Wireless Sensor Networks – An Introduction

Chapter · December 2010

DOI: 10.5772/13225 · Source: InTech

CITATIONS
44

READS
5,275

2 authors:

Qinghua Wang
Kristianstad University
20 PUBLICATIONS **260** CITATIONS
SEE PROFILE

Ilangko Balasingham
Oslo University Hospital & Norwegian University of Science and Technology
294 PUBLICATIONS **3,176** CITATIONS
SEE PROFILE

1

Wireless Sensor Networks - An Introduction

Qinghua Wang
Dept. of Electronics and Telecommunications
Norwegian University of Science and Technology
Norway

Ilangko Balasingham
Dept. of Electronics and Telecommunications
Norwegian University of Science and Technology
The Interventional Centre, Oslo University Hospital
Institute of Clinical Medicine, University of Oslo
Norway

This chapter provides a detailed introduction to the history and current state of the art with regard to wireless sensor networks (WSNs).

1. History

The origins of the research on WSNs can be traced back to the Distributed Sensor Networks (DSN) program at the Defense Advanced Research Projects Agency (DARPA) at around 1980. By this time, the ARPANET (Advanced Research Projects Agency Network) had been operational for a number of years, with about 200 hosts at universities and research institutes (Chong & Kumar, 2003). DSNs were assumed to have many spatially distributed low-cost sensing nodes that collaborated with each other but operated autonomously, with information being routed to whichever node was best able to use the information. At that time, this was actually an ambitious program. There were no personal computers and workstations; processing was mainly performed on minicomputers and the Ethernet was just becoming popular (Chong & Kumar, 2003). Technology components for a DSN were identified in a Distributed Sensor Nets workshop in 1978 (*Proceedings of the Distributed Sensor Nets Workshop*, 1978). These included sensors (acoustic), communication and processing modules, and distributed software. Researchers at Carnegie Mellon University (CMU) even developed a communication-oriented operating system called Accent (Rashid & Robertson, 1981), which allowed flexible, transparent accesss to distributed resources required for a fault-tolerant DSN. A demonstrative application of DSN was a helicopter tracking system (Myers et al., 1984), using a distributed array of acoustic microphones by means of signal abstractions and matching techniques, developed at the Massachusetts Institute of Technology (MIT).
Even though early researchers on sensor networks had in mind the vision of a DSN, the technology was not quite ready. More specifically, the sensors were rather large (i.e. shoe box and

This work was carried out during the tenure of an ERCIM "Alain Bensoussan" Fellowship Programme and is part of the MELODY Project, which is funded by the Research Council of Norway under the contract number 187857/S10.

up) which limited the number of potential applications. Further, the earliest DSNs were not tightly associated with wireless connectivity. Recent advances in computing, communication and microelectromechanical technology have caused a significant shift in WSN research and brought it closer to achieving the original vision. The new wave of research in WSNs started in around 1998 and has been attracting more and more attention and international involvement. In the new wave of sensor network research, networking techniques and networked information processing suitable for highly dynamic ad hoc environments and resource-constrained sensor nodes have been the focus. Further, the sensor nodes have been much smaller in size (i.e. pack of cards to dust particle) and much cheaper in price, and thus many new civilian applications of sensor networks such as environment monitoring, vehicular sensor network and body sensor network have emerged. Again, DARPA acted as a pioneer in the new wave of sensor network research by launching an initiative research program called SensIT (Kumar & Shepherd, 2001) which provided the present sensor networks with new capabilities such as ad hoc networking, dynamic querying and tasking, reprogramming and multi-tasking. At the same time, the IEEE noticed the low expense and high capabilities that sensor networks offer. The organization has defined the IEEE 802.15.4 standard (*IEEE 802.15 WPAN Task Group 4*, n.d.) for low data rate wireless personal area networks. Based on IEEE 802.15.4, ZigBee Alliance (*ZigBee Alliance*, n.d.) has published the ZigBee standard which specifies a suite of high level communication protocols which can be used by WSNs. Currently, WSN has been viewed as one of the most important technologies for the 21st century (*21 Ideas for the 21st Century*, 1999). Countries such as China have involved WSNs in their national strategic research programmes (Ni, 2008). The commercialization of WSNs are also being accelerated by new formed companies like Crossbow Technology (*Crossbow Technology*, n.d.) and Dust Networks (*Dust Networks, Inc.*, n.d.).

2. Hardware Platform

A WSN consists of spatially distributed sensor nodes. In a WSN, each sensor node is able to independently perform some processing and sensing tasks. Furthermore, sensor nodes communicate with each other in order to forward their sensed information to a central processing unit or conduct some local coordination such as data fusion. One widely used sensor node platform is the Mica2 Mote developed by Crossbow Technology (*Crossbow Technology*, n.d.). The usual hardware components of a sensor node include a radio transceiver, an embedded processor, internal and external memories, a power source and one or more sensors.

2.1 Embedded Processor

In a sensor node, the functionality of an embedded processor is to schedule tasks, process data and control the functionality of other hardware components. The types of embedded processors that can be used in a sensor node include Microcontroller, Digital Signal Processor (DSP), Field Programmable Gate Array (FPGA) and Application-Specific Integrated Circuit (ASIC). Among all these alternatives, the Microcontroller has been the most used embedded processor for sensor nodes because of its flexibility to connect to other devices and its cheap price. For example, the newest CC2531 development board provided by Chipcon (acquired by Texas Instruments) uses 8051 microcontroller, and the Mica2 Mote platform provided by Crossbow uses ATMega128L microcontroller.

2.2 Transceiver

A transceiver is responsible for the wireless communication of a sensor node. The various choices of wireless transmission media include Radio Frequency (RF), Laser and Infrared. RF based communication fits to most of WSN applications. The operational states of a transceiver are Transmit, Receive, Idle and Sleep. Mica2 Mote uses two kinds of RF radios: RFM TR1000 and Chipcon CC1000. The outdoor transmission range of Mica2 Mote is about 150 meters.

2.3 Memory

Memories in a sensor node include in-chip flash memory and RAM of a microcontroller and external flash memory. For example, the ATMega128L microcontroller running on Mica2 Mote has 128-Kbyte flash program memory and 4-Kbyte static RAM. Further, a 4-Mbit Atemel AT45DB041B serial flash chip can provide external memories for Mica and Mica2 Motes (Hill, 2003).

2.4 Power Source

In a sensor node, power is consumed by sensing, communication and data processing. More energy is required for data communication than for sensing and data processing. Power can be stored in batteries or capacitors. Batteries are the main source of power supply for sensor nodes. For example, Mica2 Mote runs on 2 AA batteries. Due to the limited capacity of batteries, minimizing the energy consumption is always a key concern during WSN operations. To remove the energy constraint, some preliminary research working on energy-harvesting techniques for WSNs has also been conducted. Energy-harvesting techniques convert ambient energy (e.g. solar, wind) to electrical energy and the aim is to revolutionize the power supply on sensor nodes. A survey about the energy-harvesting sensor nodes is provided by (Sudevalayam & Kulkarni, 2008).

2.5 Sensors

A sensor is a hardware device that produces a measurable response signal to a change in a physical condition such as temperature, pressure and humidity. The continual analog signal sensed by the sensors is digitized by an analog-to-digital converter and sent to the embedded processor for further processing. Because a sensor node is a micro-electronic device powered by a limited power source, the attached sensors should also be small in size and consume extremely low energy. A sensor node can have one or several types of sensors integrated in or connected to the node.

3. Operating System

The role of any operating system (OS) is to promote the development of reliable application software by providing a convenient and safe abstraction of hardware resources. OSs for WSN nodes are typically less complex than general-purpose OSs both because of the special requirements of WSN applications and because of the resource constraints in WSN hardware platforms.

TinyOS (*TinyOS Community Forum*, n.d.) is perhaps the first operating system specifically designed for WSNs. It features a component-based architecture which enables rapid innovation and implementation while minimizing code size as required by the severe memory constraints inherent in WSNs. TinyOS's component library includes network protocols, distributed services, sensor drivers, and data acquisition tools - all of which can be further refined for a

custom application. Unlike most other OSs, TinyOS is based on an event-driven programming model instead of multithreading. TinyOS programs are composed into event handlers and tasks with run-to-completion semantics. When an external event occurs, such as an incoming data packet or a sensor reading, TinyOS calls the appropriate event handler to handle the event. Event handlers can post tasks that are scheduled by the TinyOS kernel at a later stage. Both the TinyOS system and programs written for TinyOS are written in a special programming language called nesC which is an extension of the C programming language. NesC is designed to detect race conditions between tasks and event handlers. Currently, TinyOS has been ported to over a dozen platforms and numerous sensor boards. A wide community uses it in simulation to develop and test various algorithms and protocols. According to the figure published on TinyOS forum, over 500 research groups and companies are using TinyOS on the Berkeley/Crossbow Motes. Because TinyOS is open source, numerous groups are actively contributing code to the development of TinyOS and thus making it even more competitive.
Contiki (*Contiki*, n.d.) is another open source OS specifically designed for WSNs. The Contiki kernel is event-driven, like TinyOS, but the system supports multithreading on a per-application basis. Furthermore, Contiki includes protothreads that provide a thread-like programming abstraction but with a very small memory overhead. Contiki provides IP communication, both for IPv4 and IPv6. Many key mechanisms and ideas from Contiki have been widely adopted within the industry. The uIP embedded IP stack, originally released in 2001, is today used by hundreds of companies in systems such as freighter ships, satellites and oil drilling equipment. Contiki's protothreads, first released in 2005, have been used in many different embedded systems, ranging from digital TV decoders to wireless vibration sensors. Contiki's idea of using IP communication in low-power WSNs has lead to an IETF standard and an international industry alliance - IP for Smart Objects (IPSO) Alliance (*IPSO Alliance - promoting the use of IP for Smart Objects*, n.d.).
There are also other OSs that can be used by WSNs. For example, SOS (*SOS Embedded Operating System*, n.d.) is an event-driven OS for mote-class sensor nodes that adopts a more dynamic point on the design spectrum. The prime feature of SOS is its support for loadable modules. A complete system is built from smaller modules, possibly at run-time. To support the inherent dynamism in its module interface, SOS also focuses on supporting dynamic memory management. Unfortunately, SOS is no longer under active development due to the graduation of the core developers. LiteOS (*LiteOS*, n.d.) is an open source, interactive, UNIX-like operating system designed for WSNs. With the tools that come from LiteOS, it is possible to operate one or more WSNs in a Unix-like manner. It is also possible to develop programs for nodes, and wirelessly distribute such programs to sensor nodes.

4. Networking

4.1 Network Architecture

A WSN is a network consisting of numerous sensor nodes with sensing, wireless communications and computing capabilities. These sensor nodes are scattered in an unattended environment (i.e. sensing field) to sense the physical world. The sensed data can be collected by a few sink nodes which have accesses to infrastructured networks like the Internet. Finally, an end user can remotely fetch the sensed data by accessing infrastructured networks. Fig. 1 shows the operation sketch map of WSNs.
In Fig. 1, two kinds of network topologies are shown. The sensor nodes either form a flat network topology where sensor nodes also act as routers and trasfer data to a sink through

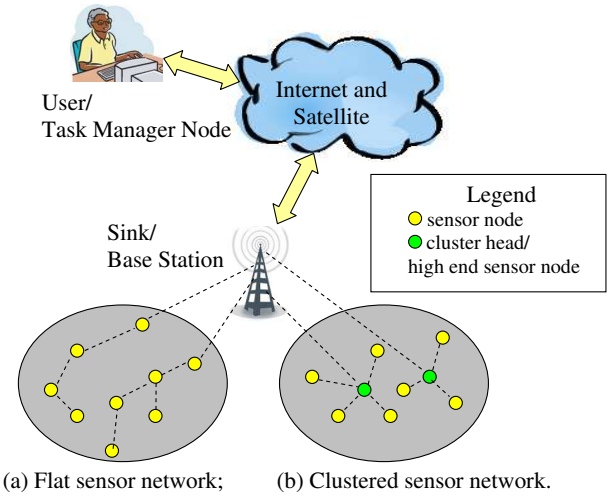

Fig. 1. The Operation of WSNs.

multi-hop routing, or a hierarchical network topology where more powerful fixed or mobile relays are used to collect and route the sensor data to a sink.

4.2 Protocol Stack of WSNs

The protocol stack used by the sink, cluster head and sensor nodes are shown in Fig. 2. According to (Akyildiz et al., 2002), the sensor network protocol stack is much like the traditional protocol stack, with the following layers: application, transport, network, data link, and physical. The physical layer is responsible for frequency selection, carrier frequency generation, signal detection, modulation and data encryption. The data link layer is responsible for the multiplexing of data streams, data frame detection, medium access and error control. It ensures reliable point-to-point and point-to-multipoint connections in a communication network. The network layer takes care of routing the data supplied by the transport layer. The network layer design in WSNs must consider the power efficiency, data-centric communication, data aggregation, etc. The transportation layer helps to maintain the data flow and may be important if WSNs are planned to be accessed through the Internet or other external networks. Depending on the sensing tasks, different types of application software can be set up and used on the application layer.

WSNs must also be aware of the following management planes in order to function efficiently: mobility, power, task, quality of service (QoS) and security management planes. Among them, the functions of task, mobility and power management planes have been elaborated in (Akyildiz et al., 2002). The power management plane is responsible for minimizing power consumption and may turn off functionality in order to preserve energy. The mobility management plane detects and registers movement of nodes so a data route to the sink is always maintained. The task management plane balances and schedules the sensing tasks assigned to the sensing field and thus only the necessary nodes are assigned with sensing tasks and the remainder are able to focus on routing and data aggregation. QoS management in WSNs (Howitt et al., 2006) can be very important if there is a real-time requirement with regard to

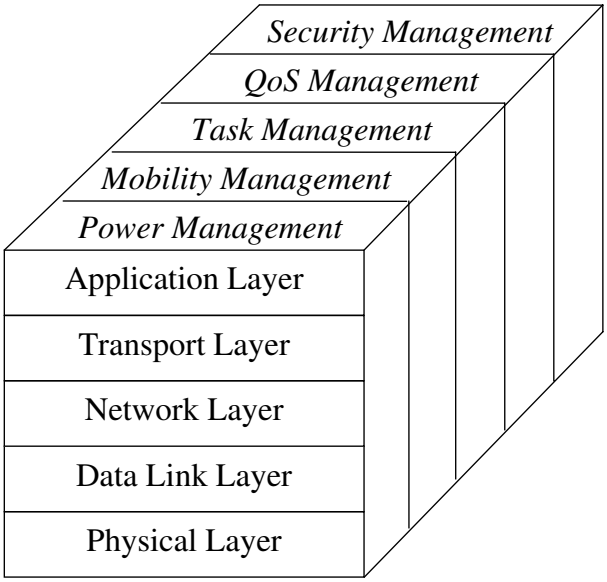

Fig. 2. The Protocol Stack of WSNs (This figure is an extended version of Figure 3 in (Akyildiz et al., 2002)).

the data services. QoS management also deals with fault tolerance, error control and performance optimization in terms of certain QoS metrics. Security management is the process of managing, monitoring, and controlling the security related behavior of a network. The primary function of security management is in controlling access points to critical or sensitive data. Security management also includes the seamless integration of different security function modules, including encryption, authentication and intrusion detection. Please refer to the author's publications (Wang & Zhang, 2008; 2009) for more information about security management in WSNs. It is obvious that networking protocols developed for WSNs must address all five of these management planes.

5. Applications

The original motivation behind the research into WSNs was military application. Examples of military sensor networks include large-scale acoustic ocean surveillance systems for the detection of submarines, self-organized and randomly deployed WSNs for battlefield surveillance and attaching microsensors to weapons for stockpile surveillance (Pister, 2000). As the costs for sensor nodes and communication networks have been reduced, many other potential applications including those for civilian purposes have emerged. The following are a few examples.

5.1 Environmental Monitoring

Environmental monitoring (Steere et al., 2000) can be used for animal tracking, forest surveillance, flood detection, and weather forecasting. It is a natural candidate for applying WSNs (Chong & Kumar, 2003), because the variables to be monitored, e.g. temperature, are usu-

ally distributed over a large region. One example is that researchers from the University of Southampton have built a glacial environment monitoring system using WSNs in Norway (Martinez et al., 2005). They collect data from sensor nodes installed within the ice and the sub-glacial sediment without the use of wires which could disturb the environment. Another example is that researchers from EPFL have performed outdoor WSN deployments on a rugged high mountain path located between Switzerland and Italy (Barrenetxea et al., 2008). Their WSN deployment is used to provide spatially dense measures to the Swiss authorities in charge of risk management, and the resulting model will assist in the prevention of avalanches and accidental deaths.

5.2 Health Monitoring

INI PENTING !

WSNs can be embedded into a hospital building to track and monitor patients and all medical resources. Special kinds of sensors which can measure blood pressure, body temperature and electrocardiograph (ECG) can even be knitted into clothes to provide remote nursing for the elderly. When the sensors are worn or implanted for healthcare purposes, they form a special kind of sensor network called a body sensor network (BSN). BSN is a rich interdisciplinary area which revolutionizes the healthcare system by allowing inexpensive, continuous and ambulatory health monitoring with real-time updates of medical records via the Internet. One of the earliest researches on BSNs was conducted in Imperial College London, where a specialized BSN sensor node and BSN Development Kit have been developed (*BSN Research in Imperial College London*, n.d.).

5.3 Traffic Control

Sensor networks have been used for vehicle traffic monitoring and control for some time. At many crossroads, there are either overhead or buried sensors to detect vehicles and to control the traffic lights. Furthermore, video cameras are also frequently used to monitor road segments with heavy traffic. However, the traditional communication networks used to connect these sensors are costly, and thus traffic monitoring is usually only available at a few critical points in a city (Chong & Kumar, 2003). WSNs will completely change the landscape of traffic monitoring and control by installing cheap sensor nodes in the car, at the parking lots, along the roadside, etc. Streetline, Inc. (*Streetline, Inc.*, n.d.) is a company which uses sensor network technology to help drivers find unoccupied parking places and avoid traffic jams. The solutions provided by Streetline can significantly improve the city traffic management and reduce the emission of carbon dioxide.

5.4 Industrial Sensing

As plant infrastructure ages, equipment failures cause more and more unplanned downtime. The ARC Advisory Group estimates that 5% of production in North America is lost to unplanned downtime. Because sensor nodes can be deeply embedded into machines and there is no infrastructure, WSNs make it economically feasible to monitor the "health" of machines and to ensure safe operation. Aging pipelines and tanks have become a major problem in the oil and gas industry. Monitoring corrosion using manual processes is extremely costly, time consuming, and unreliable. A network of wireless corrosion sensors can be economically deployed to reliably identify issues before they become catastrophic failures. Rohrback Cosasco Systems (RCS) (*Rohrback Cosasco Systems*, n.d.) is the world leader in corrosion monitoring technology and is applying WSNs in their corrosion monitoring. WSNs have also been sug-

gested for use in the food industry to prevent the incidents of contaminating the food supply chain (Connolly & O'Reilly, 2005).

5.5 Infrastructure Security

WSNs can be used for infrastructure security and counterterrorism applications. Critical buildings and facilities such as power plants, airports, and military bases have to be protected from potential invasions. Networks of video, acoustic, and other sensors can be deployed around these facilities (Chong & Kumar, 2003). An initiative in Shanghai Pudong International Airport has involved the installation of a WSN-aided intrusion prevention system on its periphery to deter any unexpected intrusions. The Expo 2010 Shanghai China (*Expo 2010 Shanghai China*, n.d.) has also secured its expo sites with the same intrusion prevention system.

6. Security

While the future of WSNs is very prospective, WSNs will not be successfully deployed if security, dependability and privacy issues are not addressed adequately. These issues become more important because WSNs are usually used for very critical applications. Furthermore, WSNs are very vulnerable and thus attractive to attacks because of their limited prices and human-unattended deployment (Wang & Zhang, 2009).

6.1 Security Threats in WSNs

A typical WSN consists of hundreds or even thousands of tiny and resource-constrained sensor nodes. These sensor nodes are distributedly deployed in uncontrollable environment for the collection of security-sensitive information. Individual sensor nodes rely on multi-hop wireless communication to deliver the sensed data to a remote base station. In a basic WSN scenario, resource constraint, wireless communication, security-sensitive data, uncontrollable environment, and even distributed deployment are all vulnerabilities. These vulnerabilities make WSNs suffer from an amazing number of security threats. WSNs can only be used in the critical applications after the potential security threats are eliminated.

6.1.1 Physical Layer Threats

Comparing WSNs with traditional networks, there are more threats to WSNs in the physical layer, due to the non-tamper-resistant WSN nodes and the broadcasting nature of wireless transmission. Typical types of attacks in the physical layer include physical layer jamming and the subversion of a node.

6.1.2 Link Layer Threats

The data link layer is responsible for the multiplexing of data streams, data frame detection, medium access, and error control. The following attacks can happen in the link layer of WSNs: Link layer jamming; Eavesdropping; Resource exhaustion and traffic analysis.

6.1.3 Network Layer Threats

Threats in the network layer mostly aim at disturbing data-centric and energy efficient multihop routing, which is the main design principle in WSNs. The following threats and attacks in this layer are identified in (Wang & Zhang, 2009): Spoofed, altered, or replayed routing information; Sybil attack; Selective forwarding; Sinkhole attack; Wormhole attack and flooding.

6.1.4 Application Layer Threats

Many WSNs' applications heavily rely on coordinated services such as localization, time synchronization, and in-network data processing to collaboratively process data (Sabbah et al., 2006). Unfortunately, these services represent unique vulnerabilities such as: False data filtering; Clock un-synchronization; False data injection.

6.2 Countermeasures

WSN Threats presented above either violate network secrecy and authentication, such as packet spoofing, or violate network availability, such as jamming attack, or violate some other network functionalities. Generally, countermeasures to the threats in WSNs should fulfill the following security requirements (Wang & Zhang, 2009):

- Availability, which ensures that the desired network services are available whenever required.
- Authentication, which ensures that the communication from one node to another node is genuine.
- Confidentiality, which provides the privacy of the wireless communication channels.
- Integrity, which ensures that the message or the entity under consideration is not altered.
- Non-reputation, which prevents malicious nodes to hide or deny their activities.
- Freshness, which implies that the data is recent and ensures that no adversary can replay old messages.
- Survivability, which ensures the acceptable level of network services even in the presence of node failures and malicious attacks.
- Self-security, countermeasures may introduce additional hardware and software infrastructures into the network, which must themselves be secure enough to withstand attacks.

Depending on applications, countermeasures should also fulfill appropriate performance requirements.

6.2.1 Key Management

When setting up a sensor network, one of the first security requirements is to establish cryptographic keys for later secure communication. The established keys should be resilient to attacks and flexible to dynamic update. The task that supports the establishment and maintenance of key relationships between valid parties according to a security policy is called key management. Desired features of key management in sensor networks include energy awareness, localized impact of attacks, and scaling to a large number of nodes.

6.2.2 Authentication

As sensor networks are mostly deployed in human-unattended environments for critical sensing measurements, the authentication of the data source as well as the data are critical concerns. Proper authentication mechanisms can provide WSNs with both sensor and user identification ability, can protect the integrity and freshness of critical data, and can prohibit and identify impersonating attack. Traditionally, authentication can be provided by public-key schemes as digital signature and by symmetric-key schemes as message authentication code (MAC). Besides, key-chain schemes using symmetric keys determined by asymmetric key-exchange protocols are also popular for broadcast authentication in WSNs.

6.2.3 Intrusion Detection

Security technologies, such as authentication and cryptography, can enhance the security of sensor networks. Nevertheless, these preventive mechanisms alone cannot deter all possible attacks (e.g., insider attackers possessing the key). Intrusion detection, which has been successfully used in Internet, can provide a second line of defense.

6.2.4 Privacy Protection

As WSN applications expand to include increasingly sensitive measurements in both military tasks and everyday life, privacy protection becomes an increasingly important concern. For example, few people may enjoy the benefits of a body area WSN, if they know that their personal data such as heart rate, blood pressure, etc., are regularly transmitted without proper privacy protection. Also, the important data sink in a battlefield surveillance WSN may be firstly destroyed, if its location can be traced by analyzing the volume of radio activities.

7. Standardization

In the area of WSNs, several standards are currently either ratified or under development. The major standardization bodies are the Institute of Electrical and Electronics Engineers (IEEE), the Internet Engineering Task Force (IETF), the International Society for Automation (ISA) and the HART Communication Foundation, etc. These standardization bodies have different focuses and they provide global, open standards for interoperable, low-power wireless sensor devices. Table 1 provides the comparisons of different standards currently available for the communication protocols of WSNs.

7.1 IEEE 802.15.4

IEEE 802.15.4 is a standard which specifies the physical layer and MAC layer for low-rate wireless personal area networks. It is the basis for the ZigBee and WirelessHART specification, each of which further attempts to offer a complete networking solution by developing the upper layers which are not covered by the standard.
The features of IEEE 802.15.4 include (*IEEE 802.15 WPAN Task Group 4*, n.d.):

- Data rates of 250 kbps, 40 kbps, and 20 kbps.
- Two addressing modes; 16-bit short and 64-bit IEEE addressing.
- Support for critical latency devices, such as joysticks.
- CSMA-CA channel access.
- Automatic network establishment by the coordinator.
- Fully handshaked protocol for transfer reliability.
- Power management to ensure low power consumption.
- 16 channels in the 2.4GHz ISM band, 10 channels in the 915MHz ISM band and one channel in the 868MHz band.

7.2 Zigbee

ZigBee is a standard for a suite of high level communication protocols based on the IEEE 802.15.4 standard for low power and low data rate radio communications. Zigbee is initiated and maintained by the Zigbee Alliance - a large consortium of industry players. The typical application areas of Zigbee include: Smart energy monitoring; Health care monitoring; Remote control; Building automation and home automation, etc.

7.3 WirelessHART

WirelessHART is an open-standard wireless mesh network communications protocol designed to meet the needs for process automation applications. The protocol utilizes IEEE 802.15.4 compatible DSSS radios and it is operating in the 2.4GHz ISM radio band. On the data link layer, the protocol uses TDMA technology to arbitrate and coordinate communications between devices. WirelessHART provides highly secure communications by using AES-128 block ciphers with individual Join and Session Keys and Data-Link level Network Key. WirelessHART supports the standard HART Application Layer and is compatible with existing HART tools, applications and system integration technology. The other outstanding features of WirelessHART include reliability and scalability. Typically, the communication reliability for a well-formed WirelessHART network is greater than 3σ and normally greater than 6σ. Adding new devices can further improve the network and its communication reliability.

8. Summary

WSNs have been identified as one of the most prospective technologies in this century. This chapter provides information concerning both its history and current state of the art. In concrete terms, the authors provide an overview about the hardware, software and networking protocol design of this important technology. The authors also discuss the security and ongoing standardization of this technology. Depending on applications, many other techniques such as localization, synchronization and in-network processing can be important, which are not discussed in this chapter.

9. References

21 Ideas for the 21st Century (1999). *Business Week* pp. 78–167.

Akyildiz, I. F., Su, W., Sankarasubramaniam, Y. & Cayirci, E. (2002). A survey on sensor networks, *IEEE Communications Magazine* **40**(8): 102–114.

Barrenetxea, G., Ingelrest, F., Schaefer, G. & Vetterli, M. (2008). Wireless sensor networks for environmental monitoring: The sensorscope experience, *Proc. of 20th IEEE International Zurich Seminar on Communications (IZS'08)*.

BSN Research in Imperial College London (n.d.). http://ubimon.doc.ic.ac.uk/bsn/m621.html.

Chong, C.-Y. & Kumar, S. P. (2003). Sensor networks: Evolution, opportunities, and challenges, *Proceedings of the IEEE* **91**(8): 1247–1256.

Connolly, M. & O'Reilly, F. (2005). Sensor networks and the food industry, *Proc. of Workshop on Real-World Wireless Sensor Networks (REALWSN'05)*.

Contiki (n.d.). http://www.sics.se/contiki.

Crossbow Technology (n.d.). http://www.xbow.com.

Dust Networks, Inc. (n.d.). http://www.dustnetworks.com.

Expo 2010 Shanghai China (n.d.). http://www.expo2010.cn.

Hill, J. L. (2003). *System Architecture for Wireless Sensor Networks*, PhD thesis, Doctor of Philosophy in Computer Science, University of California at Berkeley, USA.

Howitt, I., Manges, W. W., Kuruganti, P. T., Allgood, G., Gutierrez, J. A. & Conrad, J. M. (2006). Wireless industrial sensor networks: Framework for qos assessment and qos management, *ISA Transactions* **45**(3): 347–359.

IEEE 802.15 WPAN Task Group 4 (n.d.). http://www.ieee802.org/15/pub/TG4.html.

IPSO Alliance - promoting the use of IP for Smart Objects (n.d.). http://www.ipso-alliance.org.

Kumar, S. & Shepherd, D. (2001). Sensit: Sensor information technology for the warfighter, *Proc. of the 4th International Conference on Information Fusion (FUSION'01)*, pp. 3–9 (TuC1).

LiteOS (n.d.). http://www.liteos.net.

Martinez, K., Padhy, P., Riddoch, A., Ong, H. L. R. & Hart, J. K. (2005). Glacial environment monitoring using sensor networks, *Proc. of Workshop on Real-World Wireless Sensor Networks (REALWSN'05)*.

Myers, C., Oppenheim, A., Davis, R. & Dove, W. (1984). Knowledge-based speech analysis and enhancement, *Proc. of the International Conference on Acoustics, Speech and Signal Processing*.

Ni, L. M. (2008). China's national research project on wireless sensor networks, *Proc. of the 2008 IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC'08)*, p. 19.

Pister, K. S. J. (2000). Military applications of sensor networks, *of Institute for Defense Analyses Paper P-3531, Defense Science Study Group*.

Proceedings of the Distributed Sensor Nets Workshop (1978). Pittsburgh, USA. Department of Computer Science, Carnegie Mellon University.

Rashid, R. & Robertson, G. (1981). Accent: A communication oriented network operating system kernel, *Proc. of the 8th Symposium on Operating System Principles*, pp. 64–75.

Rohrback Cosasco Systems (n.d.). http://www.cosasco.com.

Sabbah, E., Majeed, A., Kang, K., Liu, K. & AbuGhazaleh, N. (2006). An application driven perspective on wireless sensor network security, *Proc. of the 2nd ACM Workshop on QoS and Security for Wireless and Mobile Networks*.

SOS Embedded Operating System (n.d.). https://projects.nesl.ucla.edu/public/sos-2x/doc/.

Steere, D., Baptista, A., McNamee, D., Pu, C. & Walpole, J. (2000). Research challenges in environmental observation and forecasting systems, *Proc. of 6th International Conference on Mobile Computing and Networking (MOBICOMM'00)*, pp. 292–299.

Streetline, Inc. (n.d.). http://www.streetlinenetworks.com.

Sudevalayam, S. & Kulkarni, P. (2008). Energy Harvesting Sensor Nodes: Survey and Implications, *Technical Report TR-CSE-2008-19*, Department of Computer Science and Engineering, Indian Institute of Technology Bombay.

TinyOS Community Forum (n.d.). http://www.tinyos.net.

Wang, Q. & Zhang, T. (2008). Sec-snmp: Policy-based security management for sensor networks, *Proc. of the International Conference on Security and Cryptography (SECRYPT'08), in conjunction with ICETE 2008*.

Wang, Q. & Zhang, T. (2009). A survey on security in wireless sensor networks, *in* Y. Zhang & P. Kitsos (eds), *Security in RFID and Sensor Networks*, CRC Press, Taylor & Francis Group, chapter 14, pp. 293–320.

ZigBee Alliance (n.d.). http://www.zigbee.org.

Type	IEEE 802.11b/g	Bluetooth	UWB-IR	IEEE 802.15.4	Zigbee	Wireless HART	IEEE 802.15.6[a]
Band	ISM 2.4GHz	ISM 2.4GHz	3.1GHz-10.6GHz	ISM 2.4GHz, 915MHz, 868MHz	ISM 2.4GHz	ISM 2.4GHz	400MHz, ISM 2.4GHz, 3.1-10.6GHz
Spreading	DSSS	FHSS/TDD	Baseband	DSSS	DSSS	DSSS	DSSS
Modulation	BPSK, QPSK, CCK, OFDM	GFSK	Impulse radio, time-domain	O-QPSK	O-QPSK	O-QPSK	Group PPM
Range	100m	10m	<5m	10m	10m	<250m	3-10m
Rate	<54Mbps	1Mbps	20Kbps, 250Kbps, 10Mbps	250Kbps	250Kbps	250Kbps	<10Mbps
Power	High	Low	Ultra low	Very low	Very low	Battery or line power	Ultra low
Roaming	Yes	No	Yes	Yes	Yes	Yes	Yes
No. of nodes	32 (per access point)	8 (per piconet)	10-1000	<65536	<65536	5-65536	<256
Power consumption	Medium	Low	Ultra low	Very low	Very low	Very low	Ultra low
Complexity	Complex	Very complex	Simple transmitter but complex receiver	Simple	Simple	Simple	Simple
Security	WEP, WPA	64bit or 128 bit	128 bit	NULL, 32bit, 64bit or 128bit	128 bit	128 bit	Scrambled mapping code
Target cost	High	Medium	Very low	Low	Low	Low	Low

Table 1. Comparisons of different standards available for the communication protocols of WSNs.

[a] All values in this column are not official. IEEE 802.15.6 is a communication standard optimized for wireless body area networks and it is still under active development.