq j}^{n} ef_{ij}}{n} \tag{10.13}$$

where n is the total number of beacon nodes covering the target tracking area.

Figure 10.7 shows the environment factors within an environment in which three beacons (B_1, B_2, B_3) and one mobile target M are located. The unique environment factor can be used in various ways. The most straightforward way is to convert the RSSI or LQI values into a distance by dividing them by μ_{ef} as shown below in Fig. 10.7.

$$d_{B_i M} = \frac{RSS(B_i, M)}{\mu_{ef}}, \quad i = 1, 2, 3 \tag{10.14}$$

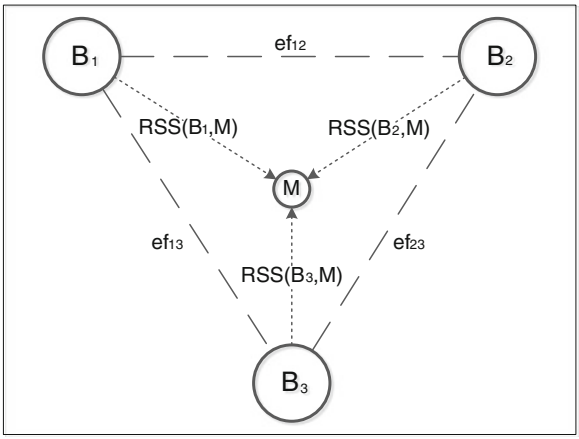


Fig. 10.7 Environment factors

10.4.2 Eliminating the Outliers of Radio Signals

Radio signals used to determine the received signal strength indicator (RSSI) or the link quality indicator (LQI) are unavoidably affected by many environment factors such as reflections, obstacle, and other electro-magnetic fields. Thus any measured RSSI and LQI will contain a number of random elements. Eliminating these noise elements will assist in improving the accuracy of the localization.

We will use RSSI values as an example here but the same process can be applied to other measurements such as LQI in localization. The Dixon method (Feng et al. 2012) is used here to eliminate the outlier of RSSI values. The standard deviation of all the RSSI values received each time is recorded as R_{SD}, and the standard deviation threshold is defined as T_{SD}. The RSSI value, noted as R_{avg}, obtained from the RSSI measurement is as follows

$$R_{avg} = \alpha R_{avg1} + (1 - \alpha) R_{avg2} \tag{10.15}$$

where

$$R_{avg1} = \frac{1}{m} \sum_{i=1}^{m} R_i, \quad R_i \leq \frac{1}{q} \sum_{j=1}^{q} R_j \tag{10.16}$$

$$R_{avg2} = \frac{1}{q-m} \sum_{i=1}^{q-m} R_i, \quad R_i > \frac{1}{q} \sum_{j=1}^{q} R_j \tag{10.17}$$

and m is the number of the RSSI values which are less than or equal to the mean of q RSSI values, α is calculated according to the following equation

$$\alpha = \begin{cases} 0.5(\frac{T_{SD} - R_{SD}}{T_{SD}} + 1), & R_{SD} \leq T_{SD} \\ 0.5(1 - \frac{R_{SD} - T_{SD}}{T_{SD}}), & R_{SD} > T_{SD} \end{cases} \tag{10.18}$$

T_{SD} depends on the concrete surroundings. The target node gets R_{SD} according to the set of received RSSI values and the value of R_{SD} is compared with T_{SD} each time. If $R_{SD} \leq T_{SD}$ the obtained RSI values are stable. From Eq. (10.18), we can calculate the value of α, $\alpha \in (0.5, 1)$. If $R_{SD} > T_{SD}$, the obtained RSSI values are unstable. According to Eq. (10.18), we can calculate the value of α, $\alpha \in (0, 0.5)$. According to Eqs. (10.15–10.17), we can obtain the RSSI values with the outlier eliminated.

10.4.3 Evolutionary Optimization

In the triangulation method shown in Fig. 10.4, the position of a target node can be calculated from the knowledge of the distance to its neighbouring beacon nodes (i.e. reference nodes or RN) and the coordinate information of those beacons:

$$(x - x_i)^2 + (y - y_i)^2 = d_i^2, \quad i = 1, 2, \ldots, N \tag{10.19}$$

where (x, y) is the coordinates of the target node; (x_i, y_i) is the coordinates of the ith beacon node; d_i is the distance between the target node and the ith beacon node; and N is the number of the beacon Nodes.

In the absence of noise in a system, each distance measurement specifies a circle for the possible positions of the target node, and the intersection of those circles determines the one and only one target position. This geometric technique, called triangulation, yields ambiguous solutions in the presence of noise in the system, since the circles defined by Eq. (10.19) may intersect at multiple points due to erroneous distance determination, as shown in Fig. 10.8.

Consequently, the localization problem becomes a searching problem, which searches an estimation of the target node coordinates $(\hat{x}, \hat{y})$ to minimize the difference between the calculated distances and the measured distances. A popular statistical localization algorithm is the nonlinear least squares (NLS) techniques, by which the location of the target node is calculated as follows:

$$(\hat{x}, \hat{y}) = \arg\min_{(X,Y)} f(x, y) = \arg\min_{(x,y)} \sum_{i=1}^{N} \beta_{\mathbf{i}} \left(\sqrt{(x - x_i)^2 + (y + y_i)^2} - d_i \right)^2 \tag{10.20}$$

where $f(x, y)$ is the cost function, N is the number of the beacon nodes, and β_i represents a weighted coefficient for the ith measurement, which commonly reflects the reliability of the measurement. The solution of Eq. (10.20) usually requires numerical search methods such as the steepest descent or the Gauss–Newton techniques (Cheng et al. 2005), which can have high computational complexity and typically requires a good initial value in order to avoid converging to a local minima of the cost function. Alternatively, linear least square estimation (LLSE) can provide suboptimal location estimation with low computational complexity, but suffers from poor accuracy.

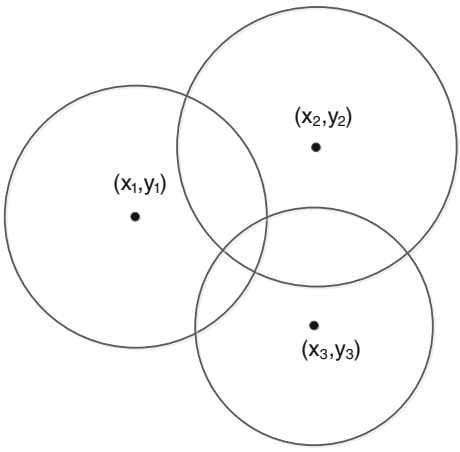


Fig. 10.8 Triangulation yields multiple intersections due to the distance measurement error

10.5 Multiple Mobile Targets Tracking

Multiple mobile targets tracking are often required for many applications. The key challenge in the multiple mobile targets tracking is how to enable enough beacons in the neighbourhood and if there are not enough beacons, there how to use some of the mobile target nodes whose locations have been determined as additional beacons.

Figure 10.9 shows that two mobile targets (m_1 and m_2) form a triangle by connecting them with one router and two end-devices in a ZigBee sensor field. In this case, tracking multiple mobile target nodes can be carried out by dividing the multiple targets scenario into a series of single mobile target tracking.

In the most cases the scenario might be much complex and mobile target nodes may not be able to find three beacons in their communication range as shown in Fig. 10.10. Mobile target node 1 (Class A) contains three beacons in its range, while mobile node 2 (Class B) has only two beacons, and mobile node 3 (Class C) has only one beacon. In this case, a mobile target node whose location has been previously determined can be used as an additional reference node. The 'first come first serve' principle is used to assign a mobile target node as a reference node. Class A offers the best localization accuracy, as the mobile target node is covered by 3 beacon nodes with known positions. The tracking accuracy of Class B will be lower than Class A, as the mobile target node is covered only by 2 beacon nodes with known position, and one mobile target node with previously determined position. Class C offers the worst tracking accuracy as the mobile target nodes is covered by only a single beacon nodes and the rest of the available reference nodes are the mobile target nodes with previously determined positions. The error will be accumulated in Classes B and C.

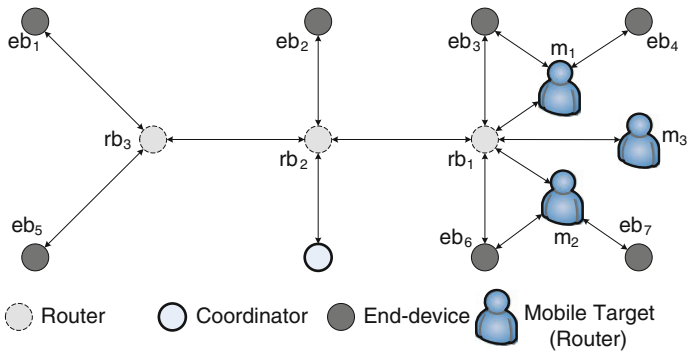


Fig. 10.9 Two mobile targets in a ZigBee sensor field

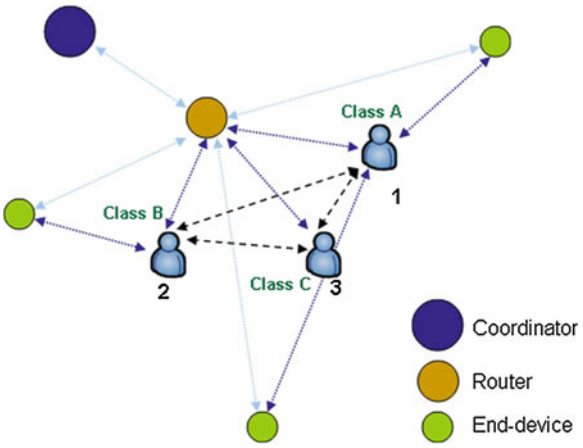


Fig. 10.10 Multiple mobile target tracking through a ZigBee network

10.6 Case Study—Underground Tunnel Mobile Target Tracking

Underground tunnels are quite challenging environments for WSN localization application due to the following characters: (1) the space shape is long and narrow, which means any WSN deployed there is of the line or chain type and has low density, and data transmission is energy expensive because of the multiple hops; (2) the air is wet and dirty due to water and dust, which significantly affects the valid wireless communication distance; (3) the surface is usually rough and the multi-path effect on radio propagation is severe. Figure 10.11 shows an underground coal mine.

A WSN based localization system deployed in an underground tunnel has a chain type topology because of the special geographic restriction. The structure of the system is shown in Fig. 10.12, consisting of a surveillance PC as a monitoring

Fig. 10.11 An underground coal mine

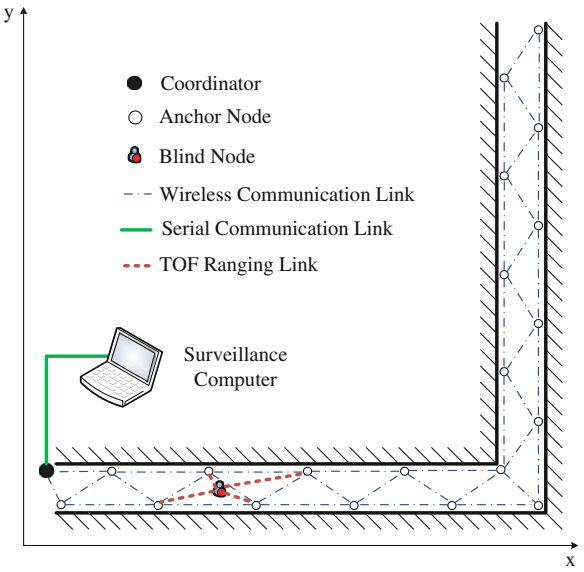


Fig. 10.12 Structure of a localization system (Qin et al. 2012)

station, a coordinator, reference nodes (called as anchor nodes here) and one or more mobile target nodes (called as blind nodes here).

The coordinator is responsible for establishing the network. It also acts as a gateway to the surveillance PC through a serial port. The surveillance PC is responsible for the configuration of the anchor nodes and localization data management. Anchor nodes are routers of the ZigBee network. They collect data from the tunnel environment and participate in localization. When receiving a localization request from a blind node in one hop range, the anchor nodes respond to it with their ID numbers and coordinates, and get the RF-TOF engine ready for range measurement. The blind node is a ZigBee router. It needs to communicate with multiple anchor nodes directly within its communication range. The blind node performs the localization algorithm.

To ensure the network communication has certain level of redundancy and the blind node can find enough anchor nodes in one hop range, the anchor nodes should be deployed along both sides of the tunnel. The distance between any two adjacent nodes on the same side should be kept equal, and shorter than the valid communication range between any two nodes. The anchor nodes on the opposite side should be placed alternately, in other words, one anchor node on one side is to be placed in the middle of two nodes on the opposite side, as shown in Fig. 10.12.

Three optimization approaches have been implemented in this application, linear least square estimation (LLSE) (Gezici et al. 2008), seven potential estimation (SPE) (Merhi et al. 2009), and particle swarm optimization estimation (PSOE) (Eberhart and Shi 2001). Details are omitted here. Figure 10.13 compares the average localization errors of these three algorithms. These three curves all have a similar trend, which means that large range errors could degrade the

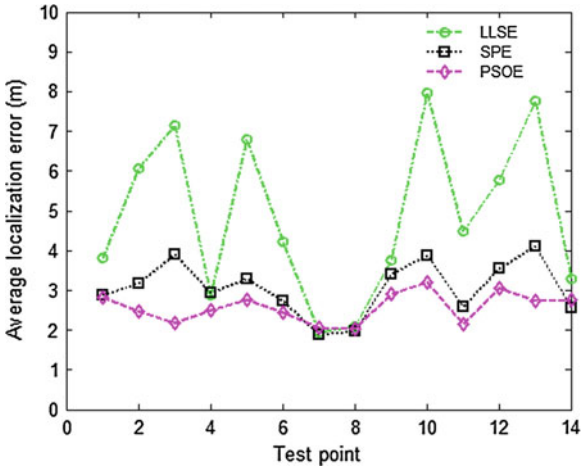


Fig. 10.13 Average localization error comparisons of three localization algorithms

performance of all these algorithms, but PSOE is the more robust with respect to the range error than the other two algorithms.

10.7 Summary

Even though there exist many more localization methods that are not discussed in this chapter, the core technologies and most basic principles of WSN based localization have been introduced. In summary, distance determination can be through radio signals such as RSSI or LQI, or travel time oriented such as ToA or TDoA. Localization methods are categorised as centralized or decentralized approaches by terms of their communication load and the place where computation occurs. They can be also grouped as range-based or range-free approaches in terms of whether or not absolute point-to-point distance is used in the estimate for calculating locations. Triangulation is a range-based approach, but fingerprint and centroid localization belong to the range-free category. One of the biggest challenges in localization is that the accuracy of tracking is often very poor. This chapter presents three ways of improving the accuracy, i.e. considering the uncertainty of the changed environment, eliminating the outliers of RSS, and finding the optimal solution for the localization. Multiple mobile targets tracking can be divided into a series of single target tracking if there exists three or more reference nodes in their communication range. Otherwise, the mobile targets, which have been previously localized, have to be used as an additional reference node even though the estimate error might be accumulated by involving them in the calculation of locations.

References

Alhmiedat, T., Yang, S.H.: A ZigBee Based Mobile Tracking System through Wireless Sensor Networks. Int. J. Adv. Mechatron. Syst. **1**(1), 63–70 (2008)

Alhmiedat, T., Yang, S.H.: Tracking mobile targets through wireless sensor networks, p. 47. Lap Lambert Academic Publishing, Saarbrücken (2011)

Blumenthal, J., Grossmann, R., Golatowski, F., and Timmermann, D.: Weighted centroid localization in Zigbee-Based sensor networks. In: IEEE International Symposium on Intelligent Signal Processing, pp. 1–6 (2007)

Cheng, B., Hudson, R., Lorenzelli, F., Vandenberghe L., Yao, K.: Distributed gauss-newton method for node localization in wireless sensor networks. In: IEEE Sixth Workshop on Signal Processing Advances in Wireless Communication, pp. 915–919 (2005)

Eberhart, R.C., Shi, Y.: Particle swarm optimization: developments, applications and resources. In: Proceedings of the IEEE Congress on Evolutionary Computation, Seoul, Korea, pp. 81–86 (2001)

Feng, W.J., Bi, X.W., Jiang, R.: A novella adaptive cooperative location algorithm for wireless sensor networks. Int. J. Autom. Comput. **9**(5), 539–544 (2012)

Gezici, S. Guvenc, I., Sahinoglu, Z.: On the performance of linear least-squares estimation in wireless positioning systems. In: IEEE International Conference on Communications, pp. 4203–4208 (2008)

Liu, Y., Yi X., He, Y.: A novel centroid localization for wireless sensor networks. Int. J. Distrib. Sens. Netw. Available online at http://1/dx.doi.org/10.1155/2012/829253 (2012)

Merhi, Z., Elgamel, M., Bayoumi, M.: A lightweight collaborative fault tolerant target localization system for wireless sensor networks. IEEE Trans. Mob. Comput. **8**, 1690–1703 (2009)

Qin Y., Wang F., Zhou, C., Yang, S.H.: A particle swarm optimization based distributed localization scheme in tunnel environment. Wireless Sensor Systems—IET Conference, June, London (2012)

Rappaport, T.S.: Wireless communications: principles and practice. Prentice-Hall Inc, New Jersey (1996)

Chapter 11
Hybrid RFID/WSNs for Logistics Management

Keywords Radio frequency identification · Hybrid sensor network · Network architecture · Logistics management

11.1 Introduction

The 4W questions Who, What, When and Where need to be answered in any logistics management system design. This chapter focuses on providing the answers to these 4W questions through building a hybrid RFID/sensor network.

With possible sensing capabilities, an RFID system will become responsive and conscious of the asset state and environmental conditions, from which we are able to ensure that our particular goods in the supply chain are not only in specific locations, but also in appropriate conditions (Pradhan 2005). Different types of RFID systems can be integrated under a hybrid RFID/sensor Network umbrella to satisfy more complex logistics applications. This chapter introduces the latest research effort to design, develop, implement and evaluate in the complex hybrid logistics applications an integrated RFID Sensor Network, in which various types of RFID systems and the wireless sensor networks can be integrated into a unified architecture.

11.2 RFID

Radio Frequency Identification (RFID) is one of the Auto-ID technologies. Auto-ID is short for automatic identification technology, which is a broad term for technologies that enable the machines to identify objects. Instead of having staff identify objects and manually enter their information into computer, the key aspect of Auto-ID technologies is their ability to capture data automatically. The main Auto-ID technologies include bar codes, smart cards, voice recognition, retinal or fingerprint scans, optical character recognition (OCR) and radio frequency identification (RFID).

S.-H. Yang, *Wireless Sensor Networks*, Signals and Communication Technology,
DOI: 10.1007/978-1-4471-5505-8_11,

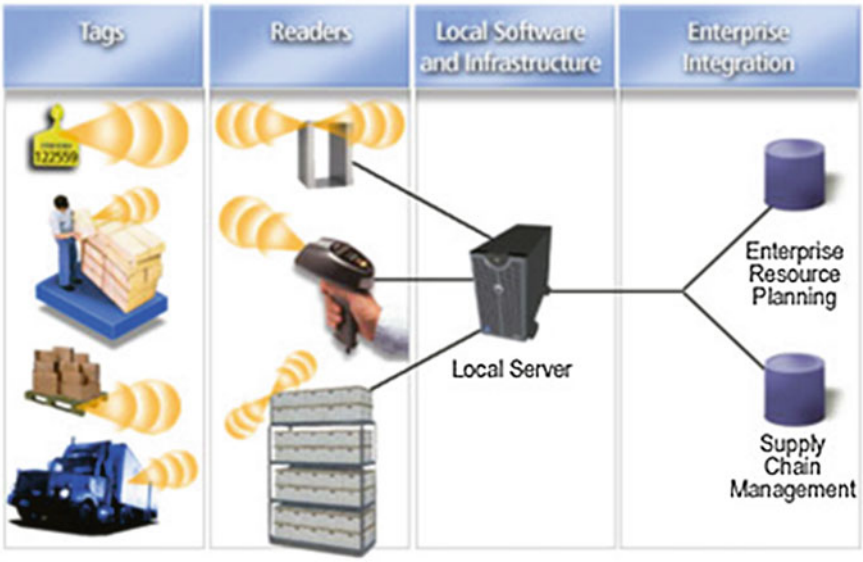


Fig. 11.1 Typical RFID system

RFID is a generic term for technologies that use radio waves to automatically identify people or objects. Compared to other Auto-ID technologies, the RFID system has its own features: instead of typing or scanning the identification code manually, the RFID systems provide a non-contact data transfer between the tag and the interrogator, without the need for an obstacle-free, line-of-sight reading; Tag information can be rewritable and the tag itself can be recycle and reused. Multiple tags can be read simultaneously by an RFID reader, which is known as the batch readability of tags, makes the identification work much more efficient; RFID tags are more reliable than a printed barcode which can easily be broken.

The basic components of a typical RFID system include: the transponder or the tag, which is a microchip in which a unique serial code is stored and transmitted when necessary via an attached antenna; the RFID reader, which is used to receive and identify the information sent by tags; the local server, to where the readers forward the information, and a management system which uses the data collected from the tags. Figure 11.1 illustrates a typical RFID system. This chapter mainly focuses at the tag and reader levels.

11.2.1 RFID Tag

The purpose of a tag is to physically store and attach data about an item to that item. The tags also have the ability to communicate that data, so that it can be read out. There are generally three types of RFID tags depending on the power source

Table 11.1 Comparison of different RFID tags (Yang and Yang 2007)

Features	Passive tag	Semi-active tag	Active tag
Own power for data transmission	No	No	Yes
Own power for chip	No	Yes	Yes
Communication with readers	Backscatter	Backscatter	RF transmit
Read range	Short	Medium	Long
Tag cost	Low	Medium	High

used, which are active, passive and semi-passive/semi-active tags. A brief comparison between them is given in Table 11.1.

A**ctive tags** have on-board transmitters and use batteries as their power resource; they transmit their ID codes through transmitters, and can consequently have a wide reading range. Active tags usually work on 455 MHz, 2.45 GHz, or 5.8 GHz, and they typically have a read range of between 20 and 100 m. Because of their ability to track objects over long distance they are usually used to track large assets, such as containers, vehicles and aircrafts.

There are two types of active RFID tags, which are the transponder and the beacon. Active transponders do not send information spontaneously; they are woken up from sleep mode only when a signal from a reader device is received and then the tag ID is transmitted to the reader. These tags are usually used in checkpoints control systems. The aim of having a tag broadcast its information only when it is within the range of a reader is to conserve battery life. Beacons are used in most real-time locating systems (RTLS), in which the precise location of an asset must be tracked. Both active transponders and beacons can have a read range of up to 100 m, and reading the tags is reliable as they use on-board transmitters to send a signal. The cost of an active tag ranges from £5 to £30.

Passive tags, or as they are also called backscatters, do not have their own power resource and on-board RF transmitter, they use inductive/propagation coupling to connect with the reader's antenna, which means that the passive tags just simply reflect back the signal emitted by the reader. As a result, the passive tags are simple and small and thus are cheaper; the price can be 10–20 pence. They are also more reliable under harsh environment conditions. However, compared to active tags, they can achieve only a much shorter reading range of 0.1–9 m. Passive tags cannot transmit information without the presence of a reader, and their communication follows the fixed pattern, where firstly the reader inquiries, and then the tags respond the inquiries.

Semi-active tags, which are also called semi-passive tags or battery-assisted tags, are also currently available in the market for specific applications. These semi-active tags contain batteries that are only used to support the embedded memories and sensors. The communication between a reader and tags, follows the same method as with the passive tags, which means the tags obtain energy from the reader transmission and reflect a signal back to it. Like the passive tags, the communication always starts as a result of the reader's enquiry, and the tags then respond. They can be read at even longer distance than passive tags, from up to 30 m, and in addition,

their performance in the presence of metals and liquids is much better than passive ones. Semi-active tags do not need time to gather energy and activate the tag chip so faster reading speed is another advantage of semi-active tags.

11.2.2 Reader

No matter which type of tags is used, they are just storing the data of the item on which they are attached. In any real applications data needs to be read out and transferred to a server or a network on demand. A reader, also called an interrogator, is the device that communicates with the tags, performs low-level events, such as reading and writing tag, and then sends the results of those events to the server. A reader can be a stationary or a handheld device; the typical components of a RFID reader include antenna, RF transceiver, microcontroller, communication interface and power supply.

11.3 Hybrid RFID/Sensor Network

RFID system can be integrated with WSNs at the hardware or software level. Hardware level integration combines RFID transponders with sensor nodes by embedding them onto the same printed circuit board and combining sensors with RFID reader devices. Therefore, the sensors work with the RFID reader device to enable both RFID and WSN functions. Software level integration allows RFID networks and WSNs to cooperate and work together at the network level. This section only considers software level integration.

11.3.1 Reader as a Sensor

In order to combine RFID systems with WSNs, the conception of a ‘sensor’ need to be extended. Generally a sensor is a device that responds to a stimulus of a particular type of environmental condition and pursues a specific physical measurement. In this RFID/sensor network, the concept of the sensor is extended to involve the RFID reader device as a sensor. What a reader device ‘senses’ is the appearance, the approaching or the passing of a RFID transponder/tag within its reading range. In this case, the RFID readers and the sensor nodes (units) of the sensor networks are considered to be in the same layer in the system architecture. The sensor network gateway device, such as a sensor coordinator, will also act as the gateway device between the RFID readers and the central server/network. All information generated by the readers will be sent to the central server via the sensor network gateway device. Figure 11.2 shows this extension of the concept.

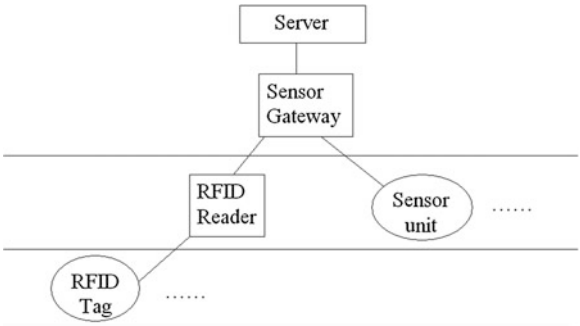


Fig. 11.2 'Reader as a sensor' architecture

Although various interfaces and protocols such as RS-232, RJ-45 or WiFi are applicable to bridging the readers and the sensor network gateway devices, the Wireless Sensor Networks protocols (IEEE 802.15.4 or Zigbee) are recommended to make the reader devices the real WSN nodes. The concept of 'reader as a sensor' has been used in several publications (Englund and Wallin 2004; Mason et al. 2006).

An example of an integrated application of this architecture is shown in Fig. 11.3. A number of RFID readers are implemented in a warehouse to perform object identification, access control and real time location tracking tasks; each of them has been given a RF transceiver and a RF antenna to realize the wireless communication ability. Wireless sensor nodes are also implemented in the warehouse to survey environmental conditions and accidents. The wireless sensor network protocol is applied to the wireless ad-hoc network constructed by the RFID readers and the sensor nodes. It is most likely with this architecture that a RFID system has been integrated into the upper layer of a wireless sensor network structure.

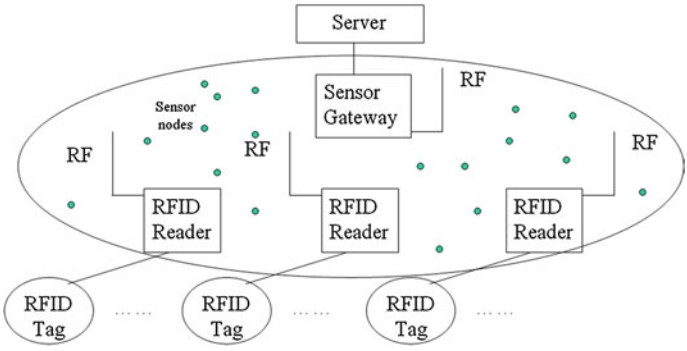


Fig. 11.3 Example application for 'Reader as a sensor' architecture

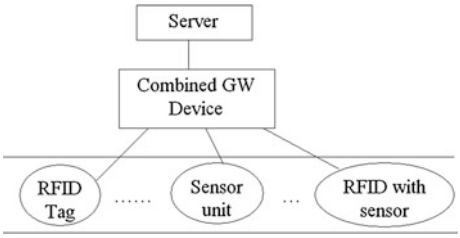


Fig. 11.4 'Tag as a sensor' architecture

11.3.2 Tag as a Sensor

In the 'Tag as a sensor' architecture shown in Fig. 11.4 the 'sensor' conception is extended to treat the RFID transponder/tag devices as a sensor. What a transponder device 'senses' is the unique identification code stored in the tag's memory. When a tagged asset or person moves within the reading range of the reader device, the tag 'senses' the identification code of the asset or the person and transmits the identification code to the reader device. In this case the RFID tags and the sensor units are considered to be in one layer of the architecture. The reader devices and sensor network coordinators are in another layer, in which a combined gateway (GW) device communicates with the sensor units, the RFID tags and the RFID tags with sensor. An example of this type of combined gateway reader device has been given in the 'RFID reader' section. If we are using only RFID sensor nodes in a network, as they work in a very similar to a typical sensor network node, the combined gateway devices could be just the sensor network gateway with some slight modification.

11.4 Generic Hybrid RFID/Sensor Network Architecture

Passive tags are the most practical solutions in any massive implementation for cheap and non-recycled goods, and consequently the passive systems are most suitable for use in the 'reader as a sensor' architecture; on the other hand active RFID tags are easily incorporated into the 'tag as a sensor' architecture. Individual sensor nodes without any identification can work in both architectures at an appropriate layer. The 'tag as a sensor' architecture does not require any additional reader devices and thus can be cost-effective for small and medium applications. For larger applications the 'Reader as a sensor' architecture may be the better choice as the tag cost becomes more critical.

In general, the above two architectures can be presented as a united hybrid RFID sensor network system as shown in Fig. 11.5. There are three layers in this architecture. From the RFID aspect, the server layer is at the top, the reader layer is in the middle and the tag layer is at the bottom. From the WSN aspect, the network coordinator is the top layer, the network routers are in the middle and the

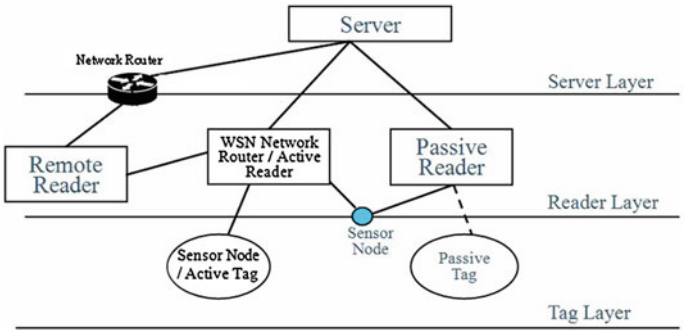


Fig. 11.5 Integrated hybrid sensor network architecture

individual sensor nodes are at the bottom. As described in Chap. 2 individual sensor nodes are not expected to communicate with each other, but can talk to their parent routers. Here we do not expect RFID tags and sensor nodes to communicate with each other in the tag layer, but they can talk to their readers or routers in the reader layer. All the readers and network routers are expected to communicate with each other if they are in their communication range, i.e. the reader layer forms a mesh network. The server sits at the top and manages the RFID readers and network routers.

In this integrated architecture all communications inside the network are expected to be supported by Wireless Sensor Network protocol such as Zigbee, except for communication between passive RFID tags and their readers. This integrated architecture for hybrid RFID sensor network is actually a combination of the 'reader as a sensor' and 'tag as a sensor' architectures. Thus it can benefit from the features of both architectures. One of the key technical problems is to make RFID readers wireless enabled and allow them to communicate with the

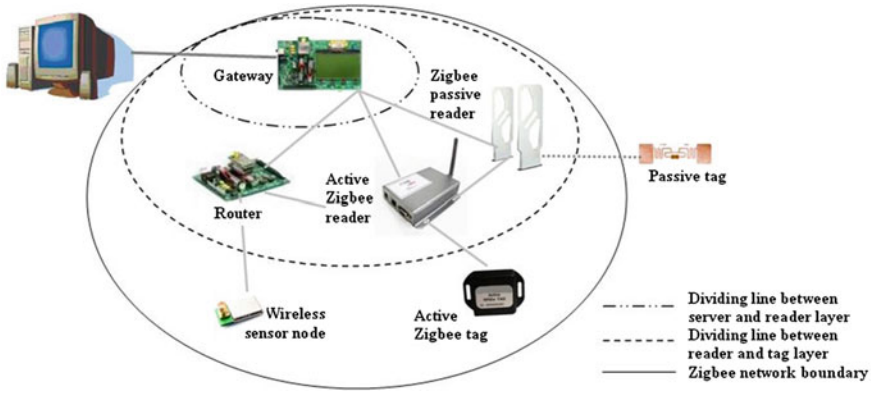


Fig. 11.6 Zigbee-based RFID/sensor network architecture (Yang et al. 2011)

routers of a WSN and then pass the information collected from RFID tags to the server through these routers. The simple way to solve this technical problem is to integrate the RFID reader with the wireless router at the hardware level. Figure 11.6 shows that an active RFID reader and a passive RFID reader have been made as an 'active Zigbee reader' and a 'Zigbee passive reader'. These two Zigbee readers can communicate with ordinary Zigbee routers and form the backbone of the Zigbee wireless network. Wireless sensor nodes only communicate with their parent routers, and active tags and passive tag communicate with their active Zigbee readers and Zigbee passive readers respectively.

11.5 Possible Use in Humanitarian Logistics Management

Humanitarian aid is defined as 'material or logistical assistance provided for humanitarian purposes', typically in response to a humanitarian crises. The primary objective of humanitarian aid is to save lives, alleviate suffering and maintain human dignity. Humanitarian logistics is a broad term that covers operations concerning supply chain strategies, processes, and technologies that will help make humanitarian aid more effective.

The transport and delivery of emergency aid goods and materials is the main task of the humanitarian supply chain. Consequently, the initial transportation of such commodities is the very first thing on the scene that needs to be managed. To correctly and efficiently monitor the flow of commodities, information on the goods inside the logistics centre, such as type, amount, position and state, should be recorded and updated in real-time. Food and medicine are key goods in the humanitarian supply chain; these types of goods require specific environmental conditions during storage and transport, which means information on environment monitoring is also necessary. Other freights include large and valuable specialised rescue equipment (Özdamar et al. 2004) as well as forklift trucks, plants and vehicles, which should also be tracked for management and safety considerations. As disaster management involves working inside a disaster affected area, location tracking of both equipment and people is equally important in humanitarian aid actions in an unknown environment.

In summary, distribution centres in the humanitarian supply chain have the following requirements for their information support systems:

- Tag and identify various types of freights, tracking them in the logistics process;
- Monitor specific storage conditions of some goods, thereby maintaining their quality;
- Tag and identify equipment such as specialised rescue equipment, vehicles, plants and medical equipment, tracking them for both logistics and safety purposes;
- Tag and identify staff and officers working and living in the centre, tracking them for both management and safety purposes;

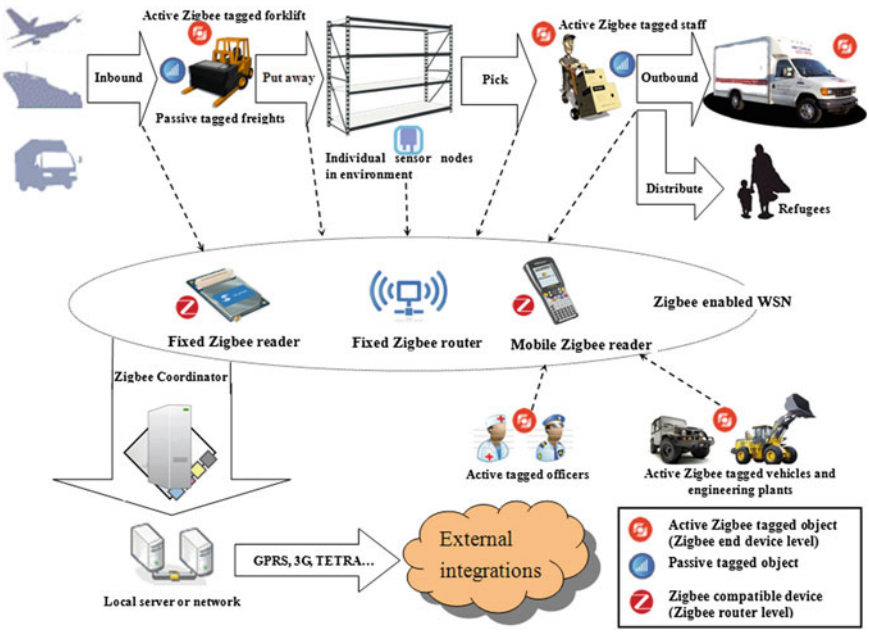


Fig. 11.7 Zigbee based RFID sensor network in humanitarian logistics centre (Yang et al. 2011)

- Have a simple but reliable network architecture and devices that do not depend on any local facilities which cannot be assured in a disaster area;
- Have an easy and fast implementation process to perform fast responses to emergency actions.

Figure 11.7 describes how the proposed integrated hybrid RFID sensor network architecture can be implemented in a humanitarian distribution centre. Because of the poor performance of standard passive RFID tags when they work with materials containing metal and water, active tags are recommended for tracking vehicles, engineering plants, large special rescue equipment and people in the scenario. Zigbee end devices are modified to act as active RFID tags; they can be manufactured in various package shapes for different purposes. For tracking the staff and officers in the centre the tag can be made as wrist strip, badge or be integrated in other personal devices such as watches and mobile phones. The package of the active tags for vehicles and equipment could come with belt or screw holes to help fit them to the vehicle chassis or equipment frame.

Those active Zigbee tags communicate with the active Zigbee readers modified from typical Zigbee routers. These reader/router devices should be implemented over the entire scenario to ensure coverage throughout the centre. The density of the readers depends on the security level or the accuracy of tracking required. Generally speaking, this can be divided into three levels: site level, sector level and room level. A site level, accuracy means the information required for the tracked

object is just whether it is on-site or not; this requires only a basic number of readers to ensure network coverage. This accuracy level can be easily satisfied as long as the tag can communicate with at least one reader/router device when it is in the centre. If a sector level accuracy is chosen then each tag should be able to find multiple reader/router devices in the centre. By indicating the reader, which has the best receiving signal strength indicator (RSSI) with the tag, the position of the tracked object can be limited to within a rough area close to a specific reader. In certain circumstances when room level or even metre level accuracy is necessary, the tag should be able to get the RSSI or time difference of arrival (TDOA) indicator from no less than three reader/router devices whenever it is in the distribution centre, thereby requiring the highest reader density.

The freights going through the centre are expected to be tracked by typical passive RFID tags. Traditional passive RFID readers are integrated with the Zigbee routers/readers to be able to read both traditional passive tags and active Zigbee tags. These hybrid Zigbee readers should be installed at all access points where logistics actions are carried out.

To increase the flexibility of the system, both the Zigbee active reader and the passive hybrid reader can also be designed as handheld devices with rechargeable batteries for temporary operations where fixed readers are not useable.

Dedicated wireless sensor nodes can also be implemented in the scenario where certain environmental conditions need to be monitored. For example food, water and medicines should be stored under certain temperature conditions; while humidity in fruit storage may be crucial (Jedermann et al. 2006). Some dedicated Zigbee routers may also be implemented to help establish and maintain a Zigbee WSN backbone with passive and active Zigbee readers. The local server or network can connect to the Zigbee coordinator or any programmed sink node in the WSN to retrieve information, which could be processed locally for decision support or could be sent over to a remote command centre via other WAN network such as GPRS, 3G or TETRA etc. All the nodes/devices can be designed to be battery assisted, which means they will use an external power supply in general

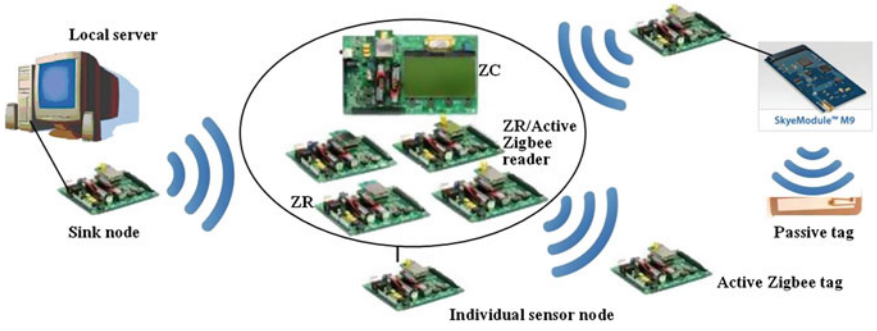


Fig. 11.8 Zigbee based RFID sensor network demonstration system (Yang et al. 2011)

situations, but can switch to battery during possible electricity supply intervals caused by either man-made accidents or the after effects of the disaster.

The structure of the hardware demonstration system is presented in Fig. 11.8. The Zigbee network is constructed using a Jennic JN5139 development kit (*Jennic.com*). A Zigbee coordinator (ZC) establishes the Zigbee network first; several Zigbee routers (ZR) could then join the network. The active Zigbee tags and readers, passive Zigbee readers and individual sensor nodes could then join the network on a plug-and-play base.

11.6 Summary

Many dedicated systems currently exist for the accomplishment of a single task, for example using passive RFID for identifying freights, WSNs for monitoring the environment and active RFID for tracking people and equipment. But none of the systems can simultaneously handle all of these required tasks. Implementation of several independent systems and integrating them in a single software/management coordinated system may cause various problems in the applications. The hybrid RFIS/sensor network provides a system combining WSNs, passive and active RFID together in both hardware and software layers. It has an unified, fully integrated and cordless system architecture. This chapter introduces ways and possible architectures for this type of hybrid networks. Two concepts 'Reader as a sensor' and 'Tag as a sensor' have been presented as the foundation of a hybrid RFID/sensor network. The key is to integrate the RFID readers with wireless routers and forms a Reader/router network. RFID tags and wireless end-devices can communicate with their readers or routers. No communication between tags and wireless end-devices is required. The possible application in humanitarian logistics management is used as a case study in the chapter.

References

Englund, C., Wallin, H.: RFID in Wireless Sensor Network. Chalmers University of Technology Report, GÄoteborg, Sweden, EX034/2004 (2004)

Jedermann, R., Behrens, C., Westphal, D., Lang, W.: Applying autonomous sensor systems in logistics-combining sensor networks, RFIDs and software agents. Sens. Actuators, A **132**(1), 370–375 (2006)

Mason, A., Show, A., Welsby, T.: RFID and wireless sensor network integration for intelligent asset tracking systems. 2nd GERI Annual Research Symposium GARS-2006, Liverpool, UK (2006)

Özdamar, L., Ekinci, E., Küçükyazici, B.: Emergency logistics planning in natural disasters. Ann. Oper. Res. **129**(1–4), 217–245 (2004)

Pradhan, S.: RFID and Sensing in the Supply Chain: Challenges and Opportunities. HP Laboratories Tech Report HPL-2005–16 (2005)

Yang, H., Yang, L., Yang, S.H.: Hybrid Zigbee RFID sensor network for humanitarian logistics centre management. J. Netw. Comput. Appl. **34**(3), 938–948 (2011)
Yang, H., Yang, S.H.: RFID sensor network–network architecture to integrate RFID, sensor and WSN. Measur. Control **40**, 56–59 (2007)

Chapter 12
Internet of Things

Keywords Internet of things · Service-oriented architecture (SOA) · Emergency response

12.1 Introduction

The concept of the Internet of Things (IoT) is to make every single 'network enabled' object in the world network connected, and represents a vision in which the Internet extends into the real world embracing everyday objects (Mattern and Floerkemeier 2010). The term 'Internet of Things' was popularized by the work of the Auto-ID Center at the Massachusetts Institute of Technology (MIT), which in 1999 started to design and propagate an across-company radio frequency identification (RFID) infrastructure (Sarma et al. 2000). One of the definitions of the IoT described it as 'a self-configured dynamic global network infrastructure with standards and interoperable communication protocols where physical and virtual 'things' have identities, physical attributes, and virtual personalities, and are seamlessly integrated into the information infrastructure' (European Commission 2009). Figure 12.1 shows these integrations between physical, virtual, and digital worlds.

The concept of 'things' in the network infrastructure refers to any real or virtual participating actors such as real world objects, human beings, virtual data and intelligent software agents. The purpose of the IoT is to create an environment in which the basic information from any one of the networked autonomous actors can be efficiently shared with others in real-time. There are many definitions of the 'Internet of Things' in the research and relevant industrial communities. The definitions may arise from the word 'Internet' and lead to an 'Internet oriented' vision, or 'things' and lead to a 'things oriented' vision. Putting the world 'Internet' and 'Things' together semantically means a world-wide network of interconnected objects uniquely addressable, based on standard communication protocols. Atzori et al. (2010) presented this third vision of IoT as 'semantic oriented' and the IoT

S.-H. Yang, *Wireless Sensor Networks*, Signals and Communication Technology,
DOI: 10.1007/978-1-4471-5505-8_12,

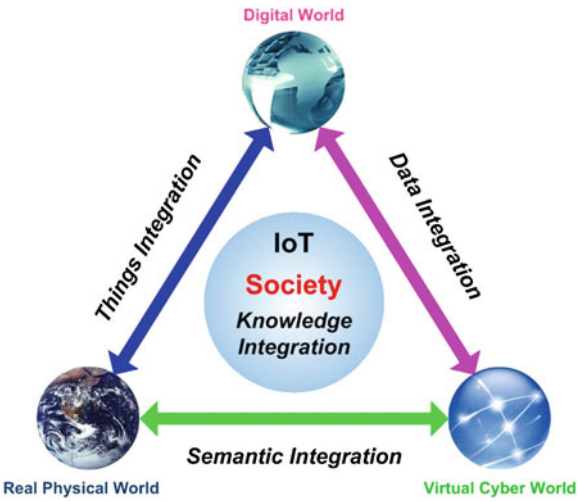


Fig. 12.1 Interaction among the real physical, the digital, virtual worlds and society (European Commission 2009)

paradigm was modified to embrace the convergence of these three visions. The research roadmap from the European Commission (2009) deemed the IoT as an integrated part of any future Internet. Some researchers tended to consider the IoT as a separate part of the Internet. Gershenfeld et al. (2004) described the IoT as an extension of the Internet to reach out into the physical world of things and places that only can support low-end computers, whilst (Fleisch 2010) argued that the IOT is not on the same level as the Internet, but it is in fact an application of the Internet just like many existing Internet-enabled services.

Since the concept of IoT was introduced in 2005, we have seen the deployment of smart 'network enabled' objects with communication, sensory and action capabilities for numerous applications such as in the areas of healthcare (Niyato et al. 2009), smart buildings (Gill et al. 2012), social networks (Welbourne et al. 2009), environment monitoring (Llic et al. 2009), transportation and logistics (Yang et al. 2011), etc. All applications of the IoT rely on the data collected from distributed smart 'network enabled' objects, i.e. WSNs and the IoT information infrastructure for data transmission.

12.2 Challenges and Features of the IoT

The concept of the IoT is mainly driven by continuous progress in microelectronics and networking technologies in pervasive and ubiquitous computing. It is a multi-disciplinary study that involves research in the fields of hardware, near-field communication, networking, data fusion and software engineering etc. Scientific and technical challenges require different competencies (Association Instituts Carnot 2011):

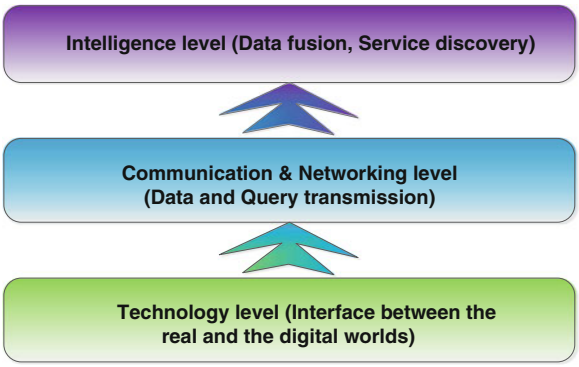


Fig. 12.2 Three main challenging domains of the IoT

- Technology level—Challenges linked to the integration of smart 'network enabled' objects under strong energy and environment constraints;
- Communication and networking level—Challenges linked to massive secure and dynamic and flexible networking and the ubiquitous service provision;
- Intelligence level—Challenges linked to the data fusion and service discovery where data collected by individual smart 'network enabled' objects such as RFID and wireless sensors are queried by distributed users.

A hierarchy of different levels of technologies involved in the IoT can be identified from Fig. 12.2. The key functionalities in the technology level are to enable interaction between the "things" which are considered as identification, sensing, storage, actuating, and other interfacing activities. The interface between the real and the digital worlds requires the capacity for the digital world to sense the real world and to act accordingly. Technologies such as RFID, sensors, wireless sensor networks (WSNs) are feeding some specific functionality to support the IoT. However, simply equipping objects with microchips and retrieving information at a local level is insufficient in itself. These smart 'network enabled' objects of the IoT are beyond current devices based on 'simple' sensors and RFID or any combination of these two. They are, in particular, based on cheap and small wireless devices with sensing, acting, communication, and advanced signal and information processing capabilities. IP enabled technologies such as 6LOWPAN (IETF 2007) which make it possible to build low cost and reliable solutions and services to enable the interconnection of various "things" in the IoT.

The fundamental features of the IoT technology have been summarized as follows (Yang et al. 2013):

- The IoT is a global and real-time solution;
- It is mainly wireless oriented and able to provide comprehensive data about its surroundings in both indoor and outdoor environments; and
- It has the ability to remotely monitor the environment and trace or track objects.

The first fundamental feature of the IoT technology is that it is a global and real-time solution. Firstly, the IoT technology is Internet-based or based on other

wide-area network-based. Thus, the scope of the IoT, has no physical boundary. Any object linked with the network can be incorporated into the IoT. Secondly, data communication is real-time or almost real-time over the IoT. In this way it differs from traditional databases or web systems.

The second feature of the IoT is that it is wireless and possesses the ability to provide comprehensive data about its surroundings. RFID sensor networks in the IoT integrate RFID networks and wireless sensor networks into a unified information infrastructure. No line-of-sight is required in RFID sensor networks for their sensing tasks. This feature significantly increases the richness of information.

The third feature of the IoT is its ability to monitor the environment and trace and track objects. By combining the use of RFID sensor networks with other technologies such as Global Positioning System (GPS) or infrared sensor detection, RFID sensor networks provide the ability for the wireless, real-time monitoring and tracking of any tagged object in an indoor or outdoor environment in order to provide complete visibility of the resources. Such visibility enables instant response to any exceptional event, distributed information sharing among multiple organizations and multiple users, and resource distribution.

12.3 Connecting WSNs with the Internet

No matter which definition of the IoT we take, the core requirement of the IoT is to provide direct or indirect connectivity with any network-enabled smart 'things' at any time and from anyplace. Comparing with the definition of the Internet 'having connectivity with any computer at any time and from anywhere', the only difference between the IoT and the Internet is the terminal devices of the IoT are smart things rather than computers. WSNs are a network of smart things. Connecting WSNs with the Internet will form a IoT information infrastructure.

There are many ways of connecting WSNs to the Internet (Kosanovic and Stojcev 2011). Xu (2013) categorised them into three groups in terms of their communication architectures, as shown in Fig. 12.3. The first approach is front-end proxy solution, where the connection is performed by middleware proxy and there is no direct connection between WSNs and the Internet. The second approach is a gateway solution where a gateway is located between WSNs and the Internet. The final approach is TCP/IP overlay solution, which is performed by an overlay network constructed on either WSNs or the Internet. Details are given below.

12.3.1 Front-end Proxy Solution

As shown in Fig. 12.4, the communication between TCP/IP users and sensor nodes is done through the proxy computer, rather than directly. The communication protocol used in the sensor network may be chosen freely. The proxy pro-actively

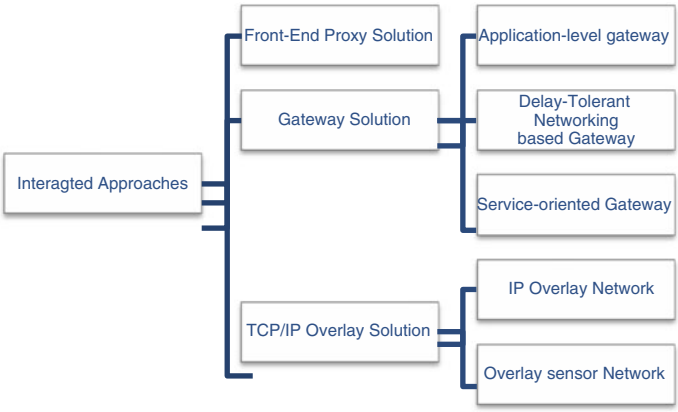


Fig. 12.3 Categories of the integration of WSNs with the Internet

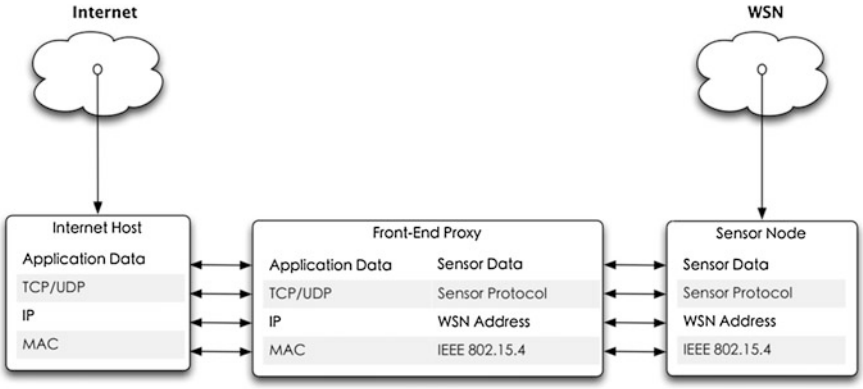


Fig. 12.4 Front-end proxy solution

collects data from WSNs and stores this information in its database. The users from the TCP/IP networks can query for specific data in a variety of ways, such as SQL queries or Web based interfaces. The drawback is that all communications to and from WSNs are broken, if the proxy stops working, and the proxy implementation usually depends on the specific task or a particular set of protocols. This means that for each application a different proxy is needed.

12.3.2 Gateway Solution

One of the essential devices that provide a connection between WSNs and TCP/IP network is a gateway. It performs several tasks such as protocol conversion,

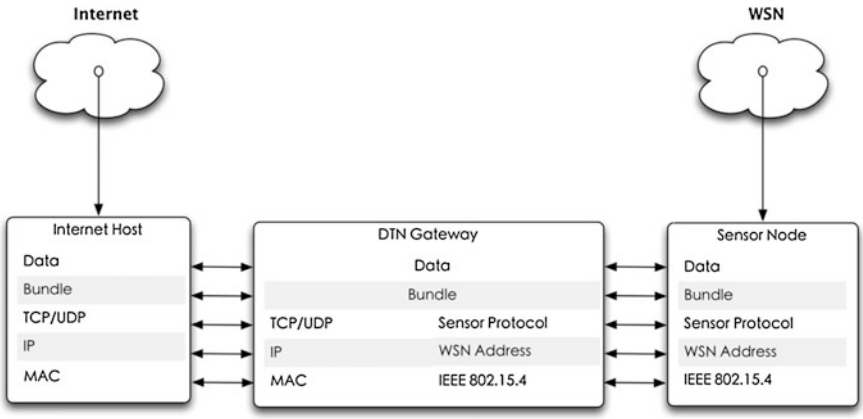


Fig. 12.5 Gateway solution

message relay, etc. As a result, the sensor nodes and the Internet hosts can directly exchange information. All solutions, that use the gateway as an interconnecting device, can be grouped into the following two categories: Application gateway and Delay Tolerant Network (DTN). An application gateway is a simple gateway-based approach, which works in the application layer. There is a translation table that maps WSNs addresses to IP addresses. The DTN is a similar solution. The main difference from the Application gateway is that it implements one new layer, in both the TCP/IP and WSN networks, referred to as the Bundle Layer. The main function of the bundle layer is to store and forward packets between the two networks, as shown in Fig. 12.5.

12.3.3 TCP/IP Overlay Solution

TCP/IP overlay sensor networks is to implement TCP/IP protocol stack above a microcomputer system with very limited resources, as shown in Fig. 12.6. Numerous problems accompany the implementation of TCP/IP in WSNs. For example, how an IP address is assigned to the sensor node and how to mix the address-based and data-based routing efficiently according to network traffic? 6LowPAN is a typical TCP/IP overlay solution. It defines the method of transmitting a IPv6 packet over IEEE 802.15.4 MAC layer. Internet users can access individual sensor nodes directly by using the IPv6 address.

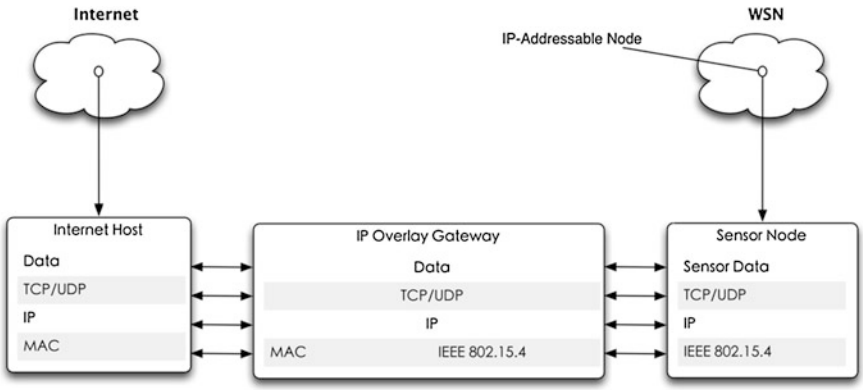


Fig. 12.6 TCP/IP overlay solution

12.4 IoT Service-Oriented Architecture

Service-oriented architecture (SoA) defines an information architecture in terms of three protagonists in the architecture, namely the service producer, service consumer, and service registry. The entities participating in a SoA are limited to a single role only at the time of a single interaction. Service producers publish their services on one or more registries. Service consumers use a search facility on the registry to find a desired service (Yang 2011).

SoA for the IoT is designed to globally support multiple uses, multiple applications and multiple data sources. It provides one unified platform to collect, share, process and query data from distributed RFID sensor networks and allow these RFID sensor networks to join and leave the IoT without affecting the rest of systems.

An IoT architecture implementing this approach is shown here as shown as Fig. 12.7. The main components are an RFID sensor networks, sensor service publishers (SSP), distributed local historical database systems, and a domain sensor name server (DSNS). The RFID sensor networks send their data via the SSP either to end users in response to their queries or to the local databases for backup if any change occurs. The DSNS works in a similar way to a Domain Name System (DNS) in the Internet and directs any query received to a corresponding SSP where the response is first formed and then sent to the end-users over the Internet. The dataflow between the end-users, the SSPs, the DSNS, and the local database systems is illustrated in Fig. 12.8. All communications in this Internet-based architecture are supported by TCP/IP, except for the communication within the RFID sensor networks where IEEE 802.15.4 and ZigBee are employed to provide data collection with low cost, low data rate and short range communications.

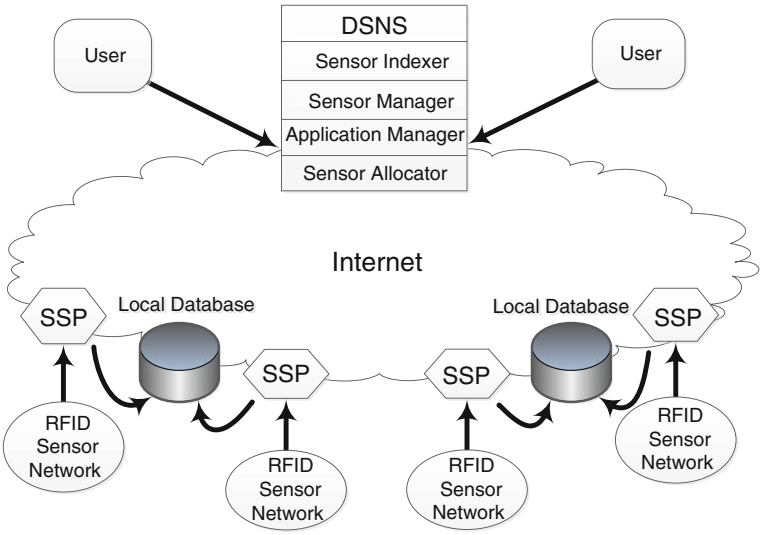


Fig. 12.7 IoT service-oriented architecture

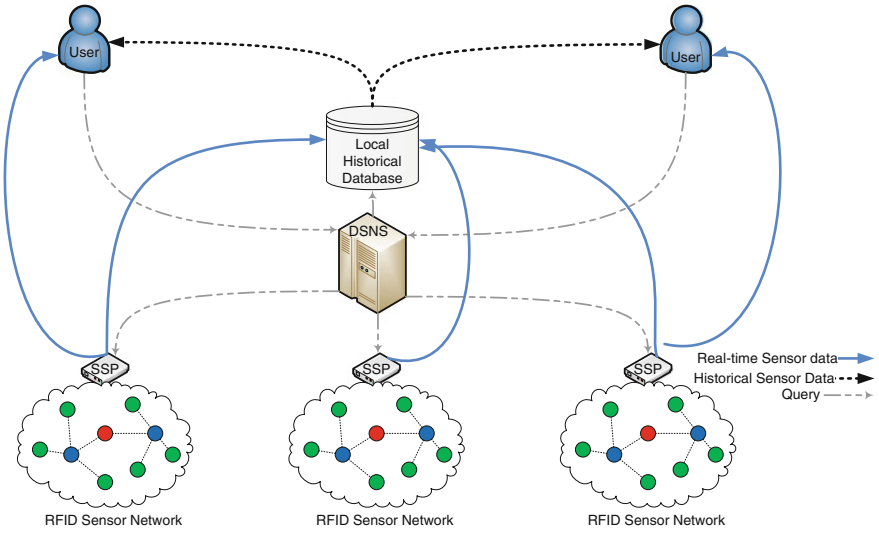


Fig. 12.8 Dataflow of the IoT architecture

12.4.1 Sensor Service Publisher

The gateway approach is the traditional approach widely used with the non-IP wireless embedded networks and other vendor-specific solutions. In the gateway approach, a gateway is introduced at the edge of the network that deals with the IP services. The proposed IoT architecture shown in Figs. 12.7 and 12.8 adopts the gateway approach. The SSP is designed as an extended gateway for reporting its own RFID sensor network's characteristics, publishing the availability of collected data and uploading these data to the Internet. The SSP is also a type of service provider. When it joins a RFID sensor network the SSP maintains a connection between the RFID sensor network and the Internet, and responds to queries from the DSNS.

12.4.2 Local Historical Database

In the traditional approaches sensory data and other information are transferred to a server and processed centrally. However, central processing has many disadvantages. For example, central approaches suffer from inadequate communication and processing capacity, vulnerability to single-point failures, and delay in real-time data fusion. In order to avoid these problems of centralized approaches, a distributed local database is introduced for each RFID sensor network deployed. Using a warehouse as an example, a local database is installed for storing any changing data collected by the RFID sensor network deployed in and around the warehouse. The automatic data collection is even-driven and trigged by any change in the warehouse. Data query is made by the end-users over the Internet and directed to the corresponding local database by the DSNS. The response to the data query is managed by and sent from the local database systems.

12.4.3 Domain Sensor Name Server

The Domain Sensor Name Server (DSNS) is a sensory data central indexing system and works in a similar way to the Domain Name Server in the Internet. The DSNS forwards the queries received from the end-users to a corresponding SSP or a corresponding local sensor database where the required data is collected or saved. A response to the query is formed and is then sent to the end-user. The DSNS features a sensor manager, an application manager, a sensor indexer and a sensor allocator. Figure 12.9 illustrates the interaction and workflow between these components, where online exception data and offline exception data go to the end-users and the local database systems respectively, real-time data order goes to

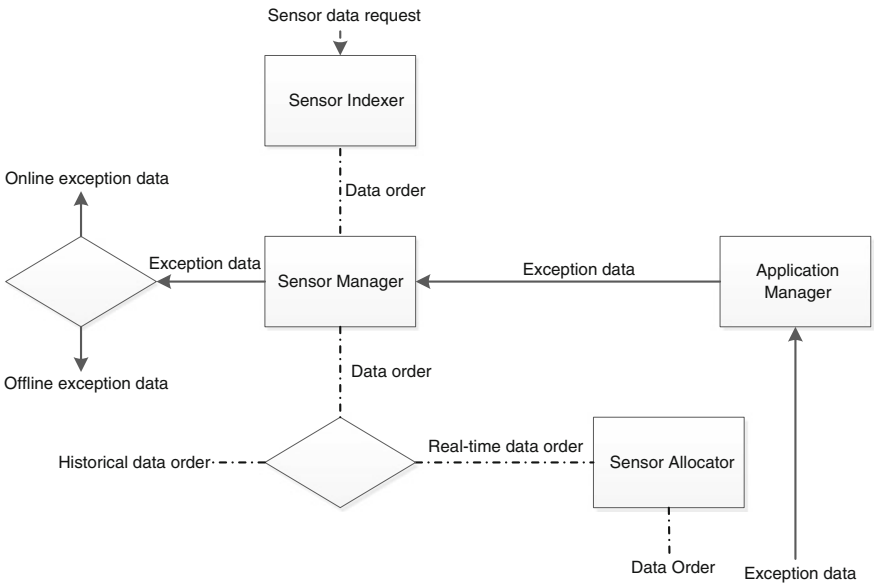


Fig. 12.9 Interaction between the components of the DSNS

the identified RFID sensor networks and historical data order goes to the identified local database systems.

- The Sensor Manager is for sensor networks registration. It provides a mechanism for registering new RFID sensor networks by specifying their types, descriptions and names. In addition, it implements a maintenance function for updating the characteristics of any previously registered RFID sensor networks.
- The Application Manager is designed to register the application information, such as the IP address, application functionality and the types of sensory data required. The reason for storing this information is to notify these applications whenever those RFID sensor networks capture any predefined exception data for them.
- The Sensor Indexer provides a search engine, which implements the search services in terms of the sensor type or sensor characteristic for registered applications. The search engine searches the RFID sensor networks in respect to the sensor type, location, and characteristic. If there are one or more RFID sensor networks that can provide the specified type of sensory data, a positive response will be fed to the Sensor Allocator. Subsequently, direct data transmission link between the identified RFID sensor network and the data requester will be arranged.
- The Sensor Allocator is the most important and heavily loaded part of the DSNS. It allocates a connection link between the data requester i.e. an

application and the data provider that is a SSP. There are two types of queries that may trigger the Sensor Allocator: on demand queries and general queries.

- *On demand queries*: These occur once a RFID sensor network has detected any predefined exception data and the data has been transmitted to the SSP. The SSP will publish the property of the data and the sensors, which collected the data to the DSNS. The Application Manager in the DSNS checks whether there is any registered application interested in the detected exception data. If there is, the Application Manager returns the identified application connection information to the Sensor Allocator. The Sensor Allocator sends a connection request with a listening port number to the application and waits for the connection confirmation from the application. If there is no application interested in the detected exception data or there is no any connection confirmation received, the local database system saves the detected sensory data for future use.
- *General queries*: General query occurs when an application sends a search request to the Sensor Manager of the DSNS and asks for a particular type of data. The Sensor Indexer will search for any SSP registered in the DSNS that can provide the data required by the application. The Sensor Allocator will return the details of the identified SSP including the IP address and the port number to the application where the general query was made. The SSP activates the listening port for receiving and confirming connection requests from the application.

12.4.4 Implementation Issues

There are many technologies which can be used to implement the above IoT architecture, including server technologies, WSN technologies, database technologies, and web-application technologies. The details given in this section is for demonstration purpose only.

12.4.4.1 DSNS

As the DSNS works in a similar way to the DNS server on the Internet, SCO Unix (SCO Group 2010) on which the DNS is based, is chosen as the operation system for the implementation of the DSNS. The SCO Unix supports five different types of configuration which are: primary server, second server, caching-only server, slave model server and client. The first four models match the features of the DSNS and have been employed in building the DSNS.

12.4.4.2 SSP

The duties of the SSP include sensory data extraction, registration of the current RFID sensor networks at the DSNS, and responding to the control commands received from the DSNS. The SSP is implemented as an IP-enabled ZigBee router in the ZigBee RFID sensor network. It consists of two parts: a sensory data extraction part, i.e. RFID sensor network sink node, and a service publishing and query response part. The sensory data extraction part collects the desired sensory data and transmits it to the service publishing and query response part, where data reading, data publishing, and service control are implemented. The service controller stores the connected RFID sensor network specifications and is assigned an IP address by the Sensor Allocator in the DSNS for data communication with the data requestors.

12.4.4.3 Local Database System

The local database system is implemented in MySQL and used to store any predefined exception sensory data. It is made Internet-available by registering the database servers at the DSNS. For dealing with multiple data queries, a thread pool approach is introduced in the local database system. The thread pool holds a number of live threads for instant use. Reusing a thread for multiple tasks can reduce the overhead and the response time caused by thread creation. The local database system may dynamically adjust the number of threads in the thread pool if the number of data queries increase to a level above a certain predefined threshold.

12.4.4.4 End-User Applications

There are two categories of end-user applications which query and consume the data provided by the IoT architecture, standalone applications and web-based applications. The standalone applications are an individual client program for heavy data usage users who need to regularly check the sensory data, and request data from the fixed RFID sensor networks. It could be programmed in JAVA to ensure the cross platform usability. The web-based applications are for the users who just want to do a quick look, but do not like to, or have not installed monitoring software in their local computers in advance. The web-based applications contain the basic functions like logging in/out, searching, registering various interests and displaying sensory data. It can be coded in JSP or other web programming languages. The communication protocol between end-users, the SSP, and the DSNS is TCP/IP. The IP enabled LR-WPAN protocol is used at the SSP to build an upper link with the Internet and a downlink with the RFID sensor networks.

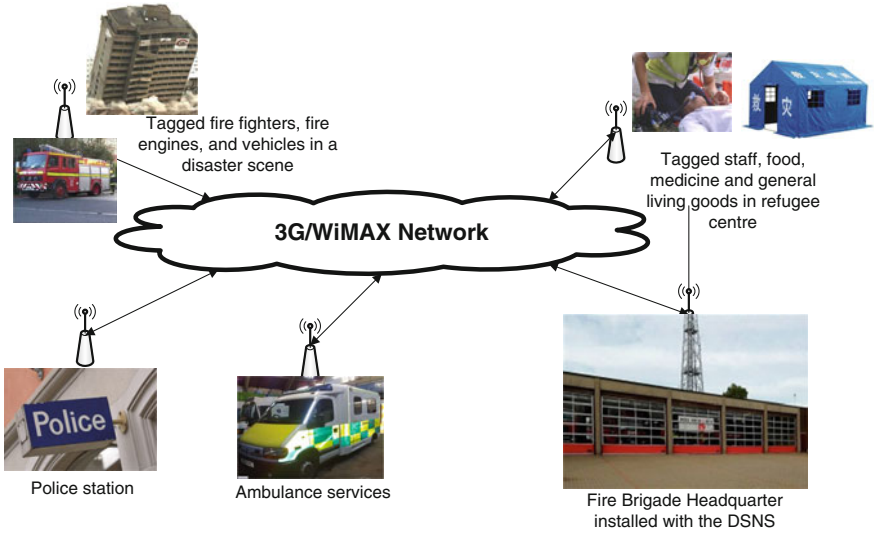


Fig. 12.10 Implementation of the IoT architecture in emergency logistics management (Xu et al. 2013)

12.5 Possible Implementations in Emergency Response

When disaster strikes, the first response teams need not only to rescue and support civilians in the disaster area, but also to maintain their capability of fighting the disaster in the short or long term. Logistics has always been considered as an important factor in emergency response operations. Logistics includes rescue equipment, vehicles and on-site staff as well as food, medicine and general living goods. The emergency response operations require the participation of a wide range of organizations, including fire brigades, police forces, ambulance service, local or national public sectors, and humanitarian aid organizations such as British Red Cross etc. Extensive information and resource sharing or other cooperation between separated organizations is crucial and becoming more common. Figure 12.10 demonstrates how the proposed IoT architecture can be implemented in logistics management for emergency response operations.

In the disaster zone all rescue equipment, fire engines, vehicles, fire fighters, medicine staffs are tagged with RFID tags. The RIFD readers are installed in the entrances to the disaster zone. These RFID readers and tags form a RFID sensor network with some instantly deployed environment sensors. A SSP is installed in a vehicle equipped with a long distance wireless communication device such as a 3G transceiver connecting with a Satellite or WiMAX network. Close to the disaster zone one or more refugee centres are set up, where staff, food, medicine, and general living goods are also tagged. Each refugee centre forms a RFID sensor network and links with a SSP which builds a communication with the remote

DSNS or the end-users over the Satellite or WiMAX network. Each SSP is also equipped with a local database system. The DSNS can be installed in the headquarters of a regional fire and rescue service which provides management functionality to the IoT architecture. The headquarters of the fire brigades, local police stations and hospitals become aware of the latest disaster development and response progress through the IoT architecture. Logistics demands from the first response teams and refugee centres are also made aware to the corresponding organizations.

12.6 Conclusions

This chapter presents the basic concept of the IoT, features, design challenges, and possible architectures. The possible application in emergency response is only the a speculative proposal which might be implemented.

In summary, the IoT should be considered as part of the overall Internet of the future, which is likely to be dramatically different from the Internet we use today. The main differences are that the future Internet or the IoT terminates at smart sensors which connect with the physical objects, but the current Internet terminates at individual computers; the future Internet allows users to reach individual physical objects by assigning an IPv6 address to each IoT objects, but the current Internet is used only for information retrieving and publishing. Therefore, it is possible that the IoT will inspire the Internet evolution that realises the above trend.

References

Association Instituts Carnot: White paper: smart networked objects and internet of things, available online at http://www.instituts-carnot.eu/files/AiCarnot-White_Paper-Smart_Networked_Objects_and_Internet_of_Things.pdf (2011)

Atzori, L., Iera, A., Morabito, G.: The internet of things: a survey. Comput. Netw. **54**(15), 2787–2805 (2010)

European Commission: Internet of things strategic research roadmap, available online at, http://www.internet-of-things-research.eu/pdf/IoT_Cluster_Strategic_Research_Agenda_2009.pdf (2009)

Fleisch, E.: What is the Internet of Things? An economic perspective, Auto-ID Labs White Paper WP-BIZAPP-053, available online at, http://www.autoidlabs.org/uploads/media/AUTOIDLABS-WP-BIZAPP-53.pdf (2010)

Gershenfeld, N., Krikorian, R., Cohen, D.: The internet of things. Sci. Am. **291**(4), 76–81 (2004)

Gill, K., Yang, S.H., Wang, W.: A scheme for preventing low-level denial of service attacks on wireless sensor network based home automation systems. IET Wirel. Sens. Syst. **2**(4), 361–368 (2012)

IETF IPv6 over IEEE 802.15.4 low-power wireless personal-area-network, available online at http://www.6lowpan.org (2007)

Kosanovic, M.R., Stojcev, M.K.: Connecting wireless sensor networks to Internet. Facta Universitaties, Series: Mech. Eng. **9**(2), 169–182 (2011)
Llic, A., Staake, T., Fleisch, E.: Using sensor information to reduce the carbon footprint of perishable goods. IEEE Pervasive Comput. **8**(1), 22–29 (2009)
Mattern, F., Floerkemeier, C.: From the internet of computers to the internet of things. Informatik-Spektrum **33**(2), 107–121 (2010)
Niyato, D., Hossain, E., Camorlinga, S.: Remote patient monitoring service using heterogeneous wireless access networks: architecture and optimization. IEEE J. Sel. Areas Commun. **27**(4), 412–423 (2009)
Sarma, S., Brock, D.L., Ashton, K.: The Networked Physical World. TR MIT-AUTOID-WH-001 MIT Auto-ID Centre, Cambridge (2000)
SCO Group. SCO Official support document, available online at http://www.sco.com (2010)
Welbourne, E., Battle, L., Cole, G., Gould, K., Rector, K., Raymer, S., Balazinska, M., Borriello, G.: Building the internet of things using RFID: the RFID ecosystem experience. IEEE Internet Comput. **13**(3), 48–55 (2009)
Xu, R.: Federated sensor network architecture design for the IoT systems, Loughborough University PhD thesis (2013)
Xu, R., Yang, L., Yang, S.H.: Architecture Design of Internet of Things in Logistics Management for Emergency Response, IEEE International Conference on Internet of Things (iThings), pp. 395–402. Beijing, China, (2013)
Yang, H., Yang, L., Yang, S.H.: Hybrid Zigbee RFID sensor network for humanitarian logistics centre management. J. Netw. Comput. Appl. **34**(3), 938–948 (2011)
Yang, L., Yang, S.H., Plotnick, L.: How the Internet of Things technology enhances emergency response operations. Technol. Forecast. Soc. Change **80**(9), 1854–1867 (2013)
Yang, S.H.: Internet-based Control Systems. Springer, Berlin (2011)

Chapter 13
ZigBee Smart Home Automation Systems

Keywords Home automation system · Energy saving · Smart home

13.1 Introduction

There are many definitions of home automation available in the literature. Bromley et al. (2003) described home automation as 'the introduction of technology within the home to enhance the quality of life of its occupants, through the provision of different services such as tele-health, multimedia entertainment and energy conservation'.

There has been significant research into the field of home automation. The X10 industry standard, developed in 1975 for communication between electronic devices, is the oldest standard identified in the author's review, providing limited control over household devices through the home's power lines. Recently, research into the field of home automation has continued to receive much attention in academia. Al-Ali and Al-Rousan (2004) developed a Java based home automation system. An embedded board physically connected all the home automation devices and, through integration with a personal computer (PC) based web server, provided remote access to the system. The use of Java technology, which incorporates built-in network security features, produces a secure solution. However, the system requires an intrusive and expensive wired installation and the use of a high end PC. Sriskanthan et al. (2002) introduced a Bluetooth based home automation system, consisting of a primary controller and a number of Bluetooth sub-controllers. Each home device is physically connected to a local Bluetooth sub-controller. The home devices communicate with their respective sub-controller using wired communications. From the sub-controller all communications are sent to the primary controller using wireless communications. It is desirable for each home device to have a dedicated Bluetooth module. However, due to the financial expense of Bluetooth technology, a single module is shared amongst several devices. This architecture reduces the amount of physical wiring required and hence the intrusiveness of the installation, through the use of wireless technology. Ardam and

S.-H. Yang, *Wireless Sensor Networks*, Signals and Communication Technology,
DOI: 10.1007/978-1-4471-5505-8_13,

Coskun (1998) introduced a phone based remote controller for home and office automation. The system differs from those describe previously in that all communications occur over a fixed telephone line and not over the Internet. The system can be accessed using any telephone that supports dual tone multiple frequency (DTMF). The disadvantages of this system are threefold: users are not provided with a graphical user interface, users have to remember an access code, and they have to remember which buttons to press for the control of connected devices.

The most significant home automation offering from a large enterprise is the LG HomNet. The system core resides on a personal computer located in the home. The system provides control over compliant appliances including washing machines, microwaves, air conditioners, ovens and refrigerators produced by LG (LG HomNET 2013). The home automation offerings in the UK include the British Telecommunications (BT) Home Hub and Sky Multiroom. The BT Home Hub incorporates high speed Internet access, a wireless personal area network and a multimedia entertainment package (BT 2013). The Sky Multi-room package includes, a multimedia network, which streams Sky content to different rooms in the home (Sky TV 2007).

Attempts have been made to provide network interoperability and remote access to home automation systems through the development of home gateways. Saito et al. (2000) defined a home gateway as 'the point of ingress between a personal area network and a public access network'. They developed a web server based home gateway to interconnect IEEE 1394, with a power line based home automation system, and the Internet. Ok and Park (2006) proposed a home gateway based on the open service gateway initiative (OSGI), which allows service providers to access home automation systems for administration and maintenance services. The proposed system is divided into two subsystems. The first is the digital home service distribution and management system (DSM), which provides a user interface for the control and monitoring of connected home automation devices. The second is the Home Gateway, which is responsible for managing the home automation system. This open architecture raises privacy issues, which, for some users, may be much greater than the advantages offered by granting third party access.

13.2 Analysis of the Existing Home Automation Systems

The adoption of home automation technology by consumers has been limited. The issues limiting wide spread consumer adoption can be grouped into five general categories (Gill et al. 2009). Firstly, complex and expensive architecture: the existing systems architectures generally incorporate a personal computer for the purposes of network management and provision of remote access. This adds additional complexity to the system, hence increasing the overall expense. Secondly, intrusive installation: the majority of systems require varying levels of

physical wiring in their architectures. This, in some cases, is due to the expense of the alternative wireless technologies. Hence, these systems require intrusive and expensive installations. Thirdly, lack of network interoperability: both home networks and the home automation systems, which utilise them, have been developed and adopted in an unplanned and ad-hoc manner. This has led to a home environment consisting of a complex maze of heterogeneous networks. These networks and the systems that utilise them normally offer little interoperability; leading to three potential problems.

- Duplication of monitoring activities, due to lack of interoperability;
- Possibility of interference, between co-existing networks; and
- Potential for two simultaneous, autonomous actions on co-existing networks, to interact, resulting in an undesirable outcome.

Fourthly, interface inflexibility: the existing systems offer varying approaches for users to control and monitor the connected devices. However, this is normally limited to a single method of control, which offers users limited flexibility. The systems, which provide more than one interface device, normally provide different user interfaces, which increase the risk of confusing users. Finally, security and safety: the existing approaches have not focused on security and safety issues that may arise from their implementation. Moreover, the systems that offer some degree of security have neglected the problems with sharing information between devices produced by multiple vendors for the purposes of establishing security.

13.3 Home Automation System Architecture

In order to overcome the above drawbacks this section presents a novel, stand alone, low-cost and flexible ZigBee based home automation system. The architecture is designed to reduce the system's complexity and lower financial costs. Hence, the system endeavours wherever possible, not to incorporate complex and expensive components, such as a high-end personal computer. The system is flexible and scalable, allowing additional home appliances designed by multiple vendors, to be securely and safely added to the home network with the minimum amount of effort. The system allows home owners to monitor and control connected devices in the home, through a variety of controls, including a ZigBee based remote control, and any Wi-Fi enabled device that supports Java. Additionally, users may remotely monitor and control their home devices using any Internet enabled device with Java support. A home gateway is implemented to facilitate interoperability between heterogeneous networks and provide a consistent interface, regardless of the accessing device. A virtual home pre-processes all communications before they are activated on the real home automation system. All communications are checked for security and safety before being allowed to continue to their respective destinations.

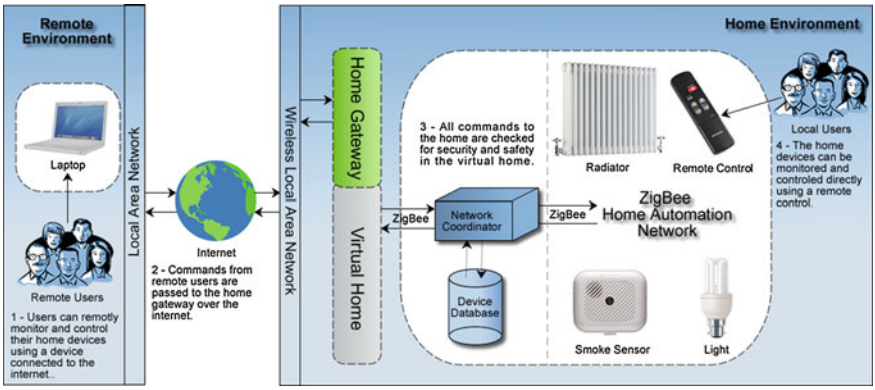


Fig. 13.1 Conceptual architecture overview

As depicted in Fig. 13.1, the ZigBee home automation system consists primarily of four steps. Remote user can access the system using the Internet. The remote user's communications cross the Internet until they reach the home network. They are then wirelessly transmitted to the Home Gateway using the homes Wi-Fi network. The Home Gateway is integrated with a virtual home. These communications are checked and processed by the home gateway and virtual home, as discussed in greater detail later. This checking process involves communication with the home networks coordinator, which is integrated with the home's device database and contains the status of all connected devices. Once checked, the communications are sent to the real home automation system and their respective device. Additionally, a local ZigBee based remote control can be used to directly control connected devices.

The home gateway, as depicted in Fig. 13.1, is charged with providing interoperability between different connecting networks. The home gateway provides two primary functions for the proposed architecture. Firstly, the home gateway provides data translation services between the Internet, Wi-Fi, and ZigBee networks. Secondly, the home gateway provides a standardised user interface for devices connecting to the ZigBee home network, remotely using the Internet or locally using the Wi-Fi network.

The virtual home, as depicted in Fig. 13.1, is responsible for the administration of security and safety for the home automation system (Gill et al. 2013). The virtual home, as the name suggests, is a virtual environment where the actions requested by users are checked. For the purposes of security, all the messages received by the virtual home are checked by authenticating the senders, checking the integrity of the messages to ensure they have not been tampered with, and protecting the confidentiality of messages through the use of encryption. The system's safety is protected by ensuring the commands received are appropriate for the respective home network and that all changes requested fall within the specified safety limits. The primary objective of the virtual home is to prevent any

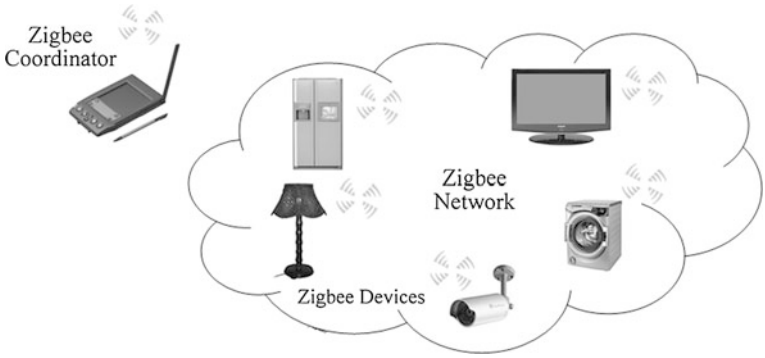


Fig. 13.2 Zigbee home automation architecture

event that may pose a security or safety concern from implementation in the home networks.

Figure 13.2 depicts the general architecture of a Zigbee based home automation network. The Zigbee coordinator is responsible for creating and maintaining the network. Each electronic device (i.e. Washing Machine, Television, Lamp etc.) in the system is a Zigbee device managed by the coordinator. All communication between devices propagates through the coordinator to the destination device. The wireless nature of ZigBee helps overcome the intrusive installation problem with the existing home automation systems identified earlier. The ZigBee standard theoretically provides 250 kbps data rate, and as 40 kbps can meet the requirements of most control systems, it is sufficient for controlling most home automation devices. The low installation and running cost offered by ZigBee helps tackle the expensive and complex architecture problems with existing home automation systems, identified earlier.

13.4 System Implementation

13.4.1 Implementation of ZigBee Home Automation Network

The ZigBee home automation network consists of a coordinator, routers and several end devices. The coordinator is responsible for initiating the ZigBee network. During the network initialisation phase, the coordinator scans the available radio channels to find the most suitable one. Normally, this will be the channel with the least activity, in order to reduce the level of interference. It is possible to limit the channels scanned, for example excluding those frequencies ranges used by the Wi-Fi network co-existing in the home environment. The coordinator is preprogrammed with the personal area network identifier (PAN ID). All home devices

connected to the ZigBee home automation network are assigned a fixed 64-bit MAC address. Additionally, each device is assigned a dynamic 16 bit short address that is fixed for the lifetime of the network. At this stage of the network initialisation, the coordinator assigns itself the short address 0x0000. After the coordinator's initialisation phase the coordinator enters "coordinator mode", during this phase, it awaits requests from ZigBee devices to join the network.

The ZigBee devices developed for the home network includes a light switch, a radiator valve, a safety sensor and a ZigBee remote controller. A ZigBee end node has been integrated with these devices. As the devices are started, during their respective initialisation stage, the node scans the available channels to identify the network it wishes to join. There may be multiple networks in the same channel; these networks are normally distinguished by their PAN ID. The node selects which network to join based on the PAN ID. The standard implementation of most ZigBee networks prevents unauthorised devices joining the network by providing a short user defined period where device may join. This, in our opinion, does not, on its own, provide sufficient network security. To enhance the systems security the proposed system encrypts all device communications including the requests to join the home network with a private key. Only those devices that are in possession of the correct private key can successfully connect to the home network. The devices that are permitted to join the network are recorded in the device database and stored on the network coordinator. A partially connected mesh topology was adopted for the ZigBee home automation network. Due to the nature of the home environment where communication interference is constantly fluctuating, the advantage of increased communication routes being available because of the adoption of a mesh topology outweighs the added routing complexity. The Zigbee home automation network was implemented using Jennic's ZigBee wireless chip JN5139.

13.4.2 Home Gateway Implementation

A thorough review of existing home gateway technologies revealed that no off-the-shelf solution exists that provides the functionality specified in the requirements for the home gateway, discussed previously. This included the provision of interoperability between the Internet, Wi-Fi and ZigBee networks. Hence, it was necessary to develop a bespoke home gateway, as shown in Fig. 13.3. The home gateway consists of a Digi Wi-Me module, a Jennic JN5139 Microcontroller and a power supply. The Digi Wi-Me module provides low cost and embedded serial to Wi-Fi connectivity. The Jennic Microcontroller provides the connection to the ZigBee network. The Digi module connects to the home's local Wi-Fi network and the Jennic microcontroller connects to the ZigBee home network as an end device.

The home gateway once started enters the configuration stage. During the configuration stage the embedded Digi Wi-Me module establishes a connection with a local Wi-Fi network. The parameters for the Wi-Fi connection such as

Fig. 13.3 Home gateway

network SSID and security parameters are preconfigured. Simultaneously, the Jennic microcontroller searches for a ZigBee home network and, as discussed previously, establishes a connection. As with the Wi-Me module, the Jennic microcontroller's connection parameters are preconfigured. This concludes the configuration stage.

Once the home gateway has been initialised, an idle state is entered into, until input is received. Input can originate from both the Wi-Fi network for input to the ZigBee network, or conversely from the ZigBee network for output to the Wi-Fi network. Input from the Wi-Fi network normally takes the form of commands from user interface devices. The input from the ZigBee network normally takes the form of responses to commands received earlier from user interface devices.

13.4.3 Virtual Home Implementation

The virtual home is a software construct developed in C. It is implemented on the home gateway. All communication and instructions are checked, as illustrated in Fig. 13.4, for security and safety, in the virtual environment, before implementation in the real home environment. The virtual home waits for input from an external source. All devices on the ZigBee network incorporate the Jennic JN5139 microcontroller and a dedicated AES Coprocessor. Sensitive communications on the home network are encrypted. Hence, the message payload received by the virtual home from legitimate sources will be encrypted with a valid symmetric key, and then sent to the home network. Once the security of messages has been established, the virtual home checks the safety implications of the messages. After decryption, the destination device address is extracted from the message and checked to ensure its existence in the device database. Once the device's existence

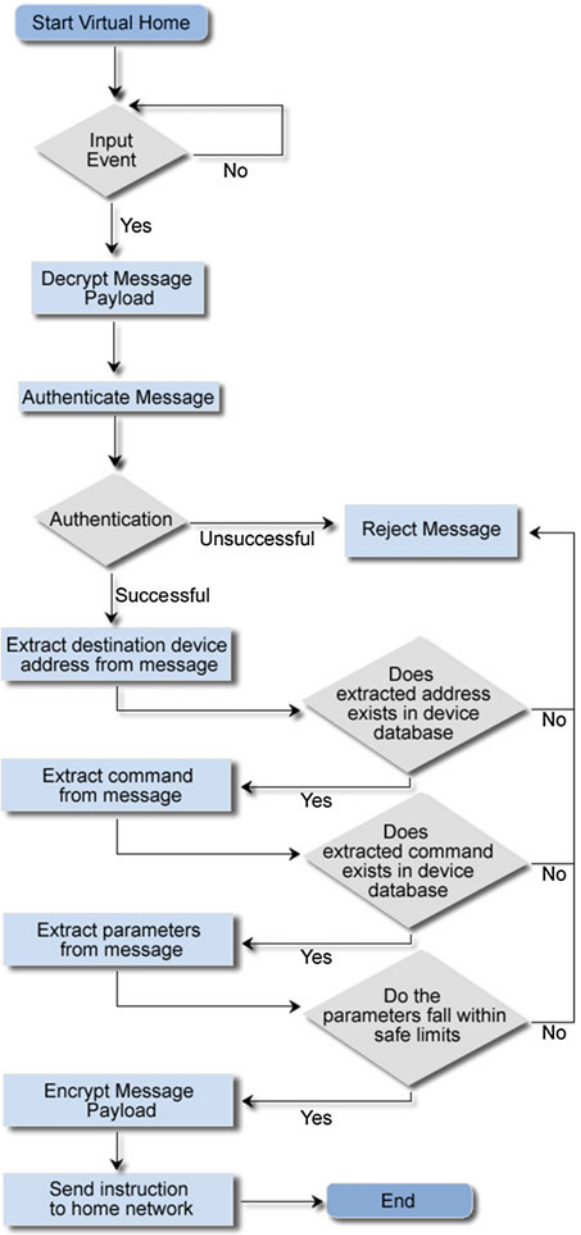


Fig. 13.4 Virtual home flow chart

on the network has been established, the command and parameters included in the message are extracted. The existence of the command for the respective device is checked to ensure the real device offers the requested functionality. The extracted parameters are compared against predefined safe ranges for the respective device

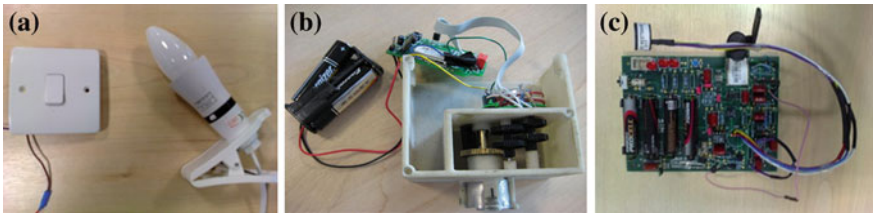


Fig. 13.5 **a** ZigBee operated light bulb in the off state; **b** ZigBee based automatic radiator valve; **c** ZigBee safety sensor

and command. Once the message has been processed by the virtual home algorithm for security and safety, and declared safe, it is re-encrypted and forwarded to the real home network device.

13.4.4 Home Automation Devices Developed

To demonstrate the feasibility and effectiveness of the proposed system three devices; a light switch, a radiator valve, and a safety sensor, were developed. These devices are depicted in Fig. 13.5a–c respectively.

Light Switch: A conventional light switch was integrated with a Jennic 5139 microcontroller, as shown in Fig. 13.5a. In this prototype, the user could access the light switch, detect the lights current state ("On" or "Off"), and adjust the state accordingly.

Radiator Valve: A prototype automatic radiator valve was developed and integrated with a Jennic 5139 microcontroller, as shown in Fig. 13.5b. The valve can be manually controlled, as are conventional valves, but also remotely monitored and controlled.

Safety Sensor: The safety sensor has special characteristics of interest. For instance, unlike most devices, the safety sensor has to continuously monitor its environment and provide feedback. This reduces the time the device can operate in sleep mode, hence considerably reducing the battery life. A safety sensor was developed (see Fig. 13.5c) to investigate the potential viability of the system with a mass-market end device that places a large demand on system resources. The safety sensor developed incorporated temperature, carbon monoxide, flame, and smoke sensors.

13.5 Systems Evaluation

The viability of the home automation architecture was evaluated through a test-house experiment and a field trial with the developed radiator valve. The radiator valve, as depicted in Fig. 13.5b, was tested in a test-house, on the ground floor as

Fig. 13.6 Radiator valve in place

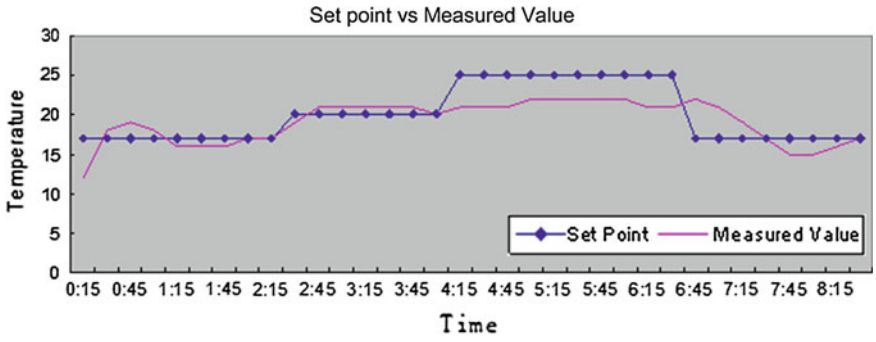


Fig. 13.7 Automatically controlled room temperature

depicted in Fig. 13.6. The radiators existing TRV valve was replaced with the prototype automatic radiator valve.

The local controller was placed 2 m away from the radiator and wirelessly connected to the home automation system. The results of the experiments are summarised in Fig. 13.7. The graph shows the desired temperature set by the user against the actual temperature of the radiator at a period of 30 min. It can be seen that the actual temperature of the radiator quickly adjusted to the desired temperature set by the user. However, the actual temperature could not reach 25 °C. It was surmised that the radiator was too small to heat such a large room to this temperature.

The experimentation highlighted that a radiator valve could successfully be implemented using the ZigBee communication standard and monitored and controlled using the home automation system. This successful evaluation supports and demonstrates the potential of the ZigBee smart home automation system to be easily adaptable from the lab environment to the commercial market.

13.6 Conclusion

This chapter has reviewed the existing state of home automation systems, and identified five areas that have hindered consumer adoption of such technologies. Briefly, the areas include: the complexity and expense of the architectures adopted by existing systems, the intrusiveness of the system installations, the lack of interoperability between different home automation technologies, and the lack of interoperability between systems developed by different manufacturers that utilize the same technology. Interface inflexibility and the inconsistent approaches adopted towards security and safety, are also problems. A ZigBee smart home automation system is presented in this chapter to overcome the drawbacks of the existing systems. The use of ZigBee communications technology helps lower the expense of the system and the intrusiveness of the respective system installation. The incorporation of the virtual home concept coordinates the systems security and safety efforts in a clear and consistent manor. The inclusion of a home gateway helps overcome the problems of network interoperability. The home gateway developed provides interoperability between the local ZigBee and Wi-Fi networks and the Internet. The feasibility and appropriateness of the home automation system architecture and technologies in the creation of a low cost, flexible and secure system has been successfully evaluated both through a test-house experiment and field trials.

References

Al-Ali, A.R., Al-Rousan, M.: Java-based home automation system. IEEE Trans. Consum. Electron. **50**(2), 498–504 (2004)

Ardam, H., Coskun, I.: A remote controller for home and office appliances by telephone. IEEE Trans. Consum. Electron. **44**(4), 1291–1297 (1998)

British Telecom: BT home hub. Available online at http://www.homehub.bt.com (2013)

Bromley, K., Perry, M., Webb, G.: Trends in smart home systems, connectivity and services. Available online at www.nextwave.org.uk (2003)

Gill, K., Yang, S.H., Yao, F., Lu, X.: A zigbee-based home automation system. IEEE Trans. Consum. Electron. **55**(2), 422–430 (2009)

Gill, K., Yang, S.H., Wang, W.: Secure remote access to home automation networks. IET Inf. Secur. **7**(2), 118–125 (2013)

LG HomNET: Solution models. Available online at http://global.dreamlg.com (2013)

Ok, S., Park, H.: Implementation of initial provisioning function for home gateway based on open service gateway initiative platform. In: Proceedings of the 8th International Conference on Advanced Communication Technology, pp. 1517–1520 (2006)

Saito, T., Tomoda, I., Takabatake, Y., Ami, J., Teramoto, K.: Home gateway architecture and its implementation. IEEE Int. Conf. Consum. Electron. 194–195 (2000)

Sky TV: Prices and packages. Available online at http://www.sky.com/portal/site/skycomskyproducts/skytv/pricesandpackages (2007)

Sriskanthan, N., Tan, F., Karande, A.: Bluetooth based home automation system. Microprocess. Microsyst. **26**(6), 281–289 (2002)

Chapter 14
Building Fire Safety Protection: SafetyNET

Keywords Fire safety · Zigbee, building environment monitoring

14.1 Introduction

Fire kills. Preventing fires saves lives and reduces injuries. Preventing fires also saves money. So it makes sense to protect people and to prevent fires from happening in the first place. It also makes sense to ensure that we fight fires safely and in the most efficient way (Yang 2007). Accurate and reliable information is critical for effective emergency response. Control centre staffs make critical decisions on the deployment of their available resources, based on information about an incident. First responders being deployed also rely on such information to prepare a practical response plan. Therefore, the accuracy and reliability of information on an incident has impacts on decisions, which can be a matter of life or death.

Yang et al. (2009) showed that the following four categories of information require collection, sharing and presentation in an on-site emergency response information system, not only for protecting emergency responders, but also for ensuring the success of the emergency response operations.

- Environmental conditions: When the first responders arrive at the incident scene, they have very limited information about the environment, and do not know whether the building/underground station is safe to enter or how to most efficiently deal with the hazard. Many front line responders may be facing unfamiliar hazards. Decision makers need to be aware of these hazards and have an overview of the environmental conditions. Then they can consider a plan to deal with these hazards before they dispatch their subsequent responders to cope with them.
- Information on response participants: Some disaster situations involves many hundreds of individuals from different organisations cooperating their response. Knowing who is involved in the response, what capability they are offering, and what resources they are bringing to the scene, gives incident commanders

S.-H. Yang, *Wireless Sensor Networks*, Signals and Communication Technology,
DOI: 10.1007/978-1-4471-5505-8_14,

information to enable them to determine the most effective and coordinated approach to the situation.

- Status of casualties: Obtaining and rapidly sharing the latest casualty data among involved organisations, and reporting accident locations, causes, and severity among involved organisations is critical to ensure that the responders can take appropriate rescue measures and quickly coordinate emergency medical services during the response.
- Available resources: Once a major disaster occurs, large amounts of equipment and other resources are quickly delivered to the area by many governmental and non-governmental organisations. Often, there is no central control or storage for the equipment. Collecting and sharing information about available equipment is critical to ensuring responders find what they need from the stock of equipment which has arrived at the incident scene.

14.2 Information Infrastructure

SafetyNET provides an information infrastructure to enable buildings, firefighters, fire tenders, and their control centre to efficiently communicate during natural or man-made disasters by using sensor networks, wireless communications, DAB (Digital Audio Broadcasting) and Tetra (Terrestrial Trunked Radio) technologies. The information infrastructure is comprised of three layers (Yang and Frederick 2006), as shown in Fig. 14.1.

The bottom layer comprises a robust wireless sensor network installed in and around the building. The sensor network utilizes robust sensor nodes to detect any changes in the environment at any specified location. The sensor network can take the place of existing fire alarm networks, meaning no previous installations are required. Information collected flows through the sensor network and is then transmitted to the fire tender network.

The middle layer comprises a vehicle-mounted mobile network installed on the fire tenders. It is achieved by upgrading the newly introduced vehicle mounted mobile data systems (VMDS) and adding, not only the up-link to the control centres, but also the downlink to the sensor network. The real time information about the building, occupants, and the locations of the fire fighters is collected from the sensor network, transmitted to and presented at the fire tender network. Up-to-date information about the building such as the floor plan and hydrant status is downloaded from the central database located at the control centre to the fire tender's network on their way to an incident. DAB is employed between the bottom and middle layers in order to maintain a time critical one-way communication channel between the fire tenders and emergency personnel.

At the top layer is the central facility located at the control centre of a fire brigade. An emergency response management system at the control centre will

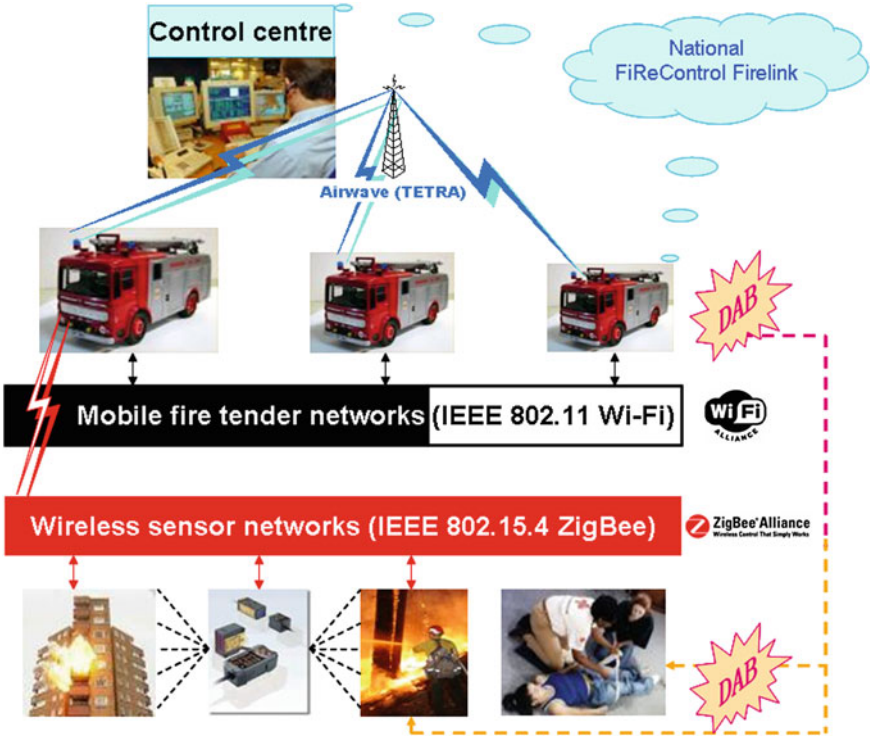


Fig. 14.1 Information architecture of SafetyNET system

provide the fire-fighters with up-to-date critical information and remotely monitor the latest development of incidents.

14.3 SafetyNET Specific Devices

A typical SafetyNET system consists of four types of devices: a ZigBee coordinator, ZigBee routers, ZigBee end devices, and ZigBee adaptors. The ZigBee coordinator is the initiator and network manager of the SafetyNET sensor network. It establishes and maintains the network. The functions of network maintenance and the routing protocol used are implemented in the ZigBee stack. The ZigBee router is similar to the ZigBee coordinator, except it does not establish the network. The ZigBee router extends the coverage of the SafetyNET systems. It also takes the responsibility for adopting new ZigBee end devices into the network, processing the leaving request of the end devices, and implementing the routing protocol. The ZigBee end devices are the SafetyNET sensor nodes, which are installed inside the building wherever the environment condition could change and are crucial to the building safety. The

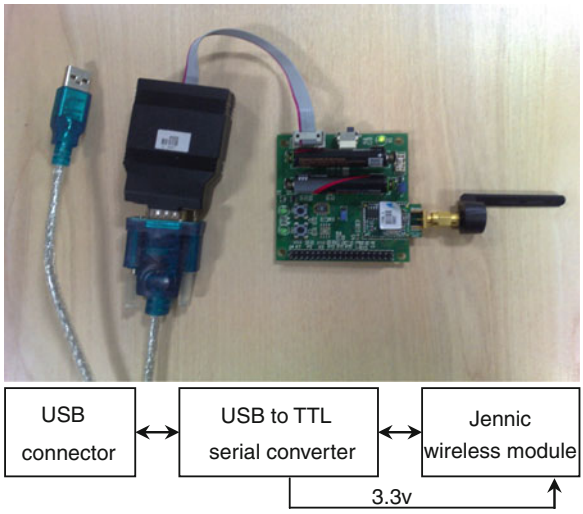


Fig. 14.2 ZigBee adaptor prototype

ZigBee adaptor is a ZigBee device, which is designed to allow any computer to access the ZigBee network. Figure 14.2 shows the ZigBee adaptor prototype. One side is a USB port that connects with a computer; while on the other side is a ZigBee antenna which accesses the ZigBee network. Figure 14.3 summarizes the SafetyNET specific devices and their functionalities.

14.4 Mobile Fire Tender Networks

The SafetyNET system chooses the IEEE 802.11 Wi-Fi communication protocol to instantly form the mobile fire tender network. In practice, mobile fire tender Wi-Fi network does not exist before any fire tender arrives on the incident site. The whole three-layer SafetyNET is only fully created when one or several fire tenders (engines/vehicles) arrive on-site, so the system activation starts from the Wi-Fi network, which is established by default by the Wi-Fi coordinator in the first fire tender to arrive (main vehicle). Follow-up vehicles, adapters, are automatically detected and added into the already-existed Wi-Fi network via their in-vehicle Wi-Fi. Figure 14.4 illustrates the principle of the Wi-Fi server/database transferring during the emergency operation, where Fire tender1 is the main vehicle and Fire tender2 arrived later. The ZigBee adapter and Wi-Fi coordinator in Fire tender2 are inactive as the only down-link from the mobile fire tender network to the building WSN is maintained in Fire tender1 and Fire tender1 is also working as the Wi-Fi coordinator.

Therefore in working conditions, only the Wi-Fi coordinator and the ZigBee adapter in the main vehicle (server vehicle) are activated. With a Wi-Fi network established by the server vehicle, the server application retrieves sensor data from the WSN, which is stored in a database via the ZigBee adapter equipped in the

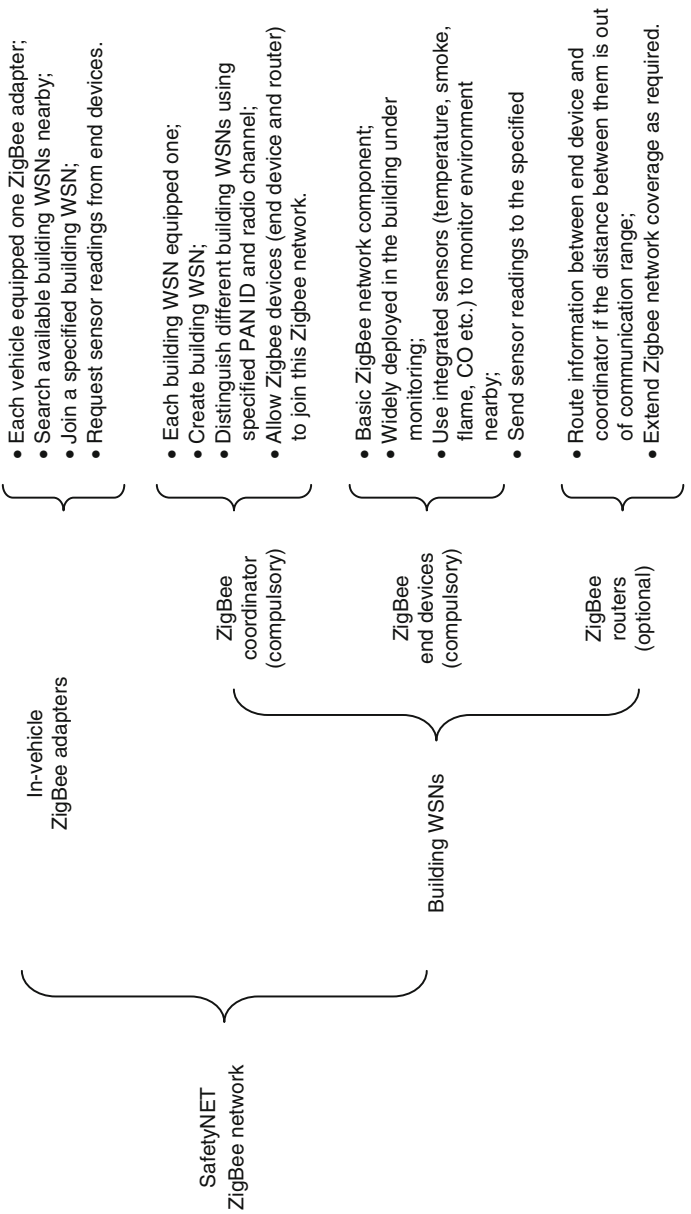


Fig. 14.3 SafetyNET ZigBee network devices and functionalities

server vehicle. The application then shares the information to all vehicle-mounted data terminals (computers) within the Wi-Fi network by displaying it onto a web-based graphical user interface (GUI). Displayed information is automatically updated every 2–3 s.

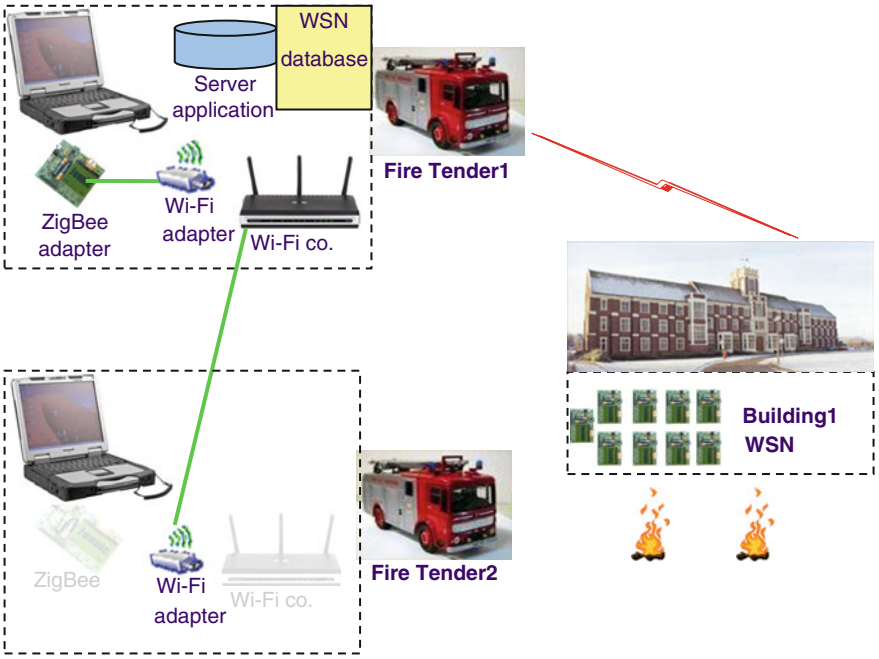


Fig. 14.4 Mobile fire tender network

At times, one of the follow-up vehicles (non-main vehicle or client vehicle) requests to be the new main vehicle (new server vehicle) either because it is a command unit arriving later and needs to get control of the data/database, or because the existing main vehicle has to leave the site for other duties. In this situation, after confirmation, the ZigBee adapter on the potential main vehicle will be activated and the server role together with the database will be transferred from the current main vehicle to the potential main vehicle. Finally, the new main vehicle would commence the retrieval of sensor data from the building WSN using its own ZigBee adapter and sharing the information to the other terminals. Of course, new data will be appended into the old database obtained; hence the sensor data is wholly maintained in a unique version. The previous main vehicle may then leave the site, and then the Wi-Fi coordinator in the new main vehicle will automatically re-establish a new Wi-Fi network for continuing information sharing within several seconds.

14.5 SafetyNET Wireless Sensor Networks

As described in Sect. 14.3, the SafetyNET wireless sensor network consists of the SafetyNET coordinator, routers, end devices, and adaptors. The SafetyNET wireless sensor network sits at the bottom layer in the information architecture shown in

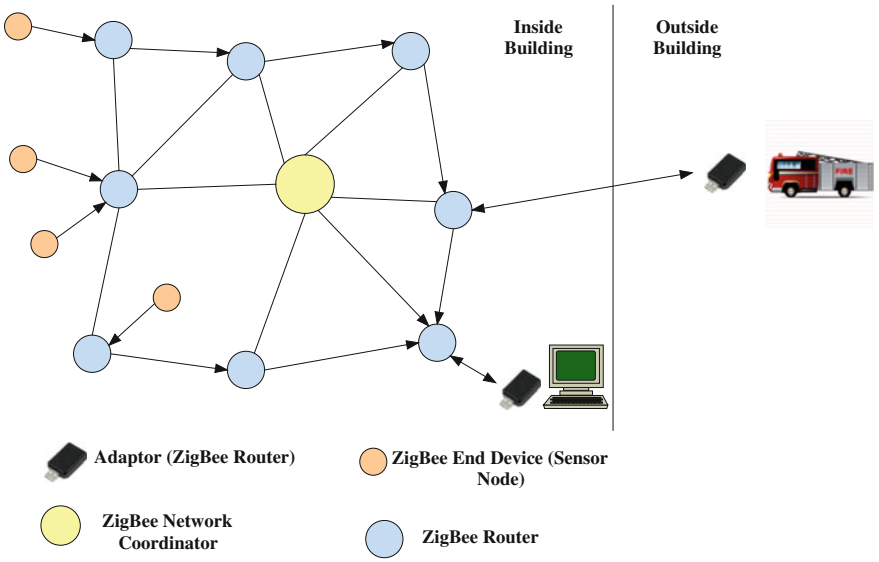


Fig. 14.5 SafetyNET wireless sensor network

Fig. 14.1. The SafetyNET coordinator and routers form a ZigBee-based mesh network in which various end devices connect with their routers, and consequently the mesh network can sense change in any environment condition. The communication link with the mobile fire tender network is maintained through the ZigBee adaptor. Figure 14.5 shows the structure of the SafetyNET mesh network, where the ZigBee coordinator and routers are supplied by the mains power, the end devices are powered by batteries and the adaptor is connected with a building management system and a tough book computer installed in a fire tender through a USB port.

14.5.1 SafetyNET Coordinator

The SafetyNET coordinator is the initiator of a ZigBee wireless sensor network. It is also a special device capable of acting as a ZigBee router in the network. The operations of the coordinator mainly include network initialization and network maintenance. The ZigBee stack handles network maintenance and routing protocol implementation. Figure 14.6 illustrates the state machine designed for the SafetyNET coordinator. The hardware initialization is for initializing the peripheral devices installed with the coordinator, including buttons, LCD and LED. These peripheral devices can be used to support interactions with the users. Network stack initialization is the setting of the network PAN ID and network channel. Network conflict handling may require the coordinator to adjust these two parameters. If there is no conflict or the conflict has been solved, the coordinator

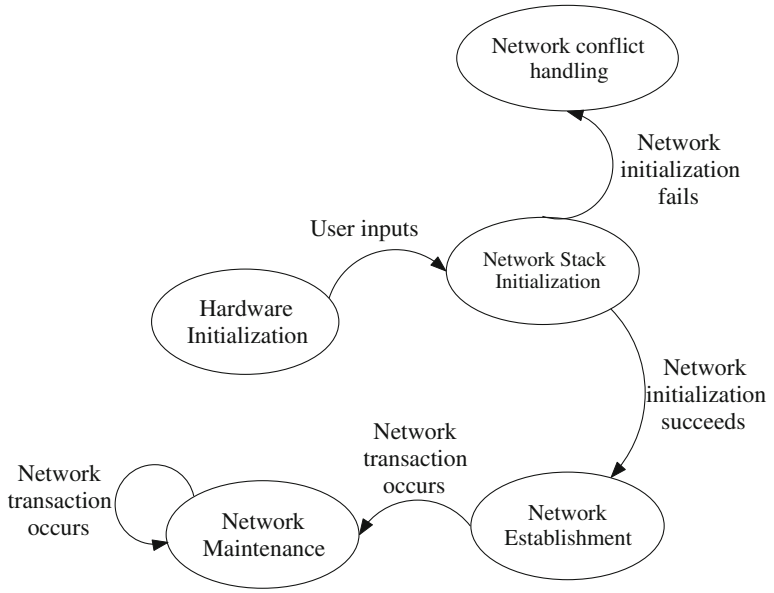


Fig. 14.6 State machine of SafetyNET coordinator

reaches the network establishment state and its PAN ID is reserved for the wireless sensor network, which means that other ZigBee coordinators cannot use the same PAN ID if they are in close vicinity of the SafetyNET wireless sensor network. In the Network maintenance state, the coordinator is responsible for accepting any request to join or leave the network made by a router device.

14.5.2 SafetyNET Routers

The functions executed by the SafetyNET router are similar to the SafetyNET coordinator, except it is not able to establish the network. Figure 14.7 illustrates the state machine of the SafetyNET router. Code security checking is to protect the device from being replaced or modified by any potential intruders. If the code security checking fails, the device will halt with any further actions. The Parameter initialization state is to load the customized or default parameters from the on-board memory (usually stored in EEPROM). When the parameter loading is completed, the parameters will be validated in order to ensure that there no error occurred during the loading process. The Network stack initialization state is when the device joins the network. If it fails to join the network, the device will go into sleep mode for a while and then wake up and try again. The Network maintenance state is a response to the request of joining or leaving the network from other

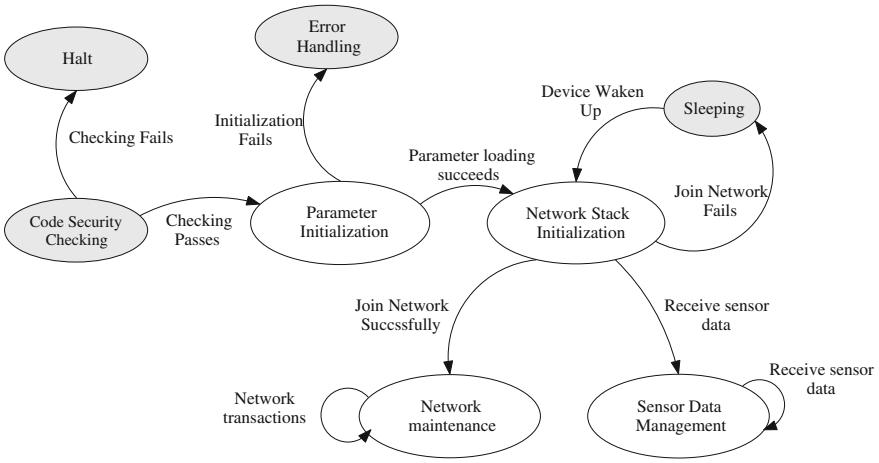


Fig. 14.7 State machine of the SafetyNET router

devices and implementing the routing protocol. The response might be the adoption or rejection of the new device, and notifying the network that a device is leaving the network. The Sensor data management state periodically collects sensor data and then relays the received data to their destination. The received data is stored in a local buffer and regularly sent to their destination.

14.5.3 SafetyNET End-Devices

ZigBee end devices are installed at any position where any change in the environment condition is crucial to the building safety monitoring. Figure 14.8 illustrates the state machine of the SafetyNET end device. The first three states code security checking, parameter initialization and network stack initialization are the same as the SafetyNET router. After the end device successfully joins the network, it goes into the sensor reading state. The sensor reading should be sent to their destination during the wireless transmission state. The routing protocol implemented in the routers will ensure that the data transmission happens. The Parameters modification state is to update the parameter of the end devices if necessary. To save energy, the SafetyNET end device will enter into sleeping mode. The content of the network stack and memory will be stored before sleeping. When the system wakes up, the SafetyNET end device will restore the memory and resume the sensing activities. In emergency situations, the sleeping state is bypassed. The parameter modification state is immediately followed by the sensing reading state.

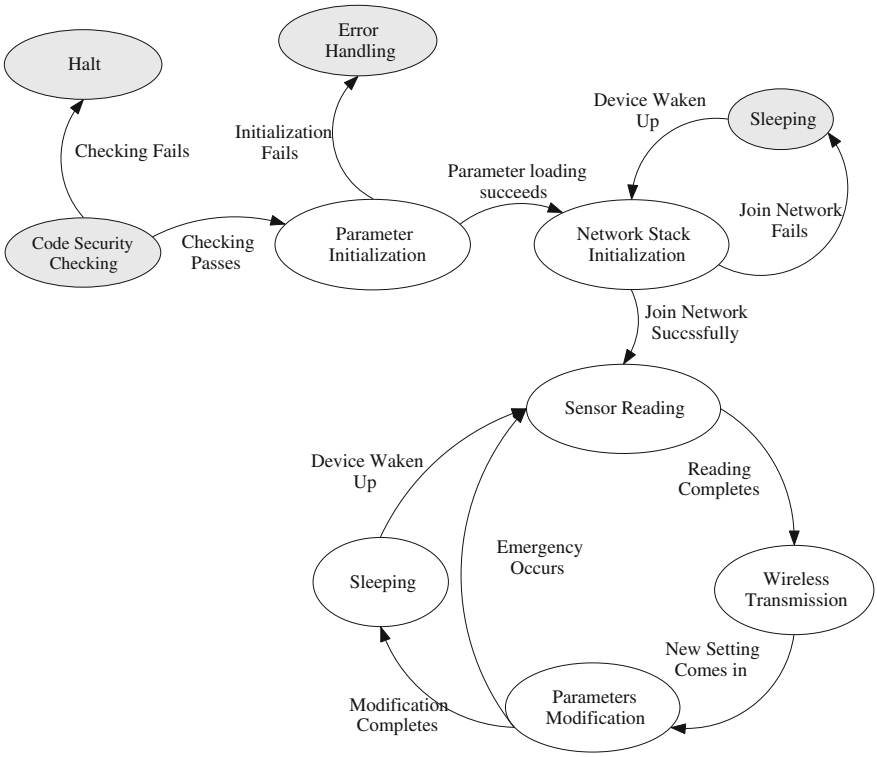


Fig. 14.8 SafetyNET end device state machine

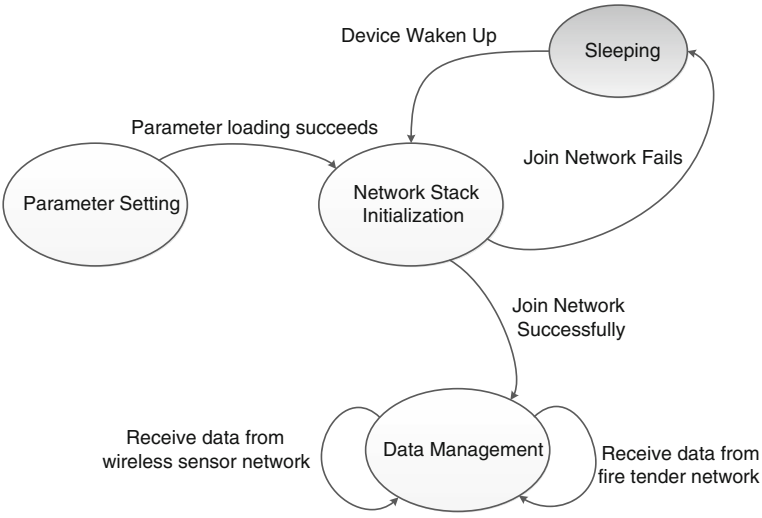


Fig. 14.9 State machine of the SafetyNET adaptor

Fig. 14.10 Training building used in the field trial and a wireless node deployed inside

14.5.4 SafetyNET Adaptors

The SafetyNET adaptor is used by the mobile fire tender network to access the ZigBee network. Figure 14.9 shows the state machine of the SafetyNET adaptor. Parameter setting includes automatically choosing the PAN ID of the SafetyNET wireless sensor network and the IEEE 802.14 channel. The Network stack initialization state is the same as the router and end device. After joining the SafetyNET wireless sensor network, the adaptor acts as a mediator between the fire tender network and the SafetyNET wireless sensor network.

14.6 Field Trial

A field trial of the SafetyNET was carried out at a training centre of a local Fire and Rescue Service. The SafetyNET system was deployed inside the training building before a fire was started in a room. Figure 14.10 shows the training building and a SafetyNET wireless sensor node deployed in the training building. A fire tender equipped with a tough book and a ZigBee adaptor was wirelessly connected with the SafetyNET wireless sensor network. The field trial result has shown that the desired environmental data information, fire and fire fighters' locations can be successfully acquired by such a system located outside the building inside which a fire incident is happening. Using the SafetyNET system, command officers can clearly understand the status of the fire incident and the progress of the rescue services being carried out by fire fighters.

14.7 Summary

SafetyNET is a real-time, rapid response system that aims to help fire and rescue service (FRS) related personnel by providing building and hazard information when a fire incident occurs. It provides information includes the monitoring of fire

and rescue personnel located at the emergency scene, the provision of floor-plans and critical environmental data. The innovation of the SafetyNET system, which makes it different from a conventional fire alarm system, is the introduction of a highly flexible wireless sensor network. By integrating low cost and low power environment sensors together with location tracking sensors coupled with battery enabled wireless transmission modules, a properly programmed wireless sensor network can easily be deployed at any required location inside a building, since the requirement for cable has been removed. The employment of advanced ad hoc wireless communication protocol also produces effective coverage. If the main power supply has been cut off, as the result of an emergency, then the wireless sensor network would be the only available system to monitor the internal building environment and the movement of incoming fire fighters. All the information the network can gather could be made available to an information processing system working on the fire tenders that would then be provided to the outside on-site commander, who's in charge of the entire fire rescue operation with comprehensive and up-to-the-minute information.

This chapter overviewed the SafetyNET system, including its information infrastructure, hardware devices, mobile fire tender network and SafetyNET wireless sensor network. All the design was based on the principles the previous chapters of this book introduced.

References

Yang, L.: On-site information sharing for emergency response management. J. Emergency Manage. **l5**(5), 55–64 (2007)

Yang, S.H., Frederick, P.: SafetyNET—a wireless sensor network for fire protection and emergency responses. Meas. Control **39**(7), 218–219 (2006)

Yang, L., Prasanna, R., King, M.: On-site information systems design for emergency first responders. J. Inf. Technol. Theor. Appl. **10**(1), 5–27 (2009)

Chapter 15
Conclusion

Keywords WSN · Remote monitoring · Remote control · Mobile object tracking

15.1 Summary

Wireless sensor networks (WSNs) are one of the emerging topics in the realm of computer science and electronic engineering. The applications of WSNs cover a wide range from natural monitoring to ambient awareness, from military to surveillance. The services offered by a WSN can be obtained through cooperation between wireless sensor nodes, and classified into monitoring, tracking, alerting, and information 'on-demand' (Roman et al. 2007). Sensor nodes can continuously monitor environmental parameters in their surroundings such as temperature in a room. Sensors can track the position of goods, important equipment and people in real-time, identifying a person or object in motion. Sensors can constantly monitor certain physical conditions and automatically alert the users of the system if an abnormal condition occurs. WSNs can serve as data sources and be queried about the actual level of a certain environmental parameter, providing information 'on-demand' (Shorey et al. 2006). These services make these wireless sensors and WSNs very useful for monitoring natural phenomena, detecting environmental changes, controlling security, estimating traffic flows, monitoring military applications, and tracking friendly forces in the battlefields.

The main challenges in the design and implementation of WSNs are the need for self-configuration and self-maintenance, and the extreme resource poverty of their individual sensor nodes in terms of memory and energy, harsh application environments, data processing capability, and life time. These new features make the design methods for traditional networking is not suitable for WSNs because of these extreme and none traditional constraints. This book mainly provides the reader with the fundamental issues and solutions in the design and implementation of WSNs, including individual sensor nodes hardware design, embedded software design, routing algorithms of WSNs, locating sink nodes in WSNs, interference mitigation, data fusions, and security. Furthermore, application technologies such

S.-H. Yang, *Wireless Sensor Networks*, Signals and Communication Technology,
DOI: 10.1007/978-1-4471-5505-8_15,

as indoor location tracking, logistics management, Internet of Things are introduced, and in addition, we present two real applications: SafetyNET and IndeedNET. We do not claim that this book provides all the answers to the challenging issues in the design and implementation of WSNs. Many of them still need to be properly addressed as it is hard to find generic solutions for these challenges. Therefore, application-specific solutions have to be provided.

15.2 Research Opportunities for Future Development

Today, most commercial WSN solutions are battery-driven and based on the IEEE 802.15.4 standard, which defines the PHY and MAC layers for low-power, low-bit rate communication. The biggest obstacle of widely deploying WSNs remains in the extreme power poverty of individual sensor nodes. In many scenarios sensor nodes spend a large part of their time in a sleep node to save energy, which is often unacceptable. Energy harvesting technologies might provide a solution to overcome this biggest obstacle, but the cost is expensive and the size of the device is large, therefore it is not practical for most application as sensor networks may consists of a very large number of nodes. Any breakthrough in the energy harvesting engineering research will certainly enlarge the available market for WSNs by bringing more practical applications of WSNs.

WSNs with RFID act as a bridge between the physical and digital worlds. Integrating WSNs with the Internet enables the global awareness of targeted environments or objects, which is one of the main objectives in the Internet of Things. There are many challenging issues in the area. Chapter 12 gives only a brief introduction to the Internet of Things. IoT is becoming a rich area in the research communities (Atzori et al. 2010; Yang et al. 2013).

In-network data processing for WSNs has been briefly introduced in Chap. 8. It has been recently emphasised in the WSN research communities due to the energy constraints of battery powered sensor nodes (Gaber et al. 2009; He et al. 2013). In-network data processing requires novel data-centric routing mechanisms as well as a reconsideration of traditional network and database interface layering (Govindan et al. 2002).

Finally, the greatest opportunity for WSNs is the large scale actual applications. This book describes our first hand research work and application experience. Any slight improvement in the challenging issues presented in this book will help.

References

Atzori, L., Iera, A., Morabito, G.: The internet of things: a survey. Comput. Netw. **54**, 2787–2805 (2010)

Gaber, M.M., Roehm, U., Herink, K.: An analytical study of central and in-network data processing for wireless sensor networks. Inf. Process. Lett. **110**, 62–70 (2009)

Govindan, R., Hellerstein, J., Hong, W., Madden, S., Franklin, M., and Shenker, S.: The sensor network as a database, Technical Report 02-771, Computer Science Department, University of Southern California (2002)

He, W., Yang, S.H., and Yang, L.: NMCA: Neighbour-aware multiple-path clustering aggregation in wireless sensor networks, IEEE International Conference on Networking, Sensing and Control, Évry, France, 10–12 Apr (2013)

Roman, R., Alcaraz, C., Lopez, J.: The role of wireless sensor networks in the area of critical information infrastructure protection. Inf. Secur. Tech. Rep. **12**(1), 24–31 (2007)

Shorey, R., Ananda, A., Chan, M.C., Ooi, W.T.: Mobile, Wireless, and Sensor Networks. Wiley, Hoboken (2006)

Yang, L., Yang, S.H., and Plotnick, L.: How the internet of things technology enhanced emergency response operations. Technol. Forecast. Soc. Change **80**(9), 1854–1867 (2013)

Index

S.-H. Yang, *Wireless Sensor Networks*, Signals and Communication Technology,
DOI: 10.1007/978-1-4471-5505-8, © Springer-Verlag London 2014