Wireless Sensor Networks

Ian F. Akyildiz

Georgia Institute of Technology, USA

Mehmet Can Vuran

University of Nebraska-Lincoln, USA

WILEY

A John Wiley and Sons, Ltd, Publication

Wireless Sensor Networks

IAN F. AKYILDIZ SERIES IN COMMUNICATIONS AND NETWORKING

The *Ian F. Akyildiz Series in Communications and Networking* offers a comprehensive range of graduate-level text books for use on the major graduate programmes in communications engineering and networking throughout Europe, the USA and Asia. The series provides technically detailed books covering cutting-edge research and new developments in wireless and mobile communications, and networking. Each book in the series contains supporting material for teaching/learning purposes (such as exercises, problems and solutions, objectives and summaries etc), and is accompanied by a website offering further information such as slides, teaching manuals and further reading.

Titles in the series:

Akyildiz and Wang: *Wireless Mesh Networks*, 978-0470-03256-5, January 2009
Akyildiz and Vuran: *Wireless Sensor Networks*, 978-0470-03601-3, August 2010
Akyildiz, Lee and Chowdhury: *Cognitive Radio Networks*, 978-0470-68852-6 (forthcoming, 2011)
Ekici: *Mobile Ad Hoc Networks*, 978-0470-68193-0 (forthcoming, 2011)

Wireless Sensor Networks

Ian F. Akyildiz

Georgia Institute of Technology, USA

Mehmet Can Vuran

University of Nebraska-Lincoln, USA

WILEY

A John Wiley and Sons, Ltd, Publication

This edition first published 2010

Registered office
John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex, PO19 8SQ,
United Kingdom

For details of our global editorial offices, for customer services and for information about how to apply for permission to reuse the copyright material in this book please see our website at www.wiley.com.

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic books.

Designations used by companies to distinguish their products are often claimed as trademarks. All brand names and product names used in this book are trade names, service marks, trademarks or registered trademarks of their respective owners. The publisher is not associated with any product or vendor mentioned in this book. This publication is designed to provide accurate and authoritative information in regard to the subject matter covered. It is sold on the understanding that the publisher is not engaged in rendering professional services. If professional advice or other expert assistance is required, the services of a competent professional should be sought.

Library of Congress Cataloging-in-Publication Data

Akyildiz, Ian Fuat.
Wireless sensor networks / Ian F. Akyildiz, Mehmet Can Vuran.
p. cm.
Includes bibliographical references and index.
ISBN 978-0-470-03601-3 (cloth)
1. Wireless sensor networks. I. Vuran, Mehmet Can. II. Title.
TK7872.D48A38 2010 681'.2–dc22
2010008113

A catalogue record for this book is available from the British Library.

ISBN 978-0-470-03601-3 (H/B)

Set in 9/11pt Times by Sunrise Setting Ltd, Torquay, UK.
Printed and bound in Singapore by Markono Print Media Pte Ltd, Singapore.

To my wife Maria and
children Celine, Rengin
and Corinne for their continous
love and support…

IFA

Hemşerim'e…
To the loving memory of my
Dad, Mehmet Vuran…

MCV

Contents

About the Series Editor

Ian F. Akyildiz is the Ken Byers Distinguished Chair Professor with the School of Electrical and Computer Engineering, Georgia Institute of Technology; Director of Broadband Wireless Networking Laboratory and Chair of the Telecommunications Group. Since June 2008 he has been an Honorary Professor with the School of Electrical Engineering at the Universitat Politècnica de Catalunya, Barcelona, Spain. He is the Editor-in-Chief of *Computer Networks Journal* (Elsevier), is the founding Editor-in-Chief of the *Ad Hoc Networks Journal* (Elsevier) in 2003 and is the founding Editor-in-Chief of the *Physical Communication (PHYCOM) Journal* (Elsevier) in 2008. He is a past editor for *IEEE/ACM Transactions on Networking* (1996–2001), the *Kluwer Journal of Cluster Computing* (1997–2001), the *ACM-Springer Journal for Multimedia Systems* (1995–2002), *IEEE Transactions on Computers* (1992–1996) as well as the *ACM-Springer Journal of Wireless Networks (ACM WINET)* (1995–2005).

Dr Akyildiz was the Technical Program Chair and General Chair of several IEEE and ACM conferences including ACM MobiCom'96, IEEE INFOCOM'98, IEEE ICC'03, ACM MobiCom'02 and ACM SenSys'03, and he serves on the advisory boards of several research centers, journals, conferences and publication companies. He is an IEEE Fellow (1996) and an ACM Fellow (1997), and served as a National Lecturer for ACM from 1989 until 1998. He received the ACM Outstanding Distinguished Lecturer Award for 1994, and has served as an IEEE Distinguished Lecturer for IEEE COMSOC since 2008. Dr Akyildiz has received numerous IEEE and ACM awards including the 1997 IEEE Leonard G. Abraham Prize award (IEEE Communications Society), the 2002 IEEE Harry M. Goode Memorial award (IEEE Computer Society), the 2003 Best Tutorial Paper Award (IEEE Communications Society), the 2003 ACM SIGMOBILE Outstanding Contribution Award, the 2004 Georgia Tech Faculty Research Author Award and the 2005 Distinguished Faculty Achievement Award.

Dr Akyildiz is the author of two advanced textbooks entitled *Wireless Mesh Networks* and *Wireless Sensor Networks*, published by John Wiley & Sons in 2010.

His current research interests are in Cognitive Radio Networks, Wireless Sensor Networks, Wireless Mesh Networks and Nano-networks.

Preface

Wireless sensor networks (WSNs) have attracted a wide range of disciplines where close interactions with the physical world are essential. The distributed sensing capabilities and the ease of deployment provided by a wireless communication paradigm make WSNs an important component of our daily lives. By providing distributed, real-time information from the physical world, WSNs extend the reach of current cyber infrastructures to the physical world.

WSNs consist of tiny sensor nodes, which act as both data generators and network relays. Each node consists of sensor(s), a microprocessor, and a transceiver. Through the wide range of sensors available for tight integration, capturing data from a physical phenomenon becomes standard. Through on-board microprocessors, sensor nodes can be programmed to accomplish complex tasks rather than transmit only what they observe. The transceiver provides wireless connectivity to communicate the observed phenomena of interest. Sensor nodes are generally stationary and are powered by limited capacity batteries. Therefore, although the locations of the nodes do not change, the network topology dynamically changes due to the power management activities of the sensor nodes. To save energy, nodes aggressively switch their transceivers off and essentially become disconnected from the network. In this dynamic environment, it is a major challenge to provide connectivity of the network while minimizing the energy consumption. The energy-efficient operation of WSNs, however, provides significantly long lifetimes that surpass any system that relies on batteries.

In March 2002, our survey paper "Wireless sensor networks: A survey" appeared in the Elsevier journal *Computer Networks*, with a much shorter and concise version appearing in *IEEE Communications Magazine* in August 2002. Over the years, both of these papers were among the top 10 downloaded papers from Elsevier and IEEE Communication Society (ComSoc) journals with over 8000 citations in total.[1] Since then, the research on the unique challenges of WSNs has accelerated significantly. In the last decade, promising results have been obtained through these research activities, which have enabled the development and manufacture of sophisticated products. This, as a result, eventually created a brand-new market powered by the WSN phenomenon. Throughout these years, the deployment of WSNs has become a reality. Consequently, the research community has gained significant experience through these deployments. Furthermore, many researchers are currently engaged in developing solutions that address the unique challenges of the present WSNs and envision new WSNs such as wireless underwater and underground sensor networks. We have contributed to this research over the years through numerous articles and four additional survey/roadmap papers on wireless sensor actor networks, underwater acoustic networks, wireless underground sensor networks, and wireless multimedia sensor networks which were published in different years within the last decade.

In summer 2003, we started to work on our second survey paper on WSNs to revisit the state-of-the-art solutions since the *dawn* of this phenomenon. The large volume of work and the interest in both academia and industry have motivated us to significantly enhance this survey to create this book, which is targeted at teaching graduate students, stimulating them for new research ideas, as well as providing academic and industry professionals with a thorough overview and in-depth understanding of the state-of-the-art in wireless sensor networking and how they can develop new ideas to advance this technology as well as support emerging applications and services. The book provides a comprehensive coverage of

[1] According to Google Scholar as of October 2009.

the present research on WSNs as well as their applications and their improvements in numerous fields. This book covers several major research results including the authors' own contributions as well as all standardization committee decisions in a cohesive and unified form. Due to the sheer amount of work that has been published over the last decade, obviously it is not possible to cover every single solution and any lack thereof is unintentional.

The contents of the book mainly follow the TCP/IP stack starting from the physical layer and covering each protocol layer in detail. Moreover, cross-layer solutions as well as services such as synchronization, localization, and topology control are discussed in detail. Special cases of WSNs are also introduced. Functionalities and existing protocols and algorithms are covered in depth. The aim is to teach the readers what already exists and how these networks can further be improved and advanced by pointing out *grand research challenges* in the final chapter of the book.

Chapter 1 is a comprehensive introduction to WSNs, including sensor platforms and network architectures. Chapter 2 summarizes the existing applications of WSNs ranging from military solutions to home applications. Chapter 3 provides a comprehensive coverage of the characteristics, critical design factors, and constraints of WSNs. Chapter 4 studies the physical layer of WSNs, including physical layer technologies, wireless communication characteristics, and existing standards at the WSN physical layer. Chapter 5 presents various medium access control (MAC) protocols for WSNs, with a special focus on the basic carrier sense multiple access with collision avoidance (CSMA/CA) techniques used extensively at this layer, as well as distinct solutions ranging from CSMA/CA variants, time division multiple access (TDMA)-based MAC, and their hybrid counterparts. Chapter 6 focuses on error control techniques in WSNs as well as their impact on energy-efficient communication. Along with Chapter 5, these two chapters provide a comprehensive evaluation of the link layer in WSNs. Chapter 7 is dedicated to routing protocols for WSNs. The extensive number of solutions at this layer are studied in four main classes: data-centric, hierarchical, geographical, and quality of service (QoS)-based routing protocols. Chapter 8 firstly introduces the challenges of transport layer solutions and then describes the protocols. Chapter 9 introduces the cross-layer interactions between each layer and their impacts on communication performance. Moreover, cross-layer communication approaches are explained in detail. Chapter 10 discusses time synchronization challenges and several approaches that have been designed to address these challenges. Chapter 11 presents the challenges for localization and studies them in three classes: ranging techniques, range-based localization protocols, and range-free localization protocols. Chapter 12 is organized to capture the topology management solutions in WSNs. More specifically, deployment, power control, activity, scheduling, and clustering solutions are explained. Chapter 13 introduces the concept of wireless sensor–actor networks (WSANs) and their characteristics. In particular, the coordination issues between sensors and actors as well as between different actors are highlighted along with suitable solutions. Moreover, the communication issues in WSANs are discussed. Chapter 14 presents wireless multimedia sensor networks (WMSNs) along with their challenges and various architectures. In addition, the existing multimedia sensor network platforms are introduced, and the protocols are described in the various layers following the general structure of the book. Chapter 15 is dedicated to underwater wireless sensor networks (UWSNs) with a major focus on the impacts of the underwater environment. The basics of underwater acoustic propagation are studied and the corresponding solutions at each layer of the protocol stack are summarized. Chapter 16 introduces wireless underground sensor networks (WUSNs) and various applications for these networks. In particular, WUSNs in soil and WUSNs in mines and tunnels are described. The channel properties in both these cases are studied. Furthermore, the existing challenges in the communication layers are described. Finally, Chapter 17 discusses the grand challenges that still exist for the proliferation of WSNs.

It is a major task and challenge to produce a textbook. Although usually the authors carry the major burden, there are several other key people involved in publishing the book. Our foremost thanks go to Birgit Gruber from John Wiley & Sons who initiated the entire idea of producing this book. Tiina Ruonamaa, Sarah Tilley, and Anna Smart at John Wiley & Sons have been incredibly helpful, persistent, and patient. Their assistance, ideas, dedication, and support for the creation of this book will always be greatly appreciated. We also thank several individuals who indirectly or directly contributed to our book. In particular, our sincere thanks go to Özgur B. Akan, Tommaso Melodia, Dario Pompili, Weilian Su, Eylem Ekici, Cagri Gungor, Kaushik R. Chowdhury, Xin Dong, and Agnelo R. Silva for their help.

I (MCV) would like to specifically thank the numerous professors who have inspired me throughout my education in both Bilkent University, Ankara, Turkey and Georgia Institute of Technology, Atlanta, GA. I would like to thank my colleagues and friends at the University of Nebraska–Lincoln and the Department of Computer Science and Engineering for the environment they created during the development of this book. I am especially thankful to my PhD advisor, Professor Ian F. Akyildiz, who introduced me to the challenges of WSNs. I wholeheartedly thank him for his strong guidance, friendship, and trust during my PhD as well as my career thereafter. I would also like to express my deep appreciation to my wife, Demet, for her love, exceptional support, constructive critiques, and her sacrifices that made the creation of this book possible. I am thankful to my mom, Ayla, for the love, support, and encouragement that only a mother can provide. Finally, this book is dedicated to the loving memory of my dad, Mehmet Vuran (or *Hemşerim* as we used to call each other). He was the greatest driving force for the realization of this book as well as many other accomplishments in my life.

I (IFA) would like to specifically thank my wife and children for their support throughout all these years. Without their continuous love, understanding, and tolerance, none of these could have been achieved. Also my past and present PhD students, who became part of my family over the last 25 years, deserve the highest and sincerest thanks for being in my life and letting me enjoy the research to the fullest with them. The feeling of seeing how they developed in their careers over the years is indescribable and one of the best satisfactions in my life. Their research results contributed a great deal to the contents of this book as well.

Ian F. Akyildiz and Mehmet Can Vuran

1

Introduction

With the recent advances in *micro electro-mechanical systems* (MEMS) technology, wireless communications, and digital electronics, the design and development of low-cost, low-power, multifunctional sensor nodes that are small in size and communicate untethered in short distances have become feasible. The ever-increasing capabilities of these tiny sensor nodes, which include sensing, data processing, and communicating, enable the realization of wireless sensor networks (WSNs) based on the collaborative effort of a large number of sensor nodes.

WSNs have a wide range of applications. In accordance with our vision [18], WSNs are slowly becoming an integral part of our lives. Recently, considerable amounts of research efforts have enabled the actual implementation and deployment of sensor networks tailored to the unique requirements of certain sensing and monitoring applications.

In order to realize the existing and potential applications for WSNs, sophisticated and extremely efficient communication protocols are required. WSNs are composed of a large number of sensor nodes, which are densely deployed either inside a physical phenomenon or very close to it. In order to enable reliable and efficient observation and to initiate the right actions, physical features of the phenomenon should be reliably detected/estimated from the collective information provided by the sensor nodes [18]. Moreover, instead of sending the raw data to the nodes responsible for the fusion, sensor nodes use their processing capabilities to locally carry out simple computations and transmit only the required and partially processed data. Hence, these properties of WSNs present unique challenges for the development of communication protocols.

The intrinsic properties of individual sensor nodes pose additional challenges to the communication protocols in terms of energy consumption. As will be explained in the later chapters, WSN applications and communication protocols are mainly tailored to provide high energy efficiency. Sensor nodes carry limited power sources. Therefore, while traditional networks are designed to improve performance metrics such as throughput and delay, WSN protocols focus primarily on power conservation. The deployment of WSNs is another factor that is considered in developing WSN protocols. The position of the sensor nodes need not be engineered or predetermined. This allows random deployment in inaccessible terrains or disaster relief operations. On the other hand, this random deployment requires the development of self-organizing protocols for the communication protocol stack. In addition to the placement of nodes, the density in the network is also exploited in WSN protocols. Due to the short transmission ranges, large numbers of sensor nodes are densely deployed and neighboring nodes may be very close to each other. Hence, multi-hop communication is exploited in communications between nodes since it leads to less power consumption than the traditional single hop communication. Furthermore, the dense deployment coupled with the physical properties of the sensed phenomenon introduce correlation in spatial and temporal domains. As a result, the spatio-temporal correlation-based protocols emerged for improved efficiency in networking wireless sensors.

Wireless Sensor Networks Ian F. Akyildiz and Mehmet Can Vuran

In this book, we present a detailed explanation of existing products, developed protocols, and research on algorithms designed thus far for WSNs. Our aim is to provide a contemporary look at the current state of the art in WSNs and discuss the still-open research issues in this field.

1.1 Sensor Mote Platforms

WSNs are composed of individual embedded systems that are capable of (1) interacting with their environment through various sensors, (2) processing information locally, and (3) communicating this information wirelessly with their neighbors. A sensor node typically consists of three components and can be either an individual board or embedded into a single system:

- **Wireless modules** or **motes** are the key components of the sensor network as they possess the communication capabilities and the programmable memory where the application code resides. A mote usually consists of a microcontroller, transceiver, power source, memory unit, and may contain a few sensors. A wide variety of platforms have been developed in recent years including Mica2 [3], Cricket [2], MicaZ [3], Iris [3], Telos [3], SunSPOT [9], and Imote2 [3].
- **A sensor board** is mounted on the mote and is embedded with multiple types of sensors. The sensor board may also include a prototyping area, which is used to connect additional custom-made sensors. Available sensor boards include the MTS300/400 and MDA100/300 [3] that are used in the Mica family of motes. Alternatively, the sensors can be integrated into the wireless module such as in the Telos or the SunSPOT platform.
- **A programming board**, also known as the gateway board, provides multiple interfaces including Ethernet, WiFi, USB, or serial ports for connecting different motes to an enterprise or industrial network or locally to a PC/laptop. These boards are used either to program the motes or gather data from them. Some examples of programming boards include the MIB510, MIB520, and MIB600 [3]. Particular platforms need to be connected to a programming board to load the application into the programmable memory. They could also be programmed over the radio.

While the particular sensor types vary significantly depending on the application, a limited number of wireless modules have been developed to aid research in WSNs. Table 1.1 captures the major characteristics of popular platforms that were designed over the past few years in terms of their processor speed, programmable and storage memory size, operating frequency, and transmission rate. The timeline for these platforms is also shown in Figure 1.1. As can be observed, the capabilities of these platforms vary significantly. However, in general, the existing platforms can be classified into two based on both their capabilities and the usage. Next, we overview these existing platforms as *low-end* and *high-end* platforms. Moreover, several standardization efforts that have been undertaken for the proliferation of application development will be explained in Section 1.1.3. Finally, the software packages that have been used within these devices are described.

1.1.1 Low-End Platforms

The low-end platforms are characterized by their limited capabilities in terms of processing, memory, and communication. These platforms are usually envisioned to be deployed in large numbers in a WSN to accomplish sensing tasks as well as providing a connectivity infrastructure. The following platforms have been mostly used in developing communication protocols recently:

Mica family: The Mica family of nodes consist of Mica, Mica2, MicaZ, and IRIS nodes and are produced by Crossbow [3]. Each node is equipped with 8-bit Atmel AVR microcontrollers with a speed of 4–16 MHz and 128–256 kB of programmable flash. While the microcontrollers are similar, the Mica family of nodes have been equipped with a wide range of transceivers. The Mica node includes a 916 or 433 MHz transceiver at 40 kbps, while the Mica2 platform is equipped with a 433/868/916 MHz

Table 1.1 Mote hardware.

Mote type	CPU speed (MHz)	Prog. mem. (kB)	RAM (kB)	Radio freq. (MHz)	Tx. rate (kbps)
Berkeley [3]					
WeC	8	8	0.5	916	10
rene	8	8	0.5	916	10
rene2	8	16	1	916	10
dot	8	16	1	916	10
mica	6	128	4	868	10/40
mica2	16	128	4	433/868/916	38.4 kbaud
micaz	16	128	4	2.4 GHz	250
Cricket [3]	16	128	4	433	38.4 kbaud
EyesIFX [17]	8	60	2	868	115
TelosB/Tmote [3]	16	48	10	2.4 GHz	250
SHIMMER [16]	8	48	10	BT/2.4 GHz[a]	250
Sun SPOT [9]	16–60	2 MB	256	2.4 GHz	250
BTnode [1]	8	128	64	BT/433–915[a]	Varies
IRIS [3]	16	128	8	2.4 GHz	250
V-Link [15]	N/A	N/A	N/A	2.4 GHz	250
TEHU-1121 [7]	N/A	N/A	N/A	0.9/2.4 GHz	N/A
NI WSN-3202 [6]	N/A	N/A	N/A	2.4 GHz	250
Imote [3]	12	512	64	2.4 GHz (BT)	100
Imote2 [3]	13–416	32 MB	256	2.4 GHz	250
Stargate [3]	400	32 MB	64 MB SD	2.4 GHz	Varies[b]
Netbridge NB-100 [3]	266	8 MB	32 MB	Varies[b]	Varies[b]

[a] BTnode and SHIMMER motes are equipped with two transceivers: Bluetooth and a low-power radio.

[b] The transmission rate of the Stargate board and the Netbridge depends on the communication device connected to it (MicaZ node, WLAN card, etc.).

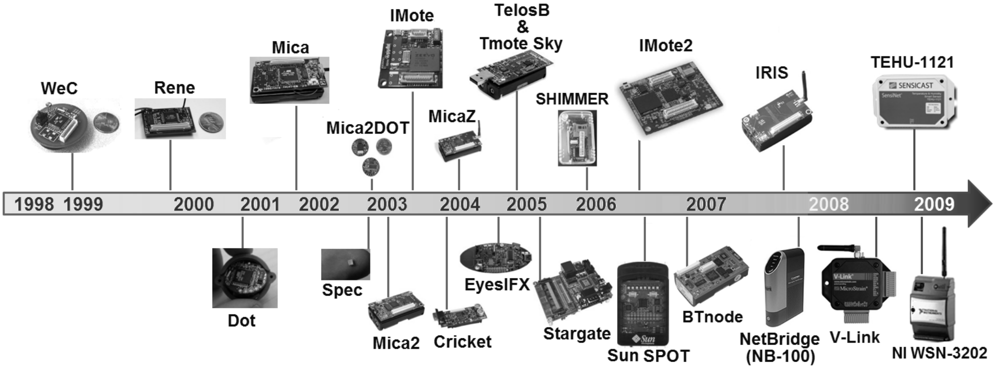


Figure 1.1 Timeline for the sensor mote platforms.

transceiver at 40 kbps. On the other hand, the MicaZ and IRIS nodes are equipped with IEEE 802.15.4 compliant transceivers, which operate at 2.4 GHz with 250 kbps data rate. Each platform has limited memory in terms of RAM (4–8 kB) and data memory (512 kB). Moreover, each version is equipped with a 51-pin connector that is used to connect additional sensor boards and programming boards to the mote.

Telos/Tmote: An architecture similar to the MicaZ platform has been adopted for the Telos motes from Crossbow and Tmote Sky motes from Sentilla (formerly Moteiv). While the transceiver is kept intact, Telos/Tmote motes have larger RAM since an 8 MHz TI MSP430 microcontroller with 10 kB RAM is used. Furthermore, Telos/Tmote platforms are integrated with several sensors including light, IR, humidity, and temperature as well as a USB connector, which eliminates the need for additional sensor or programming boards. Moreover, 6- and 10-pin connectors are included for additional sensors.

EYES: The EYES platform has been designed as a result of a 3-year European project and is similar to the Telos/Tmote architectures. A 16-bit microcontroller with 60 kB of program memory and 2 kB data memory is used in EYES [24]. Moreover, the following sensors are embedded with the mote: compass, accelerometer, and temperature, light, and pressure sensors. The EYES platform includes the TR1001 transceiver, which supports transmission rates up to 115.2 kbps with a power consumption of 14.4 mW, 16.0 mW, and 15.0 μW during receive, transmit, and sleep modes, respectively. The platform also includes an RS232 serial interface for programming.

In addition to these platforms, several low-end platforms have been developed with similar capabilities as listed in Table 1.1 and shown in Figure 1.1. An important trend to note is the appearance of proprietary platforms from the industry such as V-Link, TEHU, and the National Instruments motes in recent years (2008–2009).

The low-end platforms are used for sensing tasks in WSNs and they provide a connectivity infrastructure through multi-hop networking. These nodes are generally equipped with low-power microcontrollers and transceivers to decrease the cost and energy consumption. As a result, they are used in large numbers in the deployment of WSNs. It can be observed that wireless sensor platforms generally employ the *Industrial, Scientific, and Medical* (ISM) bands, which offer license-free communication in most countries. More specifically, most recent platforms include the CC2420 transceiver, which operates in the 2.4 GHz band and is compatible with the IEEE 802.15.4 standard. This standardization provides heterogeneous deployments of WSNs, where various platforms are used in a network. Most of the communication protocols discussed in this book are developed using these platforms.

1.1.2 High-End Platforms

In addition to sensing, local processing, and multi-hop communication, WSNs require additional functionalities that cannot be efficiently carried out by the low-end platforms. High-level tasks such as network management require higher processing power and memory compared to the capabilities of these platforms. Moreover, the integration of WSNs with existing networking infrastructure requires multiple communication techniques to be integrated through *gateway* modules. Furthermore, in networks where processing or storage hubs are integrated with sensor nodes, higher capacity nodes are required. To address these requirements, high-end platforms have been developed for WSNs.

Stargate: The Stargate board [8] is a high-performance processing platform designed for sensing, signal processing, control, and sensor network management. Stargate is based on Intel's PXA-255 Xscale 400 MHz RISC processor, which is the same processor found in many handheld computers including the Compaq IPAQ and the Dell Axim. Stargate has 32 MB of flash memory, 64 MB of SDRAM, and an on-board connector for Crossbow's Mica family motes as well as PCMCIA Bluetooth or IEEE 802.11 cards. Hence, it can work as a wireless gateway and computational hub for in-network processing algorithms.

When connected with a webcam or other capturing device, it can function as a medium-resolution multimedia sensor, although its energy consumption is still high [22].

Stargate NetBridge was developed as a successor to Stargate and is based on the Intel IXP420 XScale processor running at 266 MHz. It features one wired Ethernet and two USB 2.0 ports and is equipped with 8 MB of program flash, 32 MB of RAM, and a 2 GB USB 2.0 system disk, where the Linux operating system is run. Using the USB ports, a sensor node can be connected for gateway functionalities.

Imote and Imote2: Intel has developed two prototypal generations of wireless sensors, known as Imote and Imote2 for high-performance sensing and gateway applications [3]. Imote is built around an integrated wireless microcontroller consisting of an 8-bit 12 MHz ARM7 processor, a Bluetooth radio, 64 kB RAM, and 32 kB flash memory, as well as several I/O options. The software architecture is based on an ARM port of TinyOS.

The second generation of Intel motes, Imote2, is built around a new low-power 32-bit PXA271 XScale processor at 320/416/520 MHz, which enables DSP operations for storage or compression, and an IEEE 802.15.4 ChipCon CC2420 radio. It has large on-board RAM and flash memories (32 MB), additional support for alternate radios, and a variety of high-speed I/O to connect digital sensors or cameras. Its size is also very limited, 48 × 33 mm, and it can run the Linux operating system and Java applications.

1.1.3 Standardization Efforts

The heterogeneity in the available sensor platforms results in compatibility issues for the realization of envisioned applications. Hence, standardization of certain aspects of communication is necessary. To this end, the IEEE 802.15.4 [14] standards body was formed for the specification of low-data-rate wireless transceiver technology with long battery life and very low complexity. Three different bands were chosen for communication, i.e., 2.4 GHz (global), 915 MHz (the Americas), and 868 MHz (Europe). While the PHY layer uses binary phase shift keying (BPSK) in the 868/915 MHz bands and offset quadrature phase shift keying (O-QPSK) in the 2.4 GHz band, the MAC (Medium Access Control) layer provides communication for star, mesh, and cluster tree-based topologies with controllers. The transmission range of the nodes is assumed to be 10–100 m with data rates of 20 to 250 kbps [14]. Most of the recent platforms developed for WSN research comply with the IEEE 802.15.4 standard. Actually, the IEEE 802.15.4 standard, explained in Chapter 4, acquired a broad audience and became the *de facto* standard for PHY and MAC layers in low-power communication. This allows the integration of platforms with different capabilities into the same network.

On top of the IEEE 802.15.4 standard, several standard bodies have been formed to proliferate the development of low-power networks in various areas. It is widely recognized that standards such as Bluetooth and WLAN are not well suited for low-power sensor applications. On the other hand, standardization attempts such as ZigBee, WirelessHART, WINA, and SP100.11a, which specifically address the typical needs of wireless control and monitoring applications, are expected to enable rapid improvement of WSNs in the industry. In addition, standardization efforts such as 6LoWPAN are focused on providing compatibility between WSNs and existing networks such as the Internet.

Next, three major standardization efforts will be described in detail: namely, ZigBee [13], WirelessHART [12], and 6LoWPAN [4]. In addition, other standardization efforts will be summarized.

ZigBee

The ZigBee [13] standard has been developed by the ZigBee Alliance, which is an international, non-profit industrial consortium of leading semiconductor manufacturers and technology providers. The ZigBee standard was created to address the market need for cost-effective, standard-based wireless networking solutions that support low data rates, low power consumption, security, and reliability

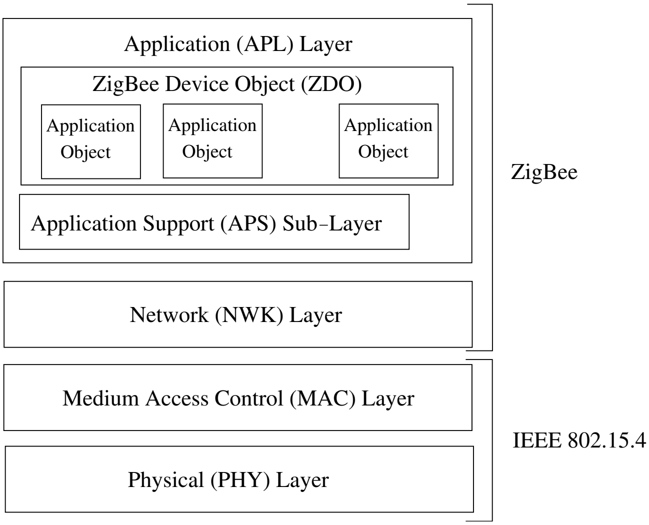


Figure 1.2 IEEE 802.15.4 and the ZigBee protocol stack [13].

through wireless personal area networks (WPANs). Five main application areas are targeted: home automation, smart energy, building automation, telecommunication services, and personal health care.

The ZigBee standard is defined specifically in conjunction with the IEEE 802.15.4 standard. Therefore, both are usually confused. However, as shown in Figure 1.2, each standard defines specific layers of the protocol stack. The PHY and MAC layers are defined by the IEEE 802.15.4 standard while the ZigBee standard defines the network layer (NWK) and the application framework. Application objects are defined by the user. To accommodate a large variety of applications, three types of traffic are defined, Firstly, *periodic data traffic* is required for monitoring applications, where sensors provide continuous information regarding a physical phenomenon The data exchange is controlled through the network controller or a router. Secondly, *Intermittent data traffic* applies to most event-based applications and is triggered through either the application or an external factor. This type of traffic is handled through each router node. To save energy, the devices may operate in disconnected mode, whereas they operate in sleep mode most of the time. Whenever information needs to be transmitted, the transceiver is turned on and the device associates itself with the network. Finally, *repetitive low-latency data traffic* is defined for certain communications such as a mouse click that needs to be completed within a certain time. This type of traffic is accommodated through the polling-based frame structure defined by the IEEE 802.15.4 standard.

The ZigBee network (NWK) layer provides management functionalities for the network operation. The procedures for establishing a new network and the devices to gain or relinquish membership of the network are defined. Furthermore, depending on the network operation, the communication stack of each device can be configured. Since ZigBee devices can be a part of different networks during their lifetime, the standard also defines a flexible addressing mechanism. Accordingly, the network coordinator assigns an address to the devices as they join the network. As a result, the unique ID of each device is not used for communication but a shorter address is assigned to improve the efficiency during communication. In a tree architecture, the address of a device also identifies its parent, which is used for routing purposes. The NWK layer also provides synchronization between devices and network controllers. Finally, multi-hop routes are generated by the NWK layer according to defined protocols.

As shown in Figure 1.2, the ZigBee standard also defines certain components in the application layer. This layer consists of the APS sub-layer, the ZigBee device object (ZDO), and the manufacturer-defined application objects [13]. The applications are implemented through these manufacturer-defined application objects and implementation is based on requirements defined by the standard. The ZDO defines functions provided by the device for network operation. More specifically, the role of devices such as a network coordinator or a router is defined through the ZDO. Moreover, whenever a device needs to be associated with the network, the binding requests are handled through the ZDO. Finally, the APS sub-layer provides discovery capability to devices so that the neighbors of a device and the functionalities provided by these neighbors can be stored. This information is also used to match the binding requests of the neighbors with specific functions.

WirelessHART

WirelessHART [12] has been developed as a wireless extension to the industry standard Highway Addressable Remote Transducer (HART) protocol. HART is the most used communication protocol in the automation and industrial applications that require real-time support with a device count around 20 million [12]. It is based on superimposing a digital FSK-modulated signal on top of the 4–20 mA analog current loop between different components. HART provides a master/slave communication scheme, where up to two masters are accommodated in the network. Accordingly, devices connected to the system can be controlled through a permanent system and handheld devices for monitoring and control purposes.

The WirelessHART standard has been released as a part of the HART 7 specification as the first open wireless communication standard specifically designed for process measurement and control applications [12]. WirelessHART relies on the IEEE 802.15.4 PHY layer standard for the 2.4 GHz band. Moreover, a TDMA-based MAC protocol is defined to provide several messaging modes: one-way publishing of process and control values, spontaneous notification by exception, ad hoc request and response, and auto-segmented block transfers of large data sets.

The network architecture of the WirelessHART standard is shown in Figure 1.3. Accordingly, five types of components are defined: *WirelessHART field devices* (WFDs) are the sensor and control elements that are connected to process or plant equipment. *Gateways* provide interfaces with wireless portions of the network and the wired infrastructure. As a result, host application and the controller can interact with the WFDs. The *network manager* maintains operation of the network by scheduling communication slots for devices, determining routing tables, and monitoring the health of the network. In addition to the three main components, the *WirelessHART adapters* provide backward compatibility by integrating existing HART field devices with the wireless network. Finally, *handhelds* are equipped with on-board transceivers to provide on-site access to the wireless network and interface with the WFDs.

Based on these components, a full protocol stack has been defined by the WirelessHART standard. As explained above, at the PHY layer, the IEEE 802.15.4 standard is employed and a TDMA-based MAC protocol is used at the data link layer. In addition, the network topology is designed as a mesh network and each device can act as a source or a router in the network. This network topology is very similar to what is generally accepted for WSNs.

At the network layer, table-based routing is used so that multiple redundant paths are established during network formation and these paths are continuously verified. Accordingly, even if a communication path between a WFD and a gateway is corrupted, alternate paths are used to provide network reliability greater than 3σ (99.7300204%). In addition to established paths, source routing techniques are used to establish ad hoc communication paths. Moreover, the network layer supports dynamic bandwidth management by assigning allocated bandwidth to certain devices. This is also supported by the underlying TDMA structure by assigning appropriate numbers of slots to these devices. The bandwidth is allocated on a demand basis and can be configured when a device joins the network.

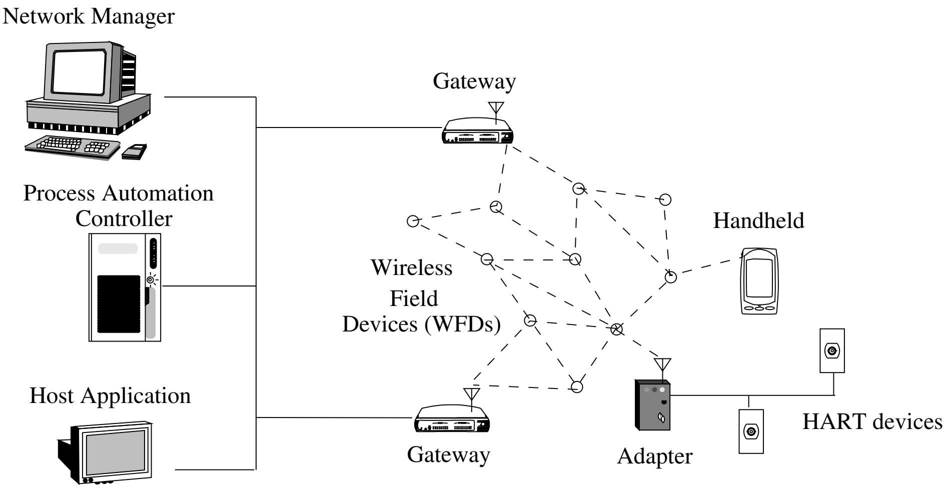


Figure 1.3 WirelessHART architecture and components [12].

The transport layer of the WirelessHART standard provides reliability on the end-to-end path and supports TCP-like reliable block transfers of large data sets. End-to-end monitoring and control of the network are also provided. Accordingly, WFDs continuously broadcast statistics related to their communication success and neighbors, which is monitored by the network manager to establish redundant routes and improve energy efficiency.

Finally, the application layer supports the standard HART application layer, where existing solutions can be implemented seamlessly.

6LoWPAN

The existing standards enable application-specific solutions to be developed for WSNs. Accordingly, stand-alone networks of sensors can be implemented for specific applications. However, these networks cannot be easily integrated with the Internet since the protocols based on IEEE 802.15.4 are not compliant with the IP. Therefore, sensors cannot easily communicate with web-based devices, servers, or browsers. Instead, gateways are required to collect the information from the WSN and communicate with the Internet. This creates single-point-of-failure problems at the gateways and stresses the neighbors of the gateway.

To integrate WSNs with the Internet, the Internet Engineering Task Force (IETF) is developing the IPv6 over Low-power Wireless Personal Area Network (6LoWPAN) standard [4]. This standard defines the implementation of the IPv6 stack on top of IEEE 802.15.4 to let any device be accessible from and with the Internet.

The basic challenge in integrating IPv6 and WSNs is the addressing structure of IPv6, which defines a header and address information field of 40 bytes. However, IEEE 802.15.4 allows up to 127 bytes for the *whole* packet including header and payload information. Accordingly, straightforward integration of both standards is not efficient. Instead, 6LoWPAN adds an adaptation layer that lets the radio stack and IPv6 communications operate together. A *stacked header* structure has been proposed for the 6LoWPAN standard [23], where, instead of a single monolithic header, four types of headers are utilized according to the type of packet being sent. In addition, stateless compression techniques are used to decrease the size of the header from 40 bytes to around 4 bytes, which is suitable for WSNs.

The four header types are as follows:

- **Dispatch header (1 byte):** This header type defines the type of header following it. The first 2 bits are set to 01 for the dispatch header and the remaining 6 bits define the type of header following it (uncompressed IPv6 header or a header compression header).
- **Mesh header (4 bytes):** This header is identified by 10 in the first 2 bits and is used in mesh topologies for routing purposes. The first 2 bits are followed by additional 2 bits that indicate whether the source and destination addresses are 16-bit short or 64-bit long addresses. A 4-bit hop left field is used to indicate the number of hops left. Originally, 15 hops are supported but an extra byte can be used to support 255 hops. Finally, the remaining fields indicate the source and the destination addresses of the packet. This information can be used by the routing protocols to find the next hop.
- **Fragmentation header (4–5 bytes):** IPv6 can support payloads up to 1280 bytes whereas this is 102 bytes for IEEE 802.15.4. This is solved by fragmenting larger payloads into several packets and the fragmentation header is used to fragment and reassemble these packets. The first fragment includes a header of 4 bytes, which is indicated by 11 in the first 2 bits and by 000 in the next 3 bits. This is followed by the datagram size and datagram tag fields. The following fragment header uses 11100 in the first 5 bits followed by the datagram size, tag, and the datagram offset.
- **Header compression header (1 byte):** Finally, the 40-byte IPv6 header is compressed into 2 bytes including the header compression header. This compression exploits the fact that IEEE 802.15.4 packet headers already include the MAC addresses of the source and destination pairs. These MAC addresses can be mapped to the lowest 64 bits of an IPv6 address. As a result, the source and destination addresses are completely eliminated from the IPv6 header. Similar techniques are used to eliminate the unnecessary fields for each communication and allow these fields to be inserted when the packet reaches a gateway to the Internet.

Header compression is not the only challenge for WSN–Internet integration. The ongoing efforts in the development of the 6LoWPAN standard aim to address some of these challenges including routing and transport control to provide seamless interoperation of WSNs and the Internet.

Other Standardization Efforts

In addition to the above efforts, several additional platforms have been engaged with defining standards for WSN applications. The ISA SP100.11a standard [5] also centers around the process and factory automation and is being developed by the Systems and Automation Society (ISA). Moreover, the Wireless Industrial Networking Alliance (WINA) [11] was formed in 2003 to stimulate the development and promote the adoption of wireless networking technologies and practices to help increase industrial efficiency. As a first step, this ad hoc group of suppliers and end-users is working to define end-user needs and priorities for industrial wireless systems. The standardization attempts such as ZigBee, WirelessHART, WINA, and SP100.11a, which specifically address the typical needs of wireless control and monitoring applications, are expected to enable rapid improvement of WSNs in the industry.

WSN applications have gained significant momentum during the past decade with the acceleration in research in this field. Although existing applications provide a wide variety of possibilities where the WSN phenomenon can be exploited, there exists many areas waiting for WSN empowerment. Moreover, further enhancements in WSN protocols will open up new areas of applications. Nevertheless, commercialization of these potential applications is still a major challenge.

1.1.4 Software

In addition to hardware platforms and standards, several software platforms have also been developed specifically for WSNs. Among these, the most accepted platform is the TinyOS [10], which is an open-source operating system designed for wireless embedded sensor networks. TinyOS incorporates a component-based architecture, which minimizes the code size and provides a flexible platform for implementing new communication protocols. Its component library includes network protocols, distributed services, sensor drivers, and data acquisition tools, which can be further modified or improved based on the specific application requirements. TinyOS is based on an event-driven execution model that enables fine-grained power management strategies.

Most of the existing software code for communication protocols today is written for the TinyOS platform. Coupled with TinyOS, a TinyOS mote simulator, TOSSIM, has been introduced to simplify the development of sensor network protocols and applications [21]. TOSSIM provides a scalable simulation environment and compiles directly from the TinyOS code. It simulates the TinyOS network stack at the bit level, allowing experimentation with low-level protocols in addition to top-level application systems. It also provides a graphical user interface tool, TinyViz, in order to visualize and interact with running simulations.

In addition to TinyOS, several software platforms and operating systems have been introduced recently. LiteOS [19] is a multi-threading operating system that provides Unix-like abstractions. Compared to TinyOS, LiteOS provides multi-threaded operation, dynamic memory management, and command-line shell support. The shell support, LiteShell, provides a command-line interface at the user side, i.e., the PC, to provide interaction with the sensor node to be programmed.

Contiki [20] is an open-source, multitasking operating system developed for use on a variety of platforms including microcontrollers such as the TI MSP430 and the Atmel AVR, which are used in the Telos, Tmote, and Mica families. Contiki has been built around an event-driven kernel but it is possible to employ preemptive multithreading for certain programs as well as dynamic loading and replacement of individual programs and services. As a result, compared to TinyOS, which is statically linked at compile-time, Contiki allows programs and drivers to be replaced during run-time and without relinking. Moreover, TCP/IP support is also provided through the μIP stack.

The recent SunSPOT platform [9] does not use an operating system but runs a Java virtual machine (VM), Squawk, on the bare metal, which is a fully capable Java ME implementation. The VM executes directly out of flash memory.

While several operating systems with additional capabilities have become available, TinyOS is still being widely used in WSN research. One of the main reasons for this popularity is the vast code space built throughout the development of WSN solutions. Clearly, it is hard to port existing applications and communication protocols to these new operating systems. This calls for platforms that support interoperability for existing code space so that additional flexibility and capabilities are provided to both the research community and industry.

1.2 WSN Architecture and Protocol Stack

The sensor nodes are usually scattered in a *sensor field* as shown in Figure 1.4. Each of these scattered sensor nodes has the capability to collect data and route data back to the *sink/gateway* and the end-users. Data are routed back to the end-user by a multi-hop infrastructureless architecture through the sink as shown in Figure 1.4. The sink may communicate with the *task manager/end-user* via the Internet or satellite or any type of wireless network (like WiFi, mesh networks, cellular systems, WiMAX, etc.), or without any of these networks where the sink can be directly connected to the end-users. Note that there may be multiple sinks/gateways and multiple end-users in the architecture shown in Figure 1.4.

In WSNs, the sensor nodes have the dual functionality of being both data originators and data routers. Hence, communication is performed for two reasons:

- **Source function:** Source nodes with event information perform communication functionalities in order to transmit their packets to the sink.
- **Router function:** Sensor nodes also participate in forwarding the packets received from other nodes to the next destination in the multi-hop path to the sink.

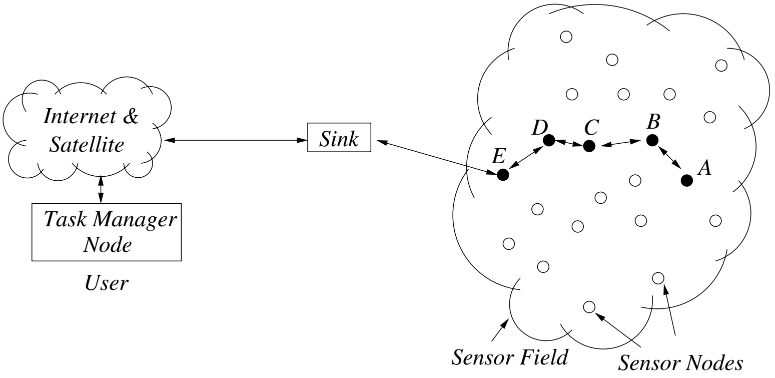


Figure 1.4 Sensor nodes scattered in a sensor field.

The protocol stack used by the sink and all sensor nodes is given in Figure 1.5. This protocol stack combines power and routing awareness, integrates data with networking protocols, communicates power efficiently through the wireless medium, and promotes cooperative efforts of sensor nodes. The protocol stack consists of the *physical layer*, *data link layer*, *network layer*, *transport layer*, *application layer*, as well as *synchronization plane*, *localization plane*, *topology management plane*, *power management plane*, *mobility management plane*, and *task management plane*. The physical layer addresses the needs of simple but robust modulation, transmission, and receiving techniques. Since the environment is noisy and sensor nodes can be mobile, the link layer is responsible for ensuring reliable communication through error control techniques and manage channel access through the MAC to minimize collision with neighbors' broadcasts. Depending on the sensing tasks, different types of application software can be built and used on the application layer. The network layer takes care of routing the data supplied by the transport layer. The transport layer helps to maintain the flow of data if the sensor network application requires it. In addition, the power, mobility, and task management planes monitor the power, movement, and task distribution among the sensor nodes. These planes help the sensor nodes coordinate the sensing task and lower the overall power consumption.

The power management plane manages how a sensor node uses its power. For example, the sensor node may turn off its receiver after receiving a message from one of its neighbors. This is to avoid getting duplicated messages. Also, when the power level of the sensor node is low, the sensor node broadcasts to its neighbors that it is low in power and cannot participate in routing messages. The remaining power is reserved for sensing. The mobility management plane detects and registers the movement of sensor nodes, so a route back to the user is always maintained, and the sensor nodes can keep track of their neighbors. By knowing these neighbor sensor nodes, the sensor nodes can balance their power and task usage. The task management plane balances and schedules the sensing tasks given to a specific region. Not all sensor nodes in that region are required to perform the sensing task at the same time. As a result, some sensor nodes perform the task more than others, depending on their power level. These management planes are needed so that sensor nodes can work together in a power-efficient way, route data in a mobile sensor network, and share resources between sensor nodes. Without them, each sensor node will just work individually. From the standpoint of the whole sensor network, it is more efficient if sensor nodes can collaborate with each other, so the lifetime of the sensor networks can be prolonged.

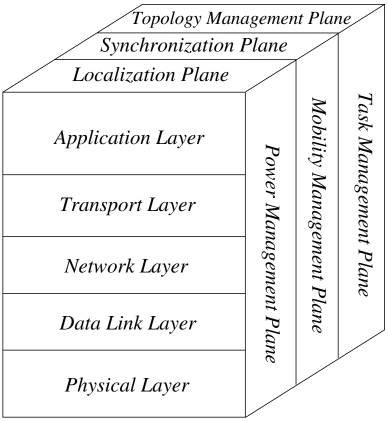


Figure 1.5 The sensor network protocol stack.

1.2.1 Physical Layer

The physical layer is responsible for frequency selection, carrier frequency generation, signal detection, modulation, and data encryption. Frequency generation and signal detection have more to do with the underlying hardware and transceiver design and hence are beyond the scope of our book. More specifically, we focus on signal propagation effects, power efficiency, and modulation schemes for sensor networks.

1.2.2 Data Link Layer

The data link layer is responsible for the multiplexing of data streams, data frame detection, and medium access and error control. It ensures reliable point-to-point and point-to-multipoint connections in a communication network. More specifically, we discuss the medium access and error control strategies for sensor networks.

MAC

The MAC protocol in a wireless multi-hop self-organizing sensor network must achieve two goals. The first goal is creation of the network infrastructure. Since thousands of sensor nodes can be densely scattered in a sensor field, the MAC scheme must establish communication links for data transfer. This forms the basic infrastructure needed for hop-by-hop wireless communication and provides the self-organizing capability. The second objective is to fairly and efficiently share communication resources between sensor nodes. These resources include time, energy, and frequency. Several MAC protocols have been developed for WSNs to address these requirements over the last decade.

Regardless of the medium access scheme, energy efficiency is of utmost importance. A MAC protocol must certainly support the operation of power saving modes for the sensor node. The most obvious means of power conservation is to turn the transceiver off when it is not required. Though this power saving method seemingly provides significant energy gains, it may hamper the connectivity of the network. Once a transceiver is turned off, the sensor node cannot receive any packets from its neighbors, essentially becoming disconnected from the network. Moreover, turning a radio on and off has an overhead in terms of energy consumption due to the startup and shutdown procedures required for both hardware and software. In fact, if the radio is blindly turned off during each idling slot, over a period of time the sensor may end up expending more energy than if the radio had been left on. As a result, operation in a power saving mode is energy efficient only if the time spent in that mode is greater than a certain threshold. There can be a number of such useful modes of operation for the wireless sensor node, depending on the number of states of the microprocessor, memory, A/D converter, and the transceiver. Each of these modes can be characterized by its power consumption and the latency overhead, which is the transition power to and from that mode.

Error Control

Another important function of the data link layer is the error control of transmission data. Two important modes of error control in communication networks are forward error correction (FEC) and automatic repeat request (ARQ), and hybrid ARQ. The usefulness of ARQ in sensor network applications is limited by the additional retransmission cost and overhead. On the other hand, decoding complexity is greater in FEC, as error correction capabilities need to be built in. Consequently, simple error control codes with low-complexity encoding and decoding might present the best solutions for sensor networks. In the design of such a scheme, it is important to have a good knowledge of the channel characteristics and implementation techniques.

1.2.3 Network Layer

Sensor nodes are scattered densely in a field either close to or inside the phenomenon as shown in Figure 1.4. The information collected relating to the phenomenon should be transmitted to the sink, which may be located far from the sensor field. However, the limited communication range of the sensor nodes prevents direct communication between each sensor node and the sink node. This requires efficient multi-hop wireless routing protocols between the sensor nodes and the sink node using intermediate sensor nodes as relays. The existing routing techniques, which have been developed for wireless ad hoc networks, do not usually fit the requirements of the sensor networks. The networking layer of sensor networks is usually designed according to the following principles:

- Power efficiency is always an important consideration.
- Sensor networks are mostly data-centric.
- In addition to routing, relay nodes can aggregate the data from multiple neighbors through local processing.
- Due to the large number of nodes in a WSN, unique IDs for each node may not be provided and the nodes may need to be addressed based on their data or location.

An important issue for routing in WSNs is that routing may be based on data-centric queries. Based on the information requested by the user, the routing protocol should address different nodes that would provide the requested information. More specifically, the users are more interested in querying an attribute of the phenomenon rather than querying an individual node. For instance, “the areas where the temperature is over 70 °F (21 °C)” is a more common query than “the temperature read by node #47.”

One other important function of the network layer is to provide internetworking with external networks such as other sensor networks, command and control systems, and the Internet. In one scenario, the sink nodes can be used as a gateway to other networks, while another scenario is to create a backbone by connecting sink nodes together and making this backbone access other networks via a gateway.

1.2.4 Transport Layer

The transport layer is especially needed when the network is planned to be accessed through the Internet or other external networks. TCP, with its current transmission window mechanisms, does not address the unique challenges posed by the WSN environment. Unlike protocols such as TCP, the end-to-end communication schemes in sensor networks are not based on global addressing. These schemes must consider that addressing based on data or location is used to indicate the destinations of the data packets. Factors such as power consumption and scalability, and characteristics like data-centric routing, mean sensor networks need different handling in the transport layer. Thus, these requirements stress the need for new types of transport layer protocols.

The development of transport layer protocols is a challenging task because the sensor nodes are influenced by hardware constraints such as limited power and memory. As a result, each sensor node cannot store large amounts of data like a server in the Internet, and acknowledgments are too costly for sensor networks. Therefore, new schemes that split the end-to-end communication probably at the sinks may be needed where UDP-type protocols are used in the sensor network.

For communication inside a WSN, transport layer protocols are required for two main functionalities: reliability and congestion control. Limited resources and high energy costs prevent end-to-end reliability mechanisms from being employed in WSNs. Instead, localized reliability mechanisms are necessary. Moreover, congestion that may occur because of the high traffic during events should be mitigated by the transport layer protocols. Since sensor nodes are limited in terms of processing, storage, and energy consumption, transport layer protocols aim to exploit the collaborative capabilities of these sensor nodes and shift the intelligence to the sink rather than the sensor nodes.

1.2.5 Application Layer

The application layer includes the main application as well as several management functionalities. In addition to the application code that is specific for each application, query processing and network management functionalities also reside at this layer.

The layered architecture stack has been initially adopted in the development of WSNs due to its success with the Internet. However, the large-scale implementations of WSN applications reveal that the wireless channel has significant impact on the higher layer protocols. Moreover, resource constraints and the application-specific nature of the WSN paradigm leads to *cross-layer* solutions that tightly integrate the layered protocol stack. By removing the boundaries between layers as well as the associated interfaces, increased efficiency in code space and operating overhead can be achieved.

In addition to the communication functionalities in the layered stack, WSNs have also been equipped with several functionalities that aid the operation of the proposed solutions. In a WSN, each sensor device is equipped with its own local clock for internal operations. Each event that is related to operation of the sensor device including sensing, processing, and communication is associated with timing information controlled through the local clock. Since users are interested in the collaborative information from multiple sensors, timing information associated with the data at each sensor device needs to be consistent. Moreover, the WSN should be able to correctly order the events sensed by distributed sensors to accurately model the physical environment. These timing requirements have led to the development of *time synchronization* protocols in WSNs.

The close interaction with physical phenomena requires location information to be associated in addition to time. WSNs are closely associated with physical phenomena in their surroundings. The gathered information needs to be associated with the location of the sensor nodes to provide an accurate view of the observed sensor field. Moreover, WSNs may be used for tracking certain objects for monitoring applications, which also requires location information to be incorporated into the tracking algorithms. Further, location-based services and communication protocols require position information. Hence, *localization* protocols have been incorporated into the communication stack.

Finally, several *topology management* solutions are required to maintain the connectivity and coverage of the WSN. The topology management algorithms provide efficient methods for network deployment that result in longer lifetime and efficient information coverage. Moreover, topology control protocols help determine the transmit power levels as well as the activity durations of sensor nodes to minimize energy consumption while still ensuring network connectivity. Finally, clustering protocols are used to organize the network into clusters to improve scalability and improve network lifetime.

The integration of each of the components for efficient operation depends on the applications running on the WSN. This application-dependent nature of the WSNs defines several unique properties compared to traditional networking solutions. Although the initial research and deployment of WSNs have focused on data transfer in wireless settings, several novel application areas of WSNs have also emerged. These include *wireless sensor and actor networks*, which consist of actuators in addition to sensors that convert sensed information into actions to act on the environment, and *wireless multimedia sensor networks*, which support multimedia traffic in terms of visual and audio information in addition to scalar data. Furthermore, recently the WSN phenomenon has been adopted in constrained environments such as underwater and underground settings to create *wireless underwater sensor networks* and *wireless underground sensor networks*. These new fields of study pose additional challenges that have not been considered by the vast number of solutions developed for *traditional* WSNs.

The flexibility, fault tolerance, high sensing fidelity, low cost, and rapid deployment characteristics of sensor networks create many new and exciting application areas for remote sensing. In the future, this wide range of application areas will make sensor networks an integral part of our lives. However, realization of sensor networks needs to satisfy the constraints introduced by factors such as fault tolerance, scalability, cost, hardware, topology change, environment, and power consumption. Since these constraints are highly stringent and specific for sensor networks, new wireless ad hoc networking techniques are required. Many researchers are currently engaged in developing the technologies needed for different layers of the sensor network protocol stack. Commercial viability of WSNs has also been shown in several fields. Along with the current developments, we encourage more insight into the problems and more development of solutions to the open research issues as described in this book.

References

[1] BTnode platform. http://www.btnode.ethz.ch/.

[2] Cricket indoor location system. http://nms.lcs.mit.edu/projects/cricket.

[3] Crossbow technology. http://www.xbow.com.

[4] IPv6 over low power WPAN working group. http://tools.ietf.org/wg/6lowpan/.

[5] ISA SP100.11a working group. http://www.isa.org/ISA100.

[6] National Instruments ni wsn-3202 wireless sensor. http://sine.ni.com/nips/cds/view/p/lang/en/nid/206921.

[7] Sensicast TEHU-1121 wireless sensor. http://www.sensicast.com/wireless_sensors.php.

[8] The Stargate platform. http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/Stargate_Datasheet.pdf.

[9] SunSPOT mote specifications. http://www.sunspotworld.com.

[10] TinyOS. Available at http://www.tinyos.net/.

[11] Wireless Industrial Networking Alliance (WINA). http://www.wina.org.

[12] WirelessHART communications protocol. http://www.hartcomm2.org/index.html.

[13] ZigBee Alliance. http://www.zigbee.org/.

[14] IEEE standard for information technology – telecommunications and information exchange between systems – local and metropolitan area networks – specific requirement part 15.4: wireless medium access control (MAC) and physical layer (PHY) specifications for low-rate wireless personal area networks (WPANS). *IEEE Std. 802.15.4a-2007 (Amendment to IEEE Std. 802.15.4-2006)*, pp. 1–203, 2007.

[15] Microstrain V-Link wireless sensor. http://www.microstrain.com/v-link.aspx, 2009.

[16] Shimmer project. http://www.eecs.harvard.edu/ konrad/projects/shimmer/, 2009.

[17] EYES project. http://www.eyes.eu.org/, March 2002–February 2005.

[18] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks*, 38(4):393–422, March 2002.

[19] Q. Cao, T. Abdelzaher, J. Stankovic, and T. He. The LiteOS operating system: towards Unix-like abstractions for wireless sensor networks. In *Proceedings of IPSN'08*, pp. 233–244, St. Louis, Missouri, USA, April 2008.

[20] A. Dunkels, B. Grönvall, and T. Voigt. Contiki – a lightweight and flexible operating system for tiny networked sensors. In *Proceedings of IEEE EmNets'04*, Tampa, Florida, USA, November 2004.

[21] P. Levis, N. Lee, M. Welsh, and D. Culler. Tossim: accurate and scalable simulation of entire TinyOS applications. In *Proceedings of ACM SenSys'03*, pp. 126–137, Los Angeles, CA, USA, November 2003.

[22] C. B. Margi, V. Petkov, K. Obraczka, and R. Manduchi. Characterizing energy consumption in a visual sensor network testbed. In *Proceedings of the IEEE/Create-Net International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities (TridentCom'06)*, Barcelona, Spain, March 2006.

[23] G. Mulligan and the 6LoWPAN Working Group. The 6LoWPAN architecture. In *Proceedings of EmNets'07*, pp. 78–82, Cork, Ireland, June 2007.

[24] L. F. W. van Hoesel, S. O. Dulman, P. J. M. Havinga, and H. J. Kip. Design of a low-power testbed for wireless sensor networks and verification. Technical report, University of Twente, 2003.

2

WSN Applications

The emergence of the WSN paradigm has triggered extensive research on many aspects of it. However, the applicability of sensor networks has long been discussed with emphasis on potential applications that can be realized using WSNs. In this chapter, we present an overview of the existing commercial and academic applications developed for WSNs.

WSNs may consist of many different types of sensors including seismic, magnetic, thermal, visual, infrared, acoustic, and radar, which are able to monitor a wide variety of ambient conditions that include the following [15, 26]: temperature, humidity, pressure, speed, direction, movement, light, soil makeup, noise levels, the presence or absence of certain kinds of objects, and mechanical stress levels on attached objects. As a result, a wide range of applications are possible. This spectrum of applications includes homeland security, monitoring of space assets for potential and human-made threats in space, ground-based monitoring of both land and water, intelligence gathering for defense, environmental monitoring, urban warfare, weather and climate analysis and prediction, battlefield monitoring and surveillance, exploration of the Solar System and beyond, monitoring of seismic acceleration, strain, temperature, wind speed and GPS data.

These ever-increasing applications of WSNs can be mainly categorized into five categories. In this chapter, several examples are described for military (Section 2.1), environmental (Section 2.2), health (Section 2.3), home (Section 2.4), and industry (Section 2.5) applications as illustrated in Figure 2.1.

2.1 Military Applications

WSNs can be an integral part of military *command, control, communications, computing, intelligence, surveillance, reconnaissance, and targeting* (C4ISRT) systems. The rapid deployment, self-organization, and fault tolerance characteristics of sensor networks make them a very promising sensing technique for military C4ISRT. Since sensor networks are based on the dense deployment of disposable and low-cost sensor nodes, destruction of some nodes by hostile action does not affect a military operation as much as the destruction of a traditional sensor, which makes the sensor network concept a better approach for battlefields. Some of the military applications of sensor networks are monitoring friendly forces, equipment, and ammunition; battlefield surveillance; reconnaissance of opposing forces and terrain; targeting; battle damage assessment; and nuclear, biological, and chemical (NBC) attack detection and reconnaissance [5, 9, 15, 29, 37]. Three of the representative military applications are discussed next.

2.1.1 Smart Dust

Smart Dust was one of the earliest applications of the WSN phenomenon. The main goal of this DARPA-funded project is to provide technologies for sensor networks that will be used for military operations in hostile environments. The use of WSNs extend the reach by deploying these networks in areas where

Wireless Sensor Networks Ian F. Akyildiz and Mehmet Can Vuran

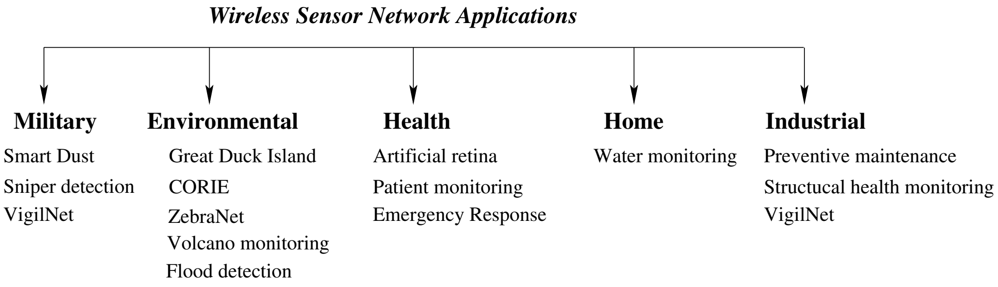


Figure 2.1 Main categories of WSN applications and the specific examples discussed in this chapter.

it is too dangerous for humans to operate continuously. WSNs obtain information needed to assess critical situations by dropping a robust, self-configuring, self-organizing WSN onto the battlefield. The military applications include collecting information from enemy movements, hazardous chemicals, and infrastructure stability. The main goal of the Smart Dust project has been developing sensor platforms in cubic millimeter packages. Instead of focusing on a particular sensor, the transceiver and microcontroller unit (MCU) design are performed.

The resulting sensor node is a 100 cubic millimeter prototype with two chips: a MEMS corner cube optical transmitter array and a CMOS ASIC with an optical receiver, charge pump, and simple digital controller. The Smart Dust project has resulted in the creation of Dust Networks Inc. [9], to commercialize this technology. Accordingly, the Smart Dust platform has been utilized in military applications such as battlefield surveillance, treaty monitoring, transportation monitoring, and scud hunting. In addition, other commercial applications are also viable with this technology. As an example, a virtual keyboard is developed, where each dust mote is attached to each fingernail and the acceleration information is used to detect movements and convert them into key strokes. Similarly, the Smart Dust node is used as a part of inventory control, where each item is equipped with a unique wireless sensor node that can communicate with the palette, truck, warehouse, and the Internet. Finally, Smart Dust nodes are used in product quality monitoring, where the temperature and humidity of meat, produce, and dairy products are monitored.

2.1.2 Sniper Detection System

The Boomerang sniper detection system [5] has been developed for accurate sniper location detection by pinpointing small-arms fire from the shooter. It has been used by the military, law-enforcement agencies, and municipalities. This application has been realized through two distinct architectures.

The countersniper system shown in Figure 2.2 consists of an array of microphones and can be mounted on a vehicle or worn by a soldier [5]. The system uses passive acoustic sensors to detect incoming fire. The detected audio from the microphones is processed to estimate the relative position of the shooter. The Boomerang system is helpful in urban settings, where a soldier or a vehicle can be subjected to fire from any location. The vehicle-mounted sensors can detect the shooter even while moving.

This centralized concept has been enhanced through a distributed network of acoustic sensors for a countersniper system as shown in Figure 2.3 to combat the effects of multi-path in sound detection [37]. The network consists of Mica2 nodes equipped with a custom-made sensor board. The sensor board is embedded with a high-power DSP to provide real-time detection, classification, and correlation of acoustic events. Once a shot has been detected by several sensors, the time and location information of these sensors is used to determine the trajectory of the bullet and estimate the location of the shooter. As a result, the distributed system improves the accuracy of the centralized system. This system is suitable for law-enforcement agencies and municipalities to provide protection during events such as speeches.

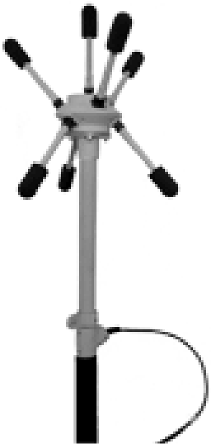


Figure 2.2 Boomerang sniper detection system [5].

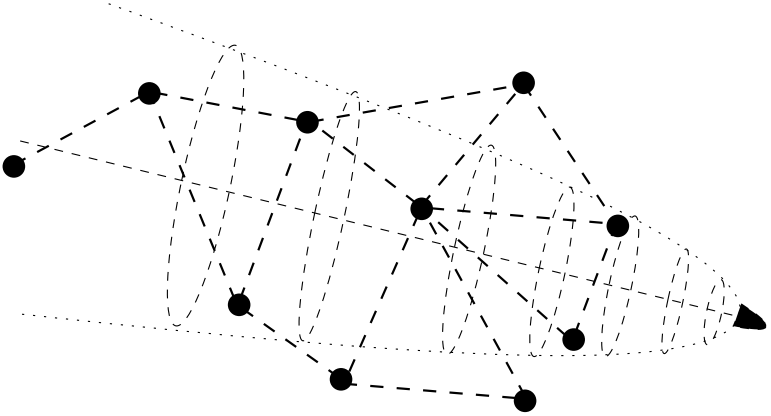


Figure 2.3 Distributed sniper detection system [37].

2.1.3 VigilNet

VigilNet [29] is a large-scale surveillance network designed for energy-efficient and stealthy target tracking in harsh environments. The design and implementation of the network were based on 70 Mica2 nodes equipped with magnetic sensors that detect the magnetic field generated by the movement of vehicles and magnetic objects. The main goal of this application is to provide energy-efficient surveillance support through distributed sensor nodes.

The lifetime of VigilNet is improved through a hierarchical structure of nodes in the network. Certain nodes denoted as *sentries* are responsible for maintaining the multi-hop routes and managing event-based operation of the *non-sentry* nodes, which remain in a low-power state until an event occurs. In case of an event, e.g., a vehicle passing a group of nodes, the network is reorganized into clusters including

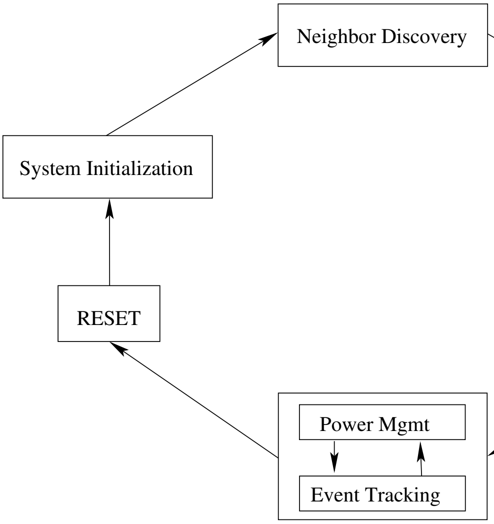


Figure 2.4 Operation phases of VigilNet [29].

the associated group of nodes with the help of sentries. This scheme provides event-based collaborative tracking with a long network lifetime.

The network operation of VigilNet is organized into several phases that are executed in a cycle as shown in Figure 2.4. The five phases are as follows:

1. *System initialization:* This phase includes synchronization, multi-hop backbone setup, and reconfiguration. Network-wide synchronization is essential for the accuracy of the tracking application. This is accomplished by broadcast messages from the sink node sent during the initialization phase. During the propagation of the broadcast messages, a backbone of sentries is also formed throughout the network. This backbone provides a stable path between the nodes and the sink and allows the nodes that are not part of the backbone to sleep. To avoid the diverse effects of asymmetric links between any pair of nodes, each node performs *link symmetry detection* (LSD) to filter out its neighbors with highly symmetric links. The backbone is then built using these links. The time synchronization packets are also used to reconfigure the network on demand. Accordingly, the operating parameters such as cycle length, duty cycle, sampling rate, and aggregation parameters are piggybacked on the synchronization packets. Finally, localization is performed manually during this phase. During deployment, an operator equipped with a GPS unit and a transceiver programs each node with its location based on the GPS reading on site. Apart from localization, each initialization service is performed at the beginning of each cycle.
2. *Neighbor discovery:* The initialization phase is followed by local broadcast of HELLO messages by each node. This message includes information about the node including its ID, whether it is a sentry or not, the number of sentries it is connected to, and its location. This information is used for local message exchanges and sentry selection.
3. *Sentry selection:* The sentries remain active during the network operation and are selected at the beginning of each cycle. Each node determines its duty as a sentry if it is part of the backbone or none of its neighbors is a sentry. This duty is communicated to other nodes using a broadcast message after a certain broadcast duration. Consequently, each node is guaranteed to be covered by at least one sentry.

4. *Status report:* This phase is used by each node to communicate its status to the base station using the backbone. The status report phase can be omitted, but is used for visualization and debugging purposes.
5. *Power management and event tracking:* The final phase of the operating cycle constitutes the majority of the lifetime of the network. Throughout the hierarchical structure, which consists of sentry and non-sentry nodes, a significant amount of energy is saved. The sleep and wakeup cycles of the non-sentry nodes are controlled by the sentry nodes through proactive or reactive control schemes. For proactive control, each sentry node broadcasts sleep beacons that let each non-sentry node switch to a sleep state. In case of an event, the sleep beacons are not sent and the non-sentry nodes participate in delivering information. On the other hand, for reactive control, the sentry nodes send beacons only when an event occurs to wake up the non-sentry nodes, which, otherwise, follow a sleep/wakeup schedule. When an event occurs, the nodes that sense the event form *event-based clusters* and a leader is selected. The sensed information is locally collected by the leader, which forwards this information to the sink only after a certain confidence level has been satisfied. The collected information consists of the location of the node and the timestamp related to the sensed event. Using this information the location and speed of the moving object are estimated.

Each cycle is periodically initiated by a reset phase followed by the system initialization phase. The system spends a very short time, e.g., 2 minutes, during the first four phases (system initialization, neighbor discovery, sentry selection, and status report), which update the system-wide parameters to be adopted in the dynamics of the network. The fifth phase (power management and event tracking) constitutes actual application operation, where the event-based tracking takes place. This phase may last up to a day until the next cycle. This sequential operation also helps isolate one system management task from another as well as the actual tracking. This helps conserve the limited resources and the wireless bandwidth.

VigilNet illustrates how several functionalities of a WSN are integrated into a coherent framework in a practical setting. The experimental evaluations reveal that the local aggregation and the hierarchical sleep/wakeup schemes significantly contribute to the energy efficiency of a tracking application in military settings.

In addition to these specific projects, WSNs have been widely used in battlefield surveillance applications, where exposure of military personnel to hazardous materials and enemy attack are minimized. Furthermore, WSNs are used for target tracking, where the location of a known or a mobile object is detected by a group of sensors and reported to the command control stations.

2.2 Environmental Applications

The autonomous coordination capabilities of WSNs are utilized in the realization of a wide variety of environmental applications. Some environmental applications of WSNs include tracking the movements of birds, small animals, and insects; monitoring environmental conditions that affect crops and livestock; irrigation; macroinstruments for large-scale Earth monitoring and planetary exploration; chemical/biological detection; precision agriculture; biological, Earth, and environmental monitoring in marine, soil, and atmospheric contexts; forest fire detection; meteorological or geophysical research; flood detection; biocomplexity mapping of the environment; and pollution studies [6, 10, 14, 17, 18, 19, 21, 30, 31, 32, 33, 34, 38, 49, 50, 51, 52, 54].

2.2.1 Great Duck Island

The Great Duck Island project [38, 50] has been developed as a collaboration between College of the Atlantic and the Intel Research Laboratory at Berkeley to study the distribution and abundance

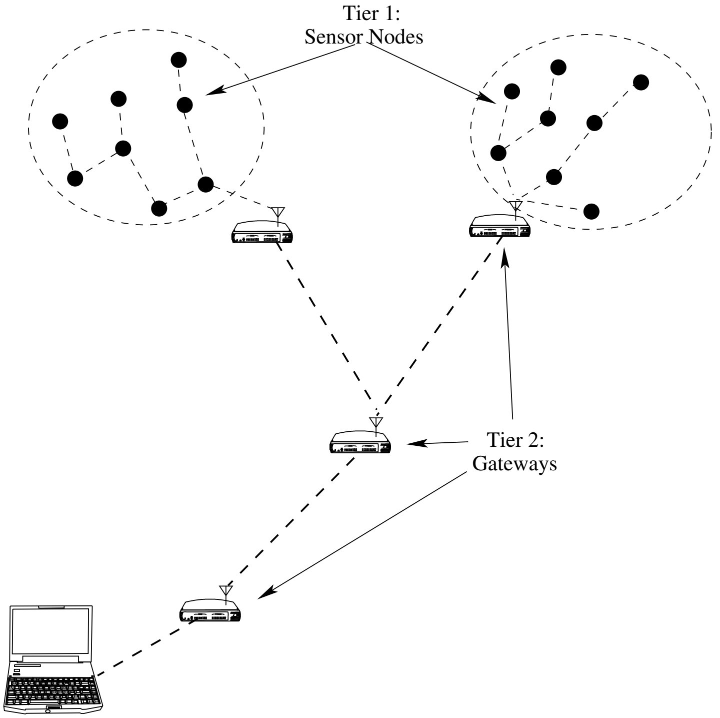


Figure 2.5 Two-tier architecture of Great Duck Island habitat monitoring network [38].

of seabirds on Great Duck Island (GDI), Maine. A network of Mica sensors is used to measure the occupancy of nesting burrows and the role of microclimatic factors in the habitat selection of seabirds, i.e., petrels. More specifically, the occupancy of the burrows during 1–3 days is monitored as well as the environmental changes during the breeding season and the corresponding changes in bird behavior.

The GDI habitat monitoring network employs a two-tier hierarchical architecture as shown in Figure 2.5. In the first tier, groups of sensors are used to gather information. For this tier, Mica motes equipped with Mica Weather Boards are used, which include temperature, photoresistor, barometric pressure, humidity, and thermopile sensors. More specifically, these sensor nodes are used for two different purposes: *burrow motes* are deployed in the burrows to detect occupancy using non-contact infrared thermopiles and temperature/humidity sensors, while *weather motes* are used to monitor the surface microclimate. Each group of sensors is connected to a gateway that collects these data and sends them to the second tier. The second tier consists of gateways that provide connectivity between the sensor groups in the field and the remote base station on the island through long-haul point-to-point communication. Two different platforms are used for this purpose: an embedded Linux system equipped with an IEEE 802.11 card and a 12 dBi omnidirectional antenna with a range of 1000 feet (30.5 m). Moreover, the Mica nodes are equipped with a 916 MHz directional antenna for a range of 1200 feet (365.75 m). The base station is also connected to the Internet by satellite, which enables remote access to the WSN.

The GDI project was conducted in 2002 and provided essential information on the characteristics of seabirds. In addition, the GDI network stands as one of the first proof-of-concept deployments of the WSN phenomenon for environmental applications.

2.2.2 CORIE

The Columbia River Ecosystem (CORIE) [6] is an example of an environmental observation and forecasting system (EOFS) built by the Center for Coastal and Land Margin Research at the Oregon Graduate Institute. It consists of sensor stations in the Columbia River estuary that carry various environmental sensors. In all, 24 stations in or around the Columbia River and other Oregon estuaries are equipped with various environmental sensors to measure water velocity, water temperature, salinity, and depth. Moreover, several meteorological stations are deployed to measure wind and air properties, and determine solar radiation at different wavelengths. The information gathered from these sensors is used for online control of vessels, marine research and rescue, and ecosystem research and management. The stations communicate via a Freewave DGR-115 spread-spectrum wireless network. In addition, the transmission of signals is accomplished by an ORBCOMM LEO satellite in case of disruptions in line of sight. Accordingly, the observations are displayed via the World Wide Web, in near real time.

The Columbia River estuary system includes the lower river, the estuary, and the near ocean. The goal of the CORIE system is to combine the observed information from the sensors with numerical data models to characterize the physical state of this system. More specifically, the goal is to characterize the complex circulation and mixing processes and predict any changes [24]. Since a large area of the river is covered by a limited number of sensors, the CORIE network represents significantly different characteristics than traditional sensor networks. Compared to low-cost sensor motes, the sensors used in CORIE are large complex systems that are expensive to deploy and maintain. Therefore, highly accurate data gathering solutions with a minimum number of stations are required. This necessitates estimation methods tailored to the data models used by the domain scientists.

2.2.3 ZebraNet

ZebraNet [33, 54] is an animal tracking system developed to investigate the long-term movement patterns of zebras, their interactions within and between species, as well as the impacts of human development. The system was deployed in Kenya to track two species of zebras.

ZebraNet consists of collars attached to the zebras to collect location information. Each collar includes a GPS unit, associated microcontroller, two types of radios (short range and long range), a lithium-ion polymer high-energy-density battery, and a solar array for recharging the battery as shown in Figure 2.6. The attached collar logs location information through the GPS unit every 3 minutes, which is shown through biological studies to result in statistically independent location samples that can be used to develop mobility models. This information is collected through a base station, which is kept by the researchers and used intermittently during their trips into the field. Therefore, ZebraNet can be characterized as a highly mobile sensor network without a static sink. Accordingly, the location information collected at each node is transmitted in a *delay-tolerant* manner, i.e., when the opportunity to communicate with the base station arises.

Since a particular node may not be within communication range of the mobile sink for a very long time, a data sharing policy is adopted in ZebraNet so that each sensor node shares the collected information with its neighbors. As a result, when one of the nodes establishes a connection with the sink, the location information of this node as well as its previous neighbors is downloaded. For data sharing, two different delay-tolerant routing protocols are used. Firstly, a simple flooding protocol is used, where each node sends its data to its neighbor whenever it detects a new neighbor. This improves the chance to deliver the data to the sink but increases the data overload in the network and limits the storage space at each node. The second protocol is based on the successful sink communication history of the nodes and assigns each node a hierarchy level [33]. If a node has recently communicated with a node it is assigned a higher level. Each node shares its location information with a node at a higher hierarchy level with higher probability.

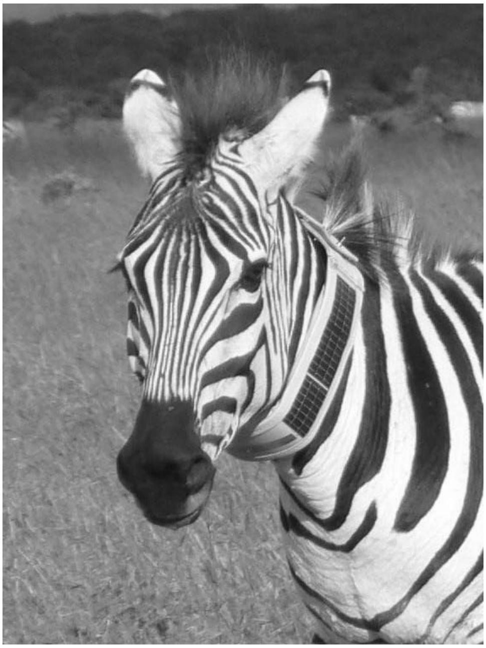


Figure 2.6 The ZebraNet sensor collar attached to a zebra [54].

The location information of the zebras collected by a group of sensors provides the domain scientists with valuable information to model the movement of these animals. In addition to its contribution to the domain sciences, this movement information is also used to fine-tune the developed protocols according to the mobility patterns of the zebras. The network deployment has provided an important insight into the operation of these networks and resulted in the development of communication protocols such as Z-MAC [47] that was devised to address the unique challenges of this application.

2.2.4 Volcano Monitoring

WSNs have also been used in extreme environments, where continuous human access is impossible. Volcano monitoring is an example of these extreme applications, where a network of sensors can be easily deployed near active volcanoes to continuously monitor their activities and provide data at a scale and resolution not previously possible with existing tools.

Two case studies were conducted on two volcanoes in Ecuador during 2004–2005 as a proof of concept of WSN applications in volcano monitoring [52]. In 2004, a small network of three sensor nodes, equipped with microphones, was used to monitor an erupting volcano, i.e., Volcán Tangurahua in central Ecuador. In 2005, 16 TMote Sky nodes equipped with seismic and acoustic sensors were used for a duration of 19 days to monitor the activities of an active volcano, Volcán Reventador, in northern Ecuador. The sensor nodes were equipped with higher gain external antennas to improve the communication range and three long-haul communication nodes were used to transmit the data to a central controller covering a 3 km array. A laptop equipped with a directional antenna was used as a sink to collect information and manage the network remotely.

The main goal of the application was to collect seismic information based on earthquakes that occur near the volcanoes. Since these earthquakes usually last less than 60 seconds, a high sampling rate is employed for the seismic sensors, i.e., 100 Hz, which limits the amount of locally stored information to 20 minutes of volcanic activity. Each sensor node uses the local storage as a cyclic buffer and filters the collected information through a short-term average/long-term average threshold detector to determine the events related to volcanic activities. When an event is generated, it is reported to the sink. If high confidence of an event is established through multiple event reports, the sink sends back a data collection command to the network. As a response to this command, each node transmits the information stored in its memory. Consequently, a large volume of data is transmitted to the sink only if an event of interest is detected by a smaller number of nodes. This reduces the bandwidth wastage and energy consumption at individual nodes.

The sensor nodes are synchronized using the Flooding Time Synchronization protocol (FTSP) [40] and a GPS unit equipped with a MicaZ node to provide location information. The results of this application provide important information related to the physical processes at work within a volcano's interior.

2.2.5 Early Flood Detection

WSNs are also used for early warning flood detection in developing countries [17]. The main challenge with existing systems is the involvement of personnel for continuous monitoring of river beds. Instead, model-based prediction systems can be used by exploiting the statistical properties of data collected by a network of sensor nodes. Such a system has been developed at MIT and tested in Honduras, where frequent floods significantly affect urban life.

Flood monitoring requires a large area to be covered with sensors. Considering the limited communication range of individual sensors, a two-tier network topology is used as shown in Figure 2.7. Three different sensors are used in the lower tier of the network for measuring rainfall, air temperature, and water flow data. These types of data are required for the prediction model. Each closely located sensor forms a group and is connected to the second-tier computation nodes. Data collection and information processing are performed at the computation nodes, which inform the third tier, i.e., control centers, in case of a potential flood. Overall, four different types of nodes are used in the system. Each node has the same base system that includes a microcontroller and a transceiver but is enhanced with sister-boards according to its function:

- **Sensing node:** This node is equipped with one of the sensors monitoring rainfall, air temperature, and water flow and collects information over a period on the order of minutes [17]. Each node is equipped with a 900 MHz transceiver and reports its data to the computation node in its cluster. While the rainfall and temperature sensors are located on the ground, the water flow data requires underwater measurements. The water level can be easily converted into water flow. To measure the water level, a water pressure sensor is interfaced with the base board and deployed under water.
- **Computation node:** Each group of closely located sensing nodes is maintained by a computation node as shown in Figure 2.7. The computation node has two main duties in the network. Firstly, the data set collected from the group of sensors is fed into the prediction model, which estimates the river flow. In case the prediction confidence is not high, the computation node requests additional information from the sensing nodes. Secondly, the computation node provides long-haul communication between the group and the control and monitoring nodes through a 144 MHz transceiver. The information collected by different groups of sensing nodes is collected at the interface nodes. Through long-haul communication techniques, the computation nodes can directly reach the interface nodes. This two-tier approach improves the scalability of the system.
- **Government office interface node:** This node provides a user interface to the network by visualizing data and enabling network maintenance tasks. Larger scale predictions can also be performed using the data received from multiple sensors located at different locations.

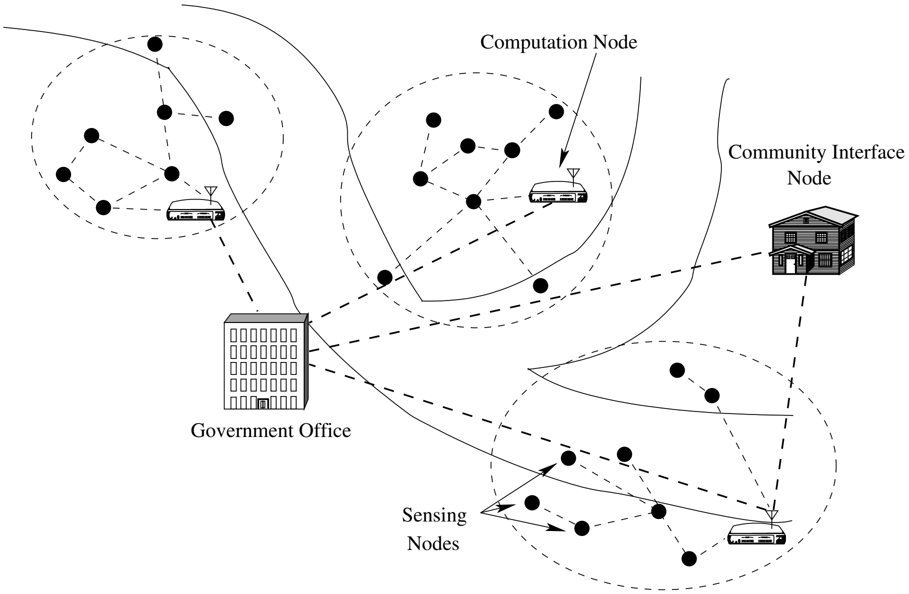


Figure 2.7 Sensor network architecture for flood detection [17].

- **Community interface node:** Finally, the flood prediction system will be connected to community interface nodes that inform the interested community about the results of the prediction. This allows the necessary action to be taken timely in the case of severe floods.

The prediction model has been validated using 7-year data collected from the Blue River, Oklahoma, through the Distributed Model Intercomparison project run by the National Oceanic and Atmospheric Administration. Moreover, the lower tier of the system has been tested in Charles River, Massachusetts, during October–November 2007 using a group of three sensors. Finally, the flood prediction system was deployed in Honduras over a long period of time between January 2004 and March 2007.

2.3 Health Applications

The developments in implanted biomedical devices and smart integrated sensors make the usage of sensor networks for biomedical applications possible. Some of the health applications for sensor networks are the provision of interfaces for the disabled; integrated patient monitoring; diagnostics; drug administration in hospitals; monitoring the movements and internal processes of insects or other small animals; telemonitoring of human physiological data; and tracking and monitoring doctors and patients inside a hospital [4, 28, 34, 39, 41, 46, 51, 53].

2.3.1 Artificial Retina

The Artificial Retina (AR) project supported by the US Department of Energy aims to build a chronically implanted artificial retina for visually impaired people [4]. More specifically, the project focuses on addressing two retinal diseases, namely age-related macular degeneration (AMD) and retinitis pigmentosa (RP). AMD is an age-related disease and results in severe vision loss at the center of the retina caused by fluid leakage or bleeding in people aged 60 and above. RP, on the other hand, affects

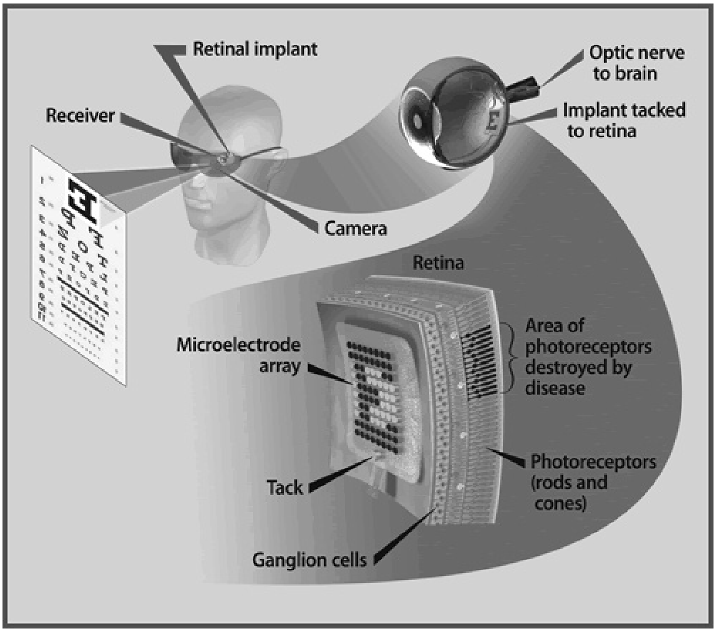


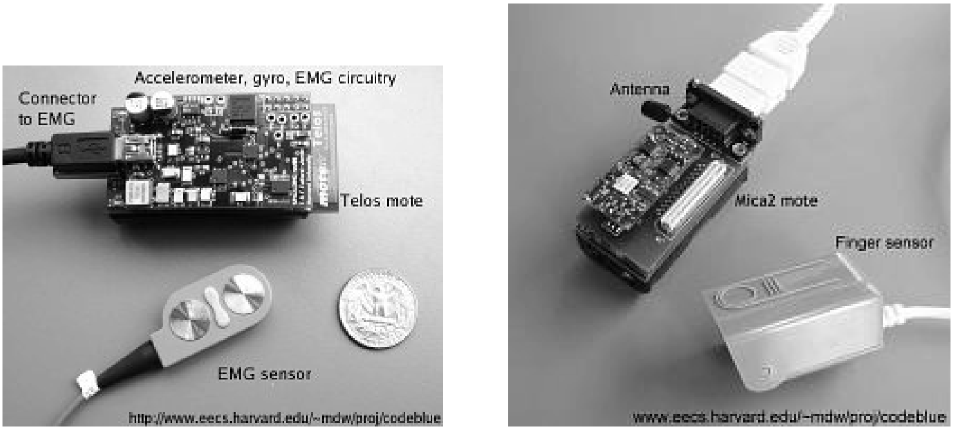
Figure 2.8 Operating principle of the artificial retina [4].

the photoreceptor or rod cells, which results in a loss of peripheral vision [4]. A healthy photoreceptor stimulates the brain through electric impulses when light is illuminated from the external world. When damaged, vision is blocked at the locations of the photoreceptors. The AR project aims to replace these damaged photoreceptors with an array of microsensors.

The receptors along the retina have different functionalities. The periphery is specialized on temporal events, whereas the center of the macula is specialized on spatially oriented information. In order to mimic the functionality of the retina, the distribution and the transmission principles of smart sensors on each segment of the retina must be tailored.

The operating principle of the AR is shown in Figure 2.8. A tiny camera placed in eyeglasses is used to capture visual information from the external world. This information is wirelessly transmitted to a microprocessor, which converts it into electrical signals. These signals are then sent to an array of microelectrodes, which are implanted along the retina at the locations of the photoreceptors. The microelectrodes then send electric impulses to the brain via the optic nerve.

So far, three models have been developed as part of the AR project. The first model, Argus I, has been completely tested and implanted into six patients between 2002 and 2004. This model consists of a 16-electrode array and helps the patients to detect whether lights are on or off, describe the motion of an object, count individual items, and locate objects. The second model, Argus II, consists of a 60-electrode array and is undergoing clinical trials. Finally, the third model, which is under development, will include more than 200 electrodes. The ultimate goal for the prosthetic device is to create a lasting device that will enable facial recognition and the ability to read large print [4].


Accelerometer, gyro, EMG circuitry
Connector to EMG
Telos mote
EMG sensor
http://www.eecs.harvard.edu/~mdw/proj/codeblue
Antenna
Mica2 mote
Finger sensor
www.eecs.harvard.edu/~mdw/proj/codeblue

(a) (b)

Figure 2.9 Medical sensor nodes used in CodeBlue: (a) Telos mote with EMG sensor and (b) Mica2 mote with pulse oximeter sensor [7].

2.3.2 Patient Monitoring

The CodeBlue project at Harvard University focuses on wearable sensors that monitor vital signs of patients throughout their daily lives [39]. To this end, sensor boards with pulse oximeter, electrocardiograph (EKG), and electromyograph (EMG) circuitry have been designed for MicaZ and Telos motes as shown in Figure 2.9. Accordingly, pulse rate, blood oxygen saturation, electrical activities of the heart, patient movements, and muscular activity can be monitored continuously. The CodeBlue software platform enables these nodes to be operated in a networked setting, where medical personnel can monitor patients through a PDA.

The architecture of the CodeBlue network is shown in Figure 2.10 in a hospital setting. Accordingly, sensor motes with various sensors are attached to patients for monitoring. Additional motes are deployed inside the hospital as an infrastructure to provide message delivery to/from the motes. The medical personnel can access the network through either PDAs or computers. The network is based on a *publish/subscribe* mechanism, where the sensor motes publish the available data and the medical personnel subscribe to this data based on certain conditions. The publish/subscribe mechanism is based on the Adaptive Demand-Drive Multicast Routing (ADMR) protocol, which maintains routing information based on the published information. Whenever a mote publishes data, each node in the network is informed about the best path to this node. Route discovery between nodes is performed periodically through limited flooding to adapt to the changes in the wireless channel.

When a sensor is connected to the network, its capabilities are communicated to the network through a discovery protocol. Subscribing to the published data is performed through the CodeBlue query (CBQ) layer, which specifies a query through a tuple $\langle S, \tau, chan, \rho, C, p\rangle$, where S is the set of subscribed node IDs, τ is the subscribed sensor type, *chan* is the channel type used for routing, ρ is the sampling rate, and C is the optional maximum number of samples to be reported. To model the *interest*s of the users, a filter predicate, p, is also used in the query, according to which the sensor information to be reported is filtered. Accordingly, a doctor may specify a sensor connected to a particular patient and request to receive heartbeat information only if it exceeds normal limits.

In addition to the publish/subscribe mechanism, CodeBlue also provides a localization service through a set of nodes that periodically broadcast beacon messages. A mobile node can then estimate its location based on the received RF signal signature from these beacon nodes. This information is essential in