3GE Collection on Computer Science:

Wireless and Sensor Systems

BIBLIOTEX
Digital Library

3GE COLLECTION ON COMPUTER SCIENCE:

WIRELESS AND SENSOR SYSTEMS

BIBLIOTEX
Digital Library

www.bibliotex.com

3GE COLLECTION ON COMPUTER SCIENCE:
WIRELESS AND SENSOR SYSTEMS

BIBLIOTEX
Digital Library

www.bibliotex.com

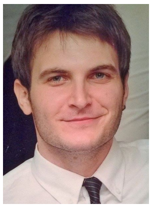
email: info@bibliotex.com

e-book Edition 2022

ISBN: 978-1-98467-703-7 (e-book)

In Collaboration with 3G E-Learning LLC. Originally Published in printed book format by 3G E-Learning LLC with ISBN 978-1-98465-996-5

EDITORIAL BOARD



Aleksandar Mratinković is born on May 5, 1988. in Arandjelovac, Serbia. He has graduated on Economic high school (2007), The College of Tourism in Belgrade (2013), and also has a master degree of Psychology (Faculty of Philosophy, University of Novi Sad). He has been engaged in different fields of psychology (Developmental Psychology, Clinical Psychology, Educational Psychology and Industrial Psychology) and has published several scientific works.

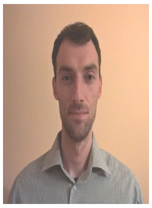
Dan Piestun (PhD) is currently a startup entrepreneur in Israel working on the interface of Agriculture and Biomedical Sciences and was formerly president-CEO of the National Institute of Agricultural Research (INIA) in Uruguay. Dan is a widely published scientist who has received many honours during his career including being a two-time recipient of the Amit Golda Meir Prize from the Hebrew University of Jerusalem, his areas of expertise includes stem cell molecular biology, plant and animal genetics and bioinformatics. Dan's passion for applied science and technological solutions did not stop him from pursuing a deep connection to the farmer, his family and nature. Among some of his interest and practices counts enjoying working as a beekeeper and onboard fishing.

Felecia Killings is the Founder and CEO of LiyahAmore Publishing, a publishing company committed to providing technical and educational services and products to Christian Authors. She operates as the Senior Editor and Writer, the Senior Writing Coach, the Content Marketing Specialist, Editor-in-Chief to the company's quarterly magazine, the Executive and Host of an international virtual network, and the Executive Director of the company's online school for Authors. She is a former high-school English instructor and professional development professor. She possesses a Master of Arts degree in Education and a Bachelor's degree in English and African American studies.

Dr. Sandra El Hajj, Ph.D. in Health Sciences from Nova Southeastern University, Florida, USA is a health professional specialized in Preventive and Global Health. With her 12 years of education obtained from one of the most prominent universities in Beirut, in addition to two leading universities in the State of Florida (USA), Dr. Sandra made sure to incorporate interdisciplinary and multicultural approaches in her work. Her long years of studies helped her create her own miniature world of knowledge linking together the healthcare field with Medical Research, Statistics, Food Technology, Environmental & Occupational Health, Preventive Health and most noteworthy her precious last degree of Global Health. Till today, she is the first and only doctor specialized in Global Health in the Middle East area.

Hazem Shawky Fouda has a PhD. in Agriculture Sciences, obtained his PhD. From the Faculty of Agriculture, Alexandria University in 2008, He is working in Cotton Arbitration & Testing General Organization (CATGO).

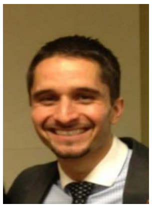
Fozia Parveen has a Dphil in Sustainable Water Engineering from the University of Oxford. Prior to this she has received MS in Environmental Sciences from National University of Science and Technology (NUST), Islamabad Pakistan and BS in Environmental Sciences from Fatima Jinnah Women University (FJWU), Rawalpindi.



Igor Krunic 2003-2007 in the School of Economics. After graduating in 2007, he went on to study at The College of Tourism, at the University of Belgrade where he got his bachelor degree in 2010. He was active as a third-year student representative in the student parliament.Then he went on the Faculty of science, at the University of Novi Sad where he successfully defended his master's thesis in 2013. The crown of his study was the work titled Opportunities for development of cultural tourism in Cacak". Later on, he became part of a multinational company where he got promoted to a deputy director of logistic. Nowadays he is a consultant and writer of academic subjects in the field of tourism.

Dr. Jovan Pehcevski obtained his PhD in Computer Science from RMIT University in Melbourne, Australia in 2007. His research interests include big data, business intelligence and predictive analytics, data and information science, information retrieval, XML, web services and service-oriented architectures, and relational and NoSQL database systems. He has published over 30 journal and conference papers and he also serves as a journal and conference reviewer. He is currently working as a Dean and Associate Professor at European University in Skopje, Macedonia.

Dr. Tanjina Nur finished her PhD in Civil and Environmental Engineering in 2014 from University of Technology Sydney (UTS). Now she is working as Post-Doctoral Researcher in the Centre for Technology in Water and Wastewater (CTWW) and published about eight International journal papers with 80 citations. Her research interest is wastewater treatment technology using adsorption process.



Stephen obtained his PhD from the University of North Carolina at Charlotte in 2013 where his graduate research focused on cancer immunology and the tumor microenvironment. He received postdoctoral training in regenerative and translational medicine, specifically gastrointestinal tissue engineering, at the Wake Forest Institute of Regenerative Medicine. Currently, Stephen is an instructor for anatomy and physiology and biology at Forsyth Technical Community College.

Michelle holds a Masters of Business Administration from the University of Phoenix, with a concentration in Human Resources Management. She is a professional author and has had numerous articles published in the Henry County Times and has written and revised several employee handbooks for various YMCA organizations throughout the United States.

Table of Contents

Preface

Wireless sensor technology has been recognized as one of the emerging technologies of this century widely used for intelligent data sensing. WSNs have become an integral part of diverse applications such as environmental monitoring, military surveillance, and medicine by providing feasible communication, reliable inspection, and performing applications. WSNs are composed of a large number of sensor nodes which are densely deployed and wirelessly communicated to send and receive environmental information. A wireless sensor network (WSN) is composed of several sensor nodes, where the main objective of a sensor node is to collect information from its surrounding environment and transmit it to one or more points of centralized control, called base stations or sinks, for further analysis and processing. With the development of network and communication technology, the inconvenience of wiring is solved with WSN into people's life; especially it has wide perspective and practicability in the area of remote sensing, industrial automation control, and domestic appliance and so on. WSN has good functions of data collection, transmission, and processing. It has many advantages compared to traditional wired network, for example, convenient organizing network, small influence to environment, low power dissipation, low cost, etc. At present, near field wireless communication technology has been used widely, especially Bluetooth, wireless local area network (WLAN), infrared, etc.

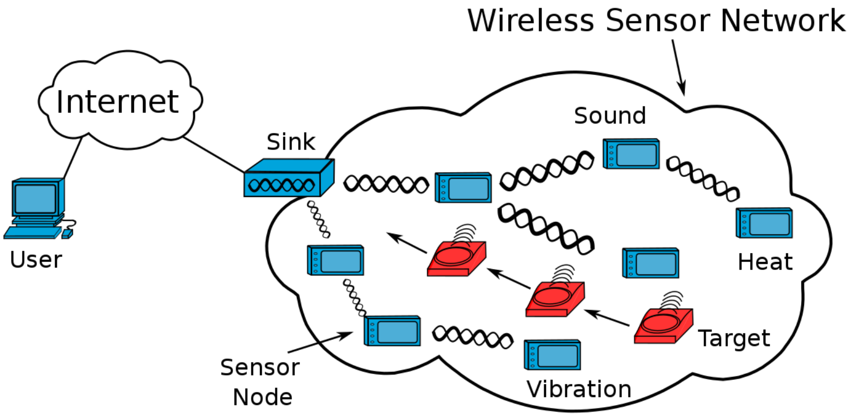
A complete overview of wireless sensor network technology is given in this book. Wireless sensor network technology has become one of technological basic needs of us. Wireless sensor networks (WSNs) have grown considerably in recent years and have a significant potential in different applications including health, environment, and military. Despite their powerful capabilities, the successful development of WSN is still a challenging task. In current real-world WSN deployments, several programming approaches have been proposed, which focus on low-level system issues. In order to simplify the design of the WSN and abstract from technical low-level details, high-level approaches have been recognized and several solutions have been proposed. The book explores many fields such as wireless networks and communications, protocols, distributed algorithms, signal processing, embedded systems, and information management.

CHAPTER 1

INTRODUCTION WIRELESS AND SENSOR SYSTEMS

INTRODUCTION

Wireless sensor networks (WSNs) refer to networks of spatially dispersed and dedicated sensors that monitor and record the physical conditions of the environment and forward the collected data to a central location. WSNs can measure environmental conditions such as temperature, sound, pollution levels, humidity and wind.

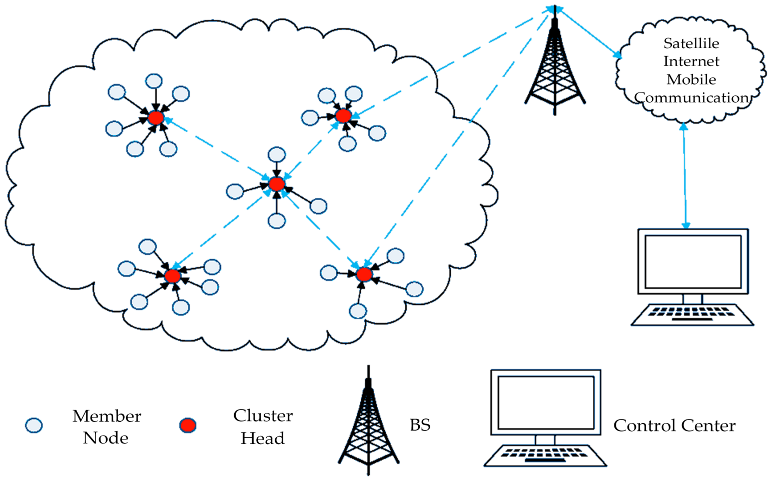


These are similar to wireless ad hoc networks in the sense that they rely on wireless connectivity and spontaneous formation of networks so that sensor data can be transported wirelessly. WSNs monitor physical or environmental conditions, such as temperature, sound, and pressure. Modern networks are bi-directional, both collecting data and enabling control of sensor activity. The development of these networks was motivated by military applications such as battlefield surveillance. Such networks are used in industrial and consumer applications, such as industrial process monitoring and control and machine health monitoring.

1.1 OVERVIEW OF WIRELESS SENSOR NETWORK

Wireless Sensor Networks (WSNs) can be defined as a self-configured and infrastructure-less wireless networks to monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutants and to cooperatively pass their data through the network to a main location or sink where the data can be observed and analyzed. A sink or base station acts like an interface between users and the network. One can retrieve required information from the network by injecting queries and gathering results from the sink. Typically a wireless sensor network contains hundreds of thousands of sensor nodes. The sensor nodes can communicate among themselves using radio signals. A wireless sensor node is equipped with sensing and computing devices, radio transceivers and power components. The individual nodes in a wireless sensor network (WSN) are inherently resource constrained: they have limited processing speed, storage capacity, and communication bandwidth. After the sensor nodes are deployed, they are responsible for self-organizing an appropriate network infrastructure often with multi-hop communication with them. Then the onboard sensors start collecting information of interest. Wireless sensor devices also respond to queries sent from a “control site” to perform specific instructions or provide sensing samples. The working mode of the sensor nodes may be either continuous or event driven. Global Positioning System (GPS) and local positioning algorithms can be used to obtain location and positioning information. Wireless sensor devices can be equipped with actuators to “act” upon certain conditions.

Wireless sensor networks (WSNs) enable new applications and require non-conventional paradigms for protocol design due to several constraints. Owing to the requirement for low device complexity together with low energy consumption (i.e. long network lifetime), a proper balance between communication and signal/data processing capabilities must be found. This motivates a huge effort in research activities, standardization process, and industrial investments on this field since the last decade.



At present time, most of the research on WSNs has concentrated on the design of energy- and computationally efficient algorithms and protocols, and the application domain has been restricted to simple data-oriented monitoring and reporting applications. Specifically, it allocates periods of inactivity for cable sensors without affecting the coverage and connectivity requirements of the network based only on local information. A delay-aware data collection network structure for wireless sensor networks is proposed. The objective of the proposed network structure is to minimize delays in the data collection processes of wireless sensor networks which extends the lifetime of the network. Most of the time, the research on wireless sensor networks have considered homogeneous sensor nodes. But nowadays researchers have focused on heterogeneous sensor networks where the sensor nodes are unlike to each other in terms of their energy. New network architectures with heterogeneous devices and the recent advancement in this technology eliminate the current limitations and expand the spectrum of possible applications for WSNs considerably and all these are changing very rapidly.

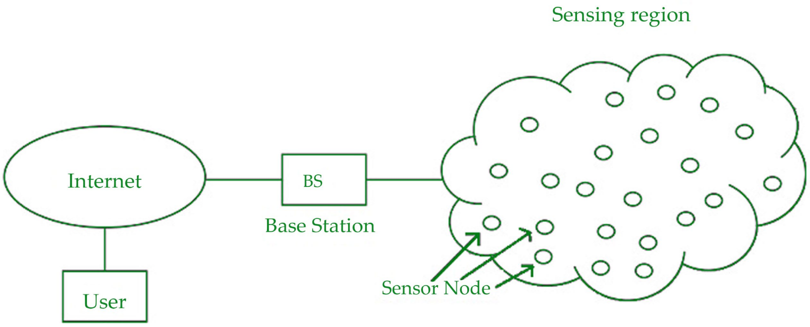


Figure 1. A typical Wireless Sensor Network.

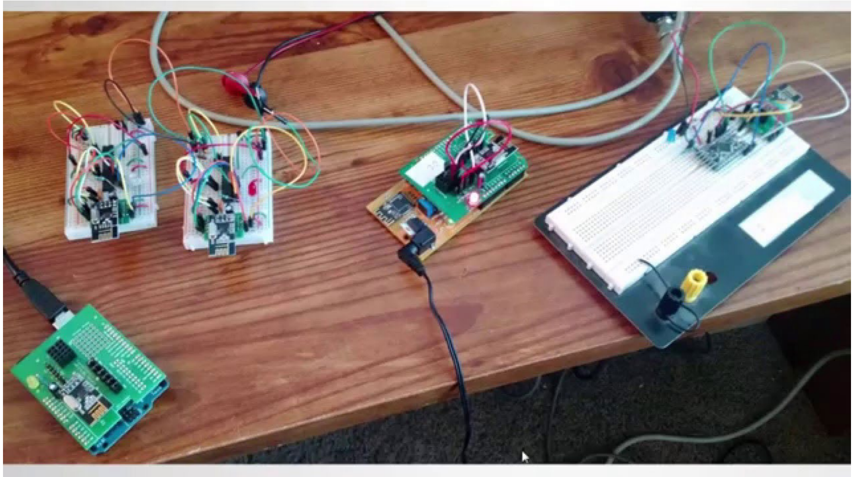
1.1.1 Applications of Wireless Sensor Network

Wireless sensor networks have gained considerable popularity due to their flexibility in solving problems in different application domains and have the potential to change our lives in many different ways. WSNs have been successfully applied in various application domains.

Military applications: Wireless sensor networks be likely an integral part of military command, control, communications, computing, intelligence, battlefield surveillance, reconnaissance and targeting systems.

Area monitoring: In area monitoring, the sensor nodes are deployed over a region where some phenomenon is to be monitored. When the sensors detect the event being monitored (heat, pressure etc), the event is reported to one of the base stations, which then takes appropriate action.

Transportation: Real-time traffic information is being collected by WSNs to later feed transportation models and alert drivers of congestion and traffic problems.



Health applications: Some of the health applications for sensor networks are supporting interfaces for the disabled, integrated patient monitoring, diagnostics, and drug administration in hospitals, tele-monitoring of human physiological data, and tracking & monitoring doctors or patients inside a hospital.

Environmental sensing: The term Environmental Sensor Networks has developed to cover many applications of WSNs to earth science research. This includes sensing volcanoes, oceans, glaciers, forests etc. Some other major areas are listed below:

- Air pollution monitoring

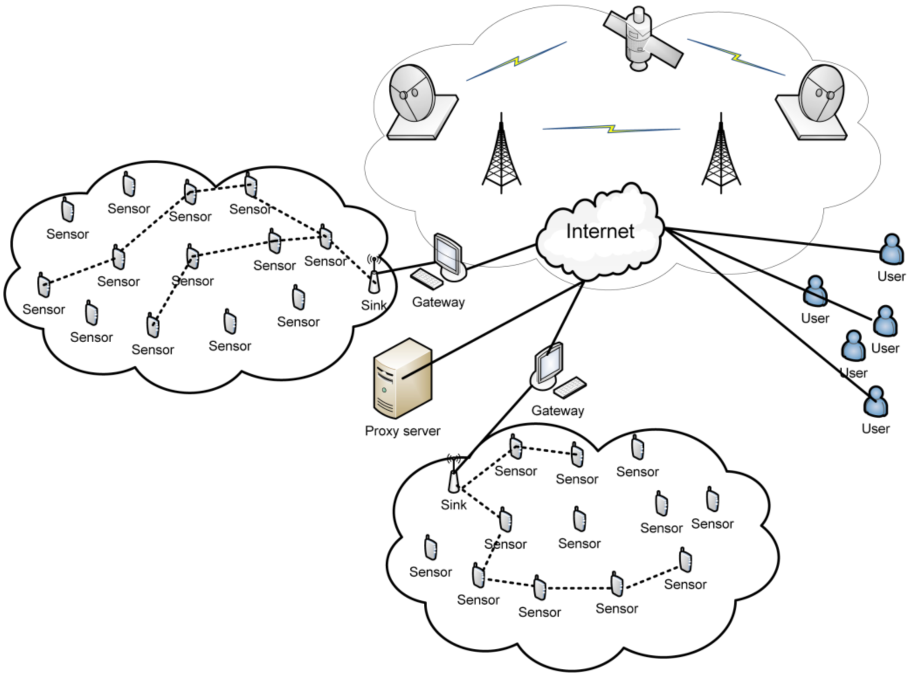
- Forest fires detection
- Greenhouse monitoring
- Landslide detection

Structural monitoring: Wireless sensors can be utilized to monitor the movement within buildings and infrastructure such as bridges, flyovers, embankments, tunnels etc enabling Engineering practices to monitor assets remotely with out the need for costly site visits.

Industrial monitoring: Wireless sensor networks have been developed for machinery condition-based maintenance (CBM) as they offer significant cost savings and enable new functionalities. In wired systems, the installation of enough sensors is often limited by the cost of wiring.

Agricultural sector: using a wireless network frees the farmer from the maintenance of wiring in a difficult environment. Irrigation automation enables more efficient water use and reduces waste.

These networks are used in environmental tracking, such as forest detection, animal tracking, flood detection, forecasting, and weather prediction, and also in commercial applications like seismic activity prediction and monitoring.



Military applications, such as tracking and environment monitoring surveillance applications use these networks. The sensor nodes from sensor networks are dropped to the field of interest and are remotely controlled by a user. Enemy tracking, security detections are also performed by using these networks.

Health applications, such as Tracking and monitoring of patients and doctors use these networks.

The most frequently used wireless sensor network applications in the field of Transport systems such as monitoring of traffic, dynamic routing management, and monitoring of parking lots, etc., use these networks.

Rapid emergency response, industrial process monitoring, automated building climate control, ecosystem and habitat monitoring, civil structural health monitoring, etc., use these networks.

1.1.2 Design issues of a Wireless Sensor Network

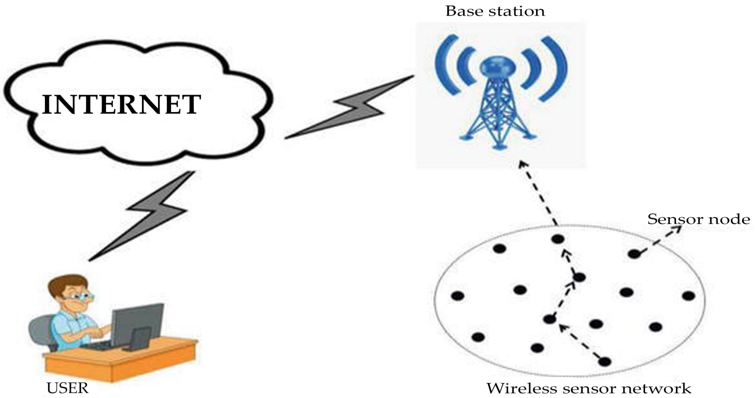
There are a lot of challenges placed by the deployment of sensor networks which are a superset of those found in wireless ad hoc networks. Sensor nodes communicate over wireless, lossy lines with no infrastructure. An additional challenge is related to the limited, usually non-renewable energy supply of the sensor nodes. In order to maximize the lifetime of the network, the protocols need to be designed from the beginning with the objective of efficient management of the energy resources.

Fault Tolerance: Sensor nodes are vulnerable and frequently deployed in dangerous environment. Nodes can fail due to hardware problems or physical damage or by exhausting their energy supply. We expect the node failures to be much higher than the one normally considered in wired or infrastructure-based wireless networks. The protocols deployed in a sensor network should be able to detect these failures as soon as possible and be robust enough to handle a relatively large number of failures while maintaining the overall functionality of the network. This is especially relevant to the routing protocol design, which has to ensure that alternate paths are available for rerouting of the packets. Different deployment environments pose different fault tolerance requirements.

Scalability: Sensor networks vary in scale from several nodes to potentially several hundred thousand. In addition, the deployment density is also variable. For collecting high-resolution data, the node density might reach the level where a node has several thousand neighbors in their transmission range. The protocols deployed in sensor networks need to be scalable to these levels and be able to maintain adequate performance.

Production Costs: Because many deployment models consider the sensor nodes to be disposable devices, sensor networks can compete with traditional information gathering approaches only if the individual sensor nodes can be produced very cheaply. The target price envisioned for a sensor node should ideally be less than $1.

Hardware Constraints: At minimum, every sensor node needs to have a sensing unit, a processing unit, a transmission unit, and a power supply. Optionally, the nodes may have several built-in sensors or additional devices such as a localization system to enable location-aware routing. However, every additional functionality comes with additional cost and increases the power consumption and physical size of the node. Thus, additional functionality needs to be always balanced against cost and low-power requirements.



Sensor Network Topology: Although WSNs have evolved in many aspects, they continue to be networks with constrained resources in terms of energy, computing power, memory, and communications capabilities. Of these constraints, energy consumption is of paramount importance, which is demonstrated by the large number of algorithms, techniques, and protocols that have been developed to save energy, and thereby extend the lifetime of the network. Topology Maintenance is one of the most important issues researched to reduce energy consumption in wireless sensor networks.

Transmission Media: The communication between the nodes is normally implemented using radio communication over the popular ISM bands. However, some sensor networks use optical or infrared communication, with the latter having the advantage of being robust and virtually interference free.

Power Consumption: As we have already seen, many of the challenges of sensor networks revolve around the limited power resources. The size of the nodes limits the size of the battery. The software and hardware design needs to carefully consider the issues of efficient energy use. For instance, data compression might reduce the amount of energy used for radio transmission, but uses additional energy for computation and/or filtering. The energy policy also depends on the application; in some applications, it might be acceptable to turn off a subset of nodes in order to conserve energy while other applications require all nodes operating simultaneously.

1.1.3 Structure of a wireless sensor network

Structure of a Wireless Sensor Network includes different topologies for radio communications networks. A short discussion of the network topologies that apply to wireless sensor networks are outlined below:

Star network (single point-to-multipoint)

A star network is a communications topology where a single base station can send and/or receive a message to a number of remote nodes. The remote nodes are not permitted to send messages to each other. The advantage of this type of network for wireless sensor networks includes simplicity, ability to keep the remote node's power consumption to a minimum. It also allows low latency communications between the remote node and the base station. The disadvantage of such a network is that the base station must be within radio transmission range of all the individual nodes and is not as robust as other networks due to its dependency on a single node to manage the network.

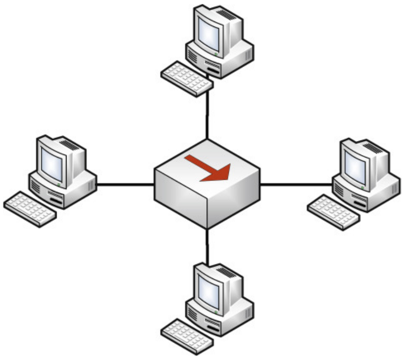


Figure 2. A Star network topology.

Mesh network

A mesh network allows transmitting data to one node to other node in the network that is within its radio transmission range. This allows for what is known as multi-hop communications, that is, if a node wants to send a message to another node that is out of radio communications range, it can use an intermediate node to forward the message to the desired node. This network topology has the advantage of redundancy and scalability. If an individual node fails, a remote node still can communicate to any other node in its range, which in turn, can forward the message to the desired location. In addition, the range of the network is not necessarily limited by the range in between single nodes; it can simply be extended by adding more nodes to the system. The disadvantage of this type of network is in power consumption for the nodes that implement the multi-hop communications are generally higher than for the nodes that don't have this capability, often limiting the battery life. Additionally, as the number of communication hops to a destination increases, the time to deliver the message also increases, especially if low power operation of the nodes is a requirement.

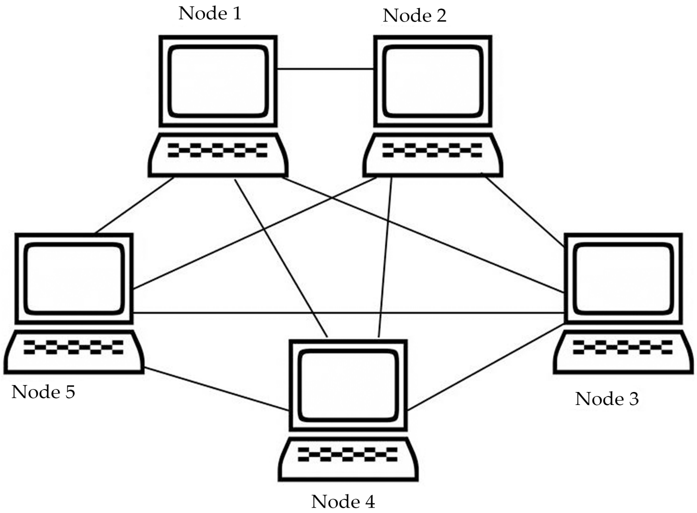


Figure 3. A Mesh network topology.

Hybrid star – Mesh network

A hybrid between the star and mesh network provides a robust and versatile communications network, while maintaining the ability to keep the wireless sensor nodes power consumption to a minimum. In this network topology, the sensor nodes with lowest power are not enabled with the ability to forward messages. This allows for minimal power consumption to be maintained. However, other nodes on the network are enabled with multi-hop capability, allowing them to forward messages from the low power nodes to other nodes on the network. Generally, the nodes with the multi-hop capability are higher power, and if possible, are often plugged into the electrical mains line. This is the topology implemented by the up and coming mesh networking standard known as ZigBee.

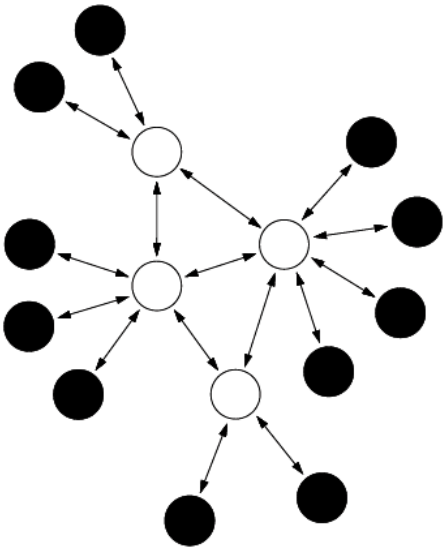
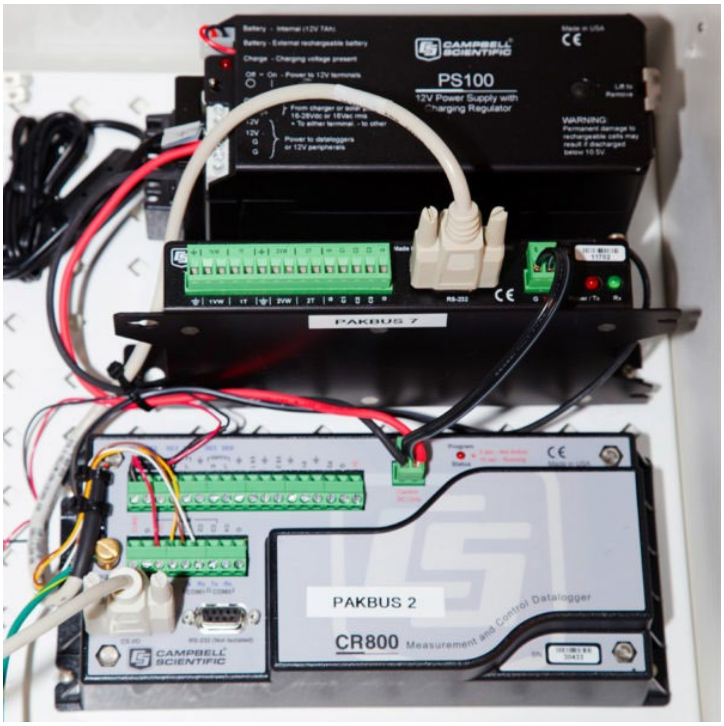


Figure 4. A Hybrid Star – Mesh network topology.

1.1.4 Structure of a Wireless Sensor Node

A sensor node is made up of four basic components such as sensing unit, processing unit, transceiver unit and a power unit which is shown in Fig. 5. It also has application dependent additional components such as a location finding system, a power generator and a mobilizer. Sensing units are usually composed of two subunits: sensors and analogue to digital converters (ADCs). The analogue signals produced by the sensors are converted to digital signals by the ADC, and then fed into the processing unit. The processing unit is generally associated with a small storage unit and it can manage the procedures that make the sensor node collaborate with the other nodes to carry out the assigned sensing tasks. A transceiver unit connects the node to the network. One of the most important components of a sensor node is the power unit. Power units can be supported by a power scavenging unit such as solar cells. The other subunits, of the node are application dependent.


CAMPBELL SCIENTIFIC
PS100
12V Power Supply with Charging Regulator
PAKBUS 7
PAKBUS 2
CR800 Measurement and Control Datalogger
CAMPBELL SCIENTIFIC

A functional block diagram of a versatile wireless sensing node is provided in Fig. 6. Modular design approach provides a flexible and versatile platform to address the needs of a wide variety of applications. For example, depending on the sensors to be deployed, the signal conditioning block can be re-programmed or replaced. This allows for a wide variety of different sensors to be used with the wireless sensing node. Similarly, the radio link may be swapped out as required for a given applications' wireless range requirement and the need for bidirectional communications.

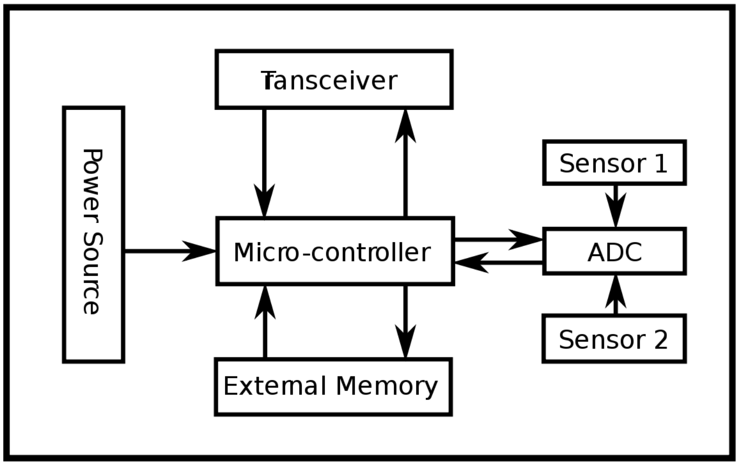


Figure 5. The components of a sensor node.

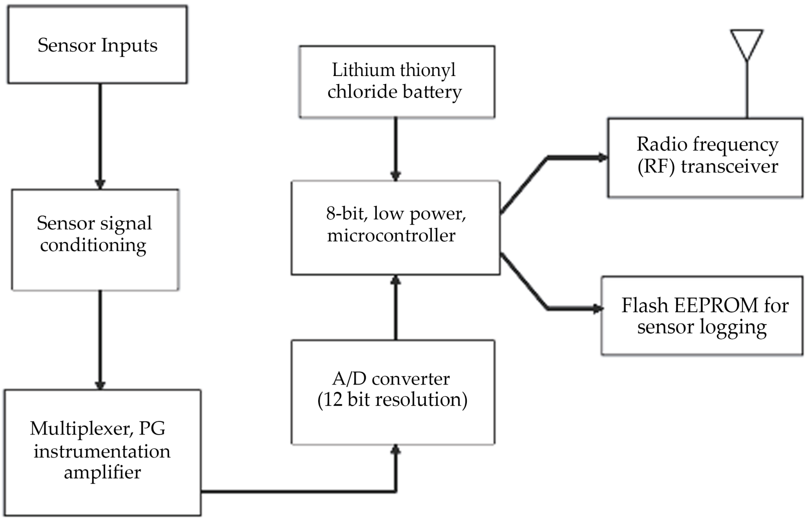

Sensor Inputs
Sensor signal conditioning
Multiplexer, PG instrumentation amplifier
Lithium thionyl chloride battery
8-bit, low power, microcontroller
A/D converter (12 bit resolution)
Radio frequency (RF) transceiver
Flash EEPROM for sensor logging

Figure 6. Functional block diagram of a sensor node.

Using flash memory, the remote nodes acquire data on command from a base station, or by an event sensed by one or more inputs to the node. Moreover, the embedded firmware can be upgraded through the wireless network in the field.

The microprocessor has a number of functions including:

- Managing data collection from the sensors
- performing power management functions

- interfacing the sensor data to the physical radio layer
- managing the radio network protocol

A key aspect of any wireless sensing node is to minimize the power consumed by the system. Usually, the radio subsystem requires the largest amount of power. Therefore, data is sent over the radio network only when it is required. An algorithm is to be loaded into the node to determine when to send data based on the sensed event. Furthermore, it is important to minimize the power consumed by the sensor itself. Therefore, the hardware should be designed to allow the microprocessor to judiciously control power to the radio, sensor, and sensor signal conditioner.

1.1.5 Communication Structure of a Wireless Sensor Network

The sensor nodes are usually scattered in a sensor field as shown in Fig. 1. Each of these scattered sensor nodes has the capabilities to collect data and route data back to the sink and the end users. Data are routed back to the end user by a multi-hop infrastructure-less architecture through the sink as shown in Fig. 1. The sink may communicate with the task manager node via Internet or Satellite.

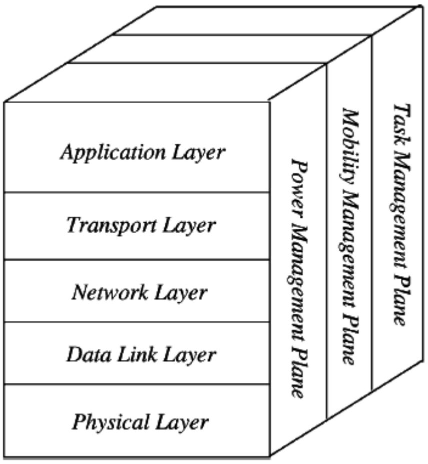


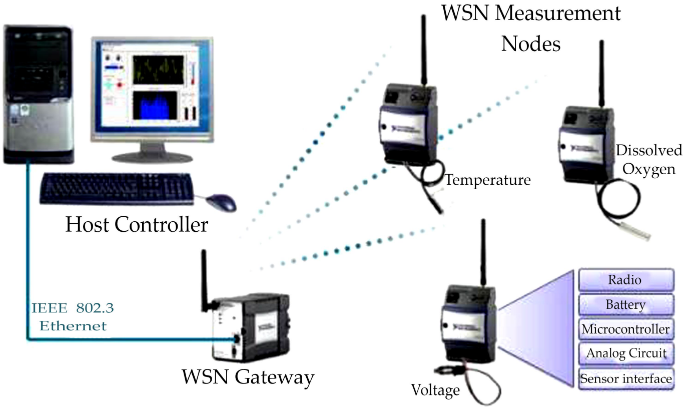
Figure 7. Wireless Sensor Network protocol stack.

The protocol stack used by the sink and the sensor nodes is given in Fig. 7. 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 application layer, transport layer, network layer, data link layer, physical layer, power management plane, mobility management plane, and task management plane. Different types of application software can be built and used on the application layer depending on the sensing tasks. This layer makes hardware and software of the lowest layer transparent to the end-user. The transport layer helps to maintain the flow of data if the sensor networks application requires it. The network layer takes care of routing the data supplied by the transport layer, specific multi-hop wireless routing protocols between sensor nodes and sink. The data link layer is responsible for multiplexing of data streams, frame detection, Media Access Control (MAC) and error control. Since the environment is noisy and sensor nodes can be mobile, the MAC protocol must be power aware and able to minimize collision with neighbors' broadcast. The physical layer addresses the needs of a simple but robust modulation, frequency selection, data encryption, transmission and receiving techniques.

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 energy consumption.

1.1.6 Energy Consumption issues in Wireless Sensor Network

Energy consumption is the most important factor to determine the life of a sensor network because usually sensor nodes are driven by battery. Sometimes energy optimization is more complicated in sensor networks because it involved not only reduction of energy consumption but also prolonging the life of the network as much as possible. The optimization can be done by having energy awareness in every aspect of design and operation. This ensures that energy awareness is also incorporated into groups of communicating sensor nodes and the entire network and not only in the individual nodes.



A sensor node usually consists of four sub-systems:

- a computing subsystem : It consists of a microprocessor(microcontroller unit, MCU) which is responsible for the control of the sensors and implementation of communication protocols. MCUs usually operate under various modes for power management purposes. As these operating modes involves consumption of power, the energy consumption levels of the various modes should be considered while looking at the battery lifetime of each node.
- a communication subsystem: It consists of a short range radio which communicate with neighboring nodes and the outside world. Radios can operate under the different modes. It is important to completely shut down the radio rather than putting it in the Idle mode when it is not transmitting or receiving for saving power.
- a sensing subsystem : It consists of a group of sensors and actuators and link the node to the outside world. Energy consumption can be reduced by using low power components and saving power at the cost of performance which is not required.
- a power supply subsystem : It consists of a battery which supplies power to the node. It should be seen that the amount of power drawn from a battery is checked because if high current is drawn from a battery for a long time, the battery will die faster even though it could have gone on for a longer time. Usually the rated current capacity of a battery being used for a sensor node is less than the minimum energy consumption. The lifetime of a battery can be increased by reducing the current drastically or even turning it off often.

To minimize the overall energy consumption of the sensor network, different types of protocols and algorithms have been studied so far all over the world. The lifetime of a sensor network can be increased significantly if the operating system, the application layer and the network protocols are designed to be energy aware. These protocols and algorithms have to be aware of the hardware and able to use special features of the micro-processors and transceivers to minimize the sensor node's energy consumption. This may push toward a custom solution for different types of sensor node design. Different types of sensor nodes deployed also lead to different types of sensor networks. This may also lead to the different types of collaborative algorithms in wireless sensor networks arena.

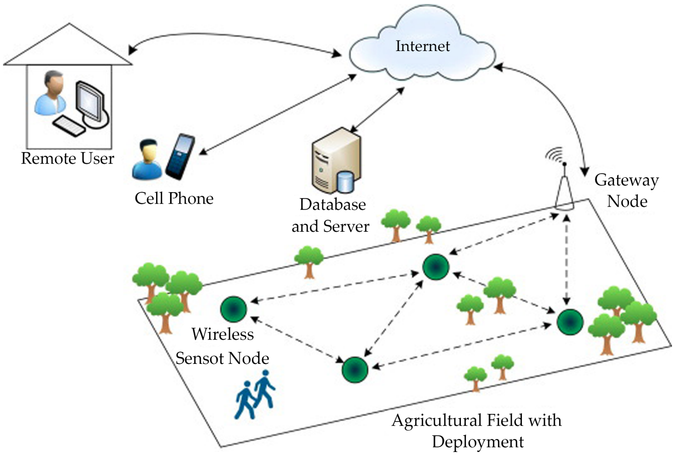
1.2 TYPES AND CLASSIFICATION OF WIRELESS SENSOR NETWORKS

Depending on the environment, the types of networks are decided so that those can be deployed underwater, underground, on land, and so on. Different types of WSNs include:

- Terrestrial WSNs
- Underground WSNs
- Underwater WSNs
- Multimedia WSNs
- Mobile WSNs

Terrestrial WSNs

Terrestrial WSNs are capable of communicating base stations efficiently, and consist of hundreds to thousands of wireless sensor nodes deployed either in an unstructured (ad hoc) or structured (Pre-planned) manner. In an unstructured mode, the sensor nodes are randomly distributed within the target area that is dropped from a fixed plane. The preplanned or structured mode considers optimal placement, grid placement, and 2D, 3D placement models.



In this WSN, the battery power is limited; however, the battery is equipped with solar cells as a secondary power source. The Energy conservation of these WSNs is achieved by using low duty cycle operations, minimizing delays, and optimal routing, and so on.

Underground WSNs

The underground wireless sensor networks are more expensive than the terrestrial WSNs in terms of deployment, maintenance, and equipment cost considerations and careful planning. The WSNs networks consist of several sensor nodes that are hidden in the ground to monitor underground conditions. To relay information from the sensor nodes to the base station, additional sink nodes are located above the ground.

Underground WSNs

The underground wireless sensor networks deployed into the ground are difficult to recharge. The sensor battery nodes equipped with limited battery power are difficult to recharge. In addition to this, the underground environment makes wireless communication a challenge due to the high level of attenuation and signal loss.

Under Water WSNs

More than 70% of the earth is occupied with water. These networks consist of several sensor nodes and vehicles deployed underwater. Autonomous underwater vehicles are used for gathering data from these sensor nodes. A challenge of underwater communication is a long propagation delay, and bandwidth and sensor failures.

Under Water WSNs

Underwater, WSNs are equipped with a limited battery that cannot be recharged or replaced. The issue of energy conservation for underwater WSNs involves the development of underwater communication and networking techniques.

Multimedia WSNs

Multimedia wireless sensor networks have been proposed to enable tracking and monitoring of events in the form of multimedia, such as imaging, video, and audio. These networks consist of low-cost sensor nodes equipped with microphones and cameras. These nodes are interconnected with each other over a wireless connection for data compression, data retrieval, and correlation.

Multimedia WSNs

The challenges with the multimedia WSN include high energy consumption, high bandwidth requirements, data processing, and compressing techniques. In addition to this, multimedia contents require high bandwidth for the content to be delivered properly and easily.

Mobile WSNs

These networks consist of a collection of sensor nodes that can be moved on their own and can be interacted with the physical environment. The mobile nodes can compute sense and communicate.

Mobile wireless sensor networks are much more versatile than static sensor networks. The advantages of MWSN over static wireless sensor networks include better and improved coverage, better energy efficiency, superior channel capacity, and so on.

1.2.1 Classification of Wireless Sensor Networks

The classification of WSNs can be done based on the application but its characteristics mainly change based on the type. Generally, WSNs are classified into different categories like the following.

- Static & Mobile
- Deterministic & Nondeterministic

- Single Base Station & Multi Base Station
- Static Base Station & Mobile Base Station
- Single-hop & Multi-hop WSN
- Self Reconfigurable & Non-Self Configurable
- Homogeneous & Heterogeneous

Static & Mobile WSN

All the sensor nodes in several applications can be set without movement so these networks are static WSNs. Especially in some applications like biological systems uses mobile sensor nodes which are called mobile networks. The best example of a mobile network is the monitoring of animals.

Deterministic & Nondeterministic WSN

In a deterministic type of network, the sensor node arrangement can be fixed and calculated. This sensor node's pre-planned operation can be possible in simply some applications. In most applications, the location of sensor nodes cannot be determined because of the different factors like hostile operating conditions & harsh environment, so these networks are called non-deterministic that need a complex control system.

Single Base Station & Multi Base Station

In a single base station network, a single base station is used and it can be arranged very close to the region of the sensor node. The interaction between sensor nodes can be done through the base station. In a multi-base station type network, multiple base stations are used & a sensor node is used to move data toward the nearby base station.

Static Base Station & Mobile Base Station

Base stations are either mobile or static similar to sensor nodes. As the name suggests, the static type base station includes a stable position generally close to the sensing area whereas the mobile base station moves in the region of the sensor so that the sensor nodes load can be balanced.

Single-hop & Multi-hop WSN

In a single-hop type network, the arrangement of sensor nodes can be done directly toward the base station whereas, in a multi-hop network, both the cluster heads & peer nodes are utilized to transmit the data to reduce the energy consumption.

Self Reconfigurable & Non-Self Configurable

In a non-self configurable network, the arrangement of sensor networks cannot be done by them within a network & depends on a control unit for gathering data. In wireless sensor networks, the sensor nodes maintain and organize the network and collaboratively work by using other sensor nodes to accomplish the task.

Homogeneous and Heterogeneous

In a homogeneous wireless sensor network, all the sensor nodes mainly include similar energy utilization, storage capabilities & computational power. In the heterogeneous network case, some sensor nodes include high computational power as well as energy necessities as compared to others. The processing & communication tasks are separated consequently.

1.2.2 Types of Attacks in WSNs

There are different types of attacks against wireless sensor networks. These attacks can be faced by a variety of measurements. Attacks are classified into two types the active attacks and passive attacks.

In the active type attack, an attacker attempts to modify or detach the transmitted messages over the network. An attacker can give a reply to old messages and also insert his own traffic to interrupt the network operation otherwise to cause a rejection of service.

The passive attack can be restricted to listening & examining exchanged traffic. So this kind of attack can be easier to recognize & it is complex to notice. As the attacker does not make any change on exchanged data. The goal of the attacker is to get confidential information otherwise the significant nodes data within the network by examining routing data.

- Tampering

- Identity replication attack
- Blackhole
- Wormhole attack
- Selective forwarding
- Exhaustion
- Sybil attack
- Blackmail attack
- HELLO flood attack
- Jamming

1.2.3 Types of Mobility in Wireless Sensor Networks

In ad hoc networks, mobility is a basic feature for all nodes. In WSNs, mobility exists generally to separate the elements of the network & more specifically it depends on the application. Wireless sensor network applications have been involving in different fields but in many fields, there is no involvement of mobility. So mobility plays a key role where wireless sensor networks are used. In WSNs, we can differentiate three different types of mobility like the following.

- Sensor nodes mobility
- Sin nodes mobility
- Monitored object or event mobility

The first type of mobility like sensor nodes mobility mainly occurs whenever the sensor node's slightest element is mobile. The best examples of this type of mobility are once sensor nodes go away & freely move within the monitored area. These are set up on animals for monitoring & tracking of animals.

The second type of mobility refers to a condition where sink nodes are capable of separately moving within the monitored area for collecting information from the sensor network. Lastly, the third type of mobility mainly happens once a wireless sensor network is used for tracking/ monitoring purposes & functions under the event-driven data model.

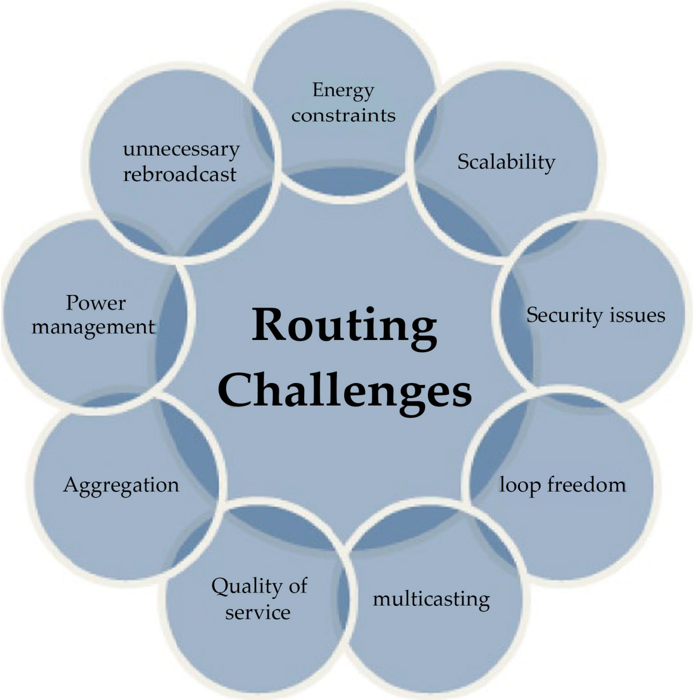
In the same way, once the wireless sensor network is used for tracking target, movement of target modeling is extremely useful for guessing the pattern & amount of produced data within the network throughout tracking the target.

1.2.4 Types of Routing Protocols in Wireless Sensor Networks

The routing protocol can be defined as it is one kind of a process used to choose the appropriate lane for the data to move from basic to end. This process faces numerous difficulties while choosing the route. Here, this route depends on the type of network, the performance metrics & channel characteristics.

Routing Challenges

For WSN, the design task of routing protocols is pretty challenging due to several characteristics which distinguish them from wireless infrastructure with fewer networks. In WSNs, different types of routing challenges are available where some of them are discussed below.



It is approximately complex to assign a universal identifiers system for sensor nodes with high quantity. Thus, wireless sensor nodes are not capable of utilizing protocols based on classical IP. The information which is detected is essential from different sources to a particular base station. However, this does not happen in normal communication networks.

In most cases, the data which is created has important redundancy because several sensing nodes can produce similar data while detecting. So, it is necessary to use such redundancy through the routing protocols, the accessible bandwidth & energy.

Furthermore, wireless nodes are definitely restricted in bandwidth, transmission energy relations, storage, capacity & onboard energy. Because of these dissimilarities, the number of the latest routing protocols have been estimated to handle routing challenges within WSNs.

Design Challenges

There are some main design challenges in WSNs because of a lack of resources like bandwidth, processing storage & energy. When designing the latest routing protocols, then the following basics must be fulfilled through a network engineer.

- Efficiency of Energy
- Location of Sensor
- Complexity
- Transmission of Data & Transmission Models
- Scalability
- Strength
- Delay

1.2.5 Challenges of WSN

The different challenges in wireless sensor networks include the following.

- Fault Performance
- Scalability
- Production Cost
- Operation Environment
- Quality of Service
- Data Aggregation
- Data Compression
- Data Latency

Fault Performance

Some sensor nodes stop working because of power loss, so physical damage may occur. This shouldn't affect the sensor network's overall performance, so this is known as the issue of fault tolerance. Fault tolerance is nothing but the ability to maintain the functionalities of the sensor network without any interruption because of the failures of sensor nodes.

Stability

The number of nodes used in the detecting area may be in the order of thousands, hundreds & routing schemes should be scalable enough for responding to events.

Production Cost

The sensor networks include a number of sensor nodes where a single node price is very significant to validate the cost of the overall network and thus each sensor node's price must be kept low.

Operation Environment

The arrangement of sensor networks can be done within large machinery, under the ocean, in the field of chemically or biologically contaminated. in homes, battlefields, connected to fast-moving vehicles, animals, for monitoring in forests, etc.

Quality of Service

The quality of service which needs by the application could be energy efficiency, lifetime length, and reliable data.

Data Aggregation

The combination of data from various sources with different functions like average, max, min, is known as data aggregation.

Data Compression

The data reduction is known as data compression

Data Latency

These are treated like the essential factors that influence the design of routing protocol. The data latency can be caused through data aggregation & multi-hop relays.

1.2.6 Issues in Wireless Sensor Networks

There are different issues occurred in wireless sensor networks like design issues, topology issues, and other issues. The design issues in different types of wireless sensor networks mainly include

- Low latency
- Fault
- Coverage Problems
- Transmission Media
- Scalability

The topology issues of wireless sensor networks include the following.

- Sensor Holes
- Geographic Routing
- Coverage Topology

The major issues of a wireless sensor network include the following. These issues mainly affect the design and performance of the WSN.

- Operating System & Hardware for WSN
- Middleware
- Characteristics of Wireless Radio Communication
- Schemes for Medium Access
- Deployment
- Localization
- Sensor Networks Programming Models
- Synchronization
- Architecture
- Calibration
- Database Centric and Querying
- Network Layer

- Data Dissemination & Data Aggregation
- Transport Layer

Limitations

The limitations of wireless sensor networks include the following.

- Possess very little storage capacity – a few hundred kilobytes
- Possess modest processing power-8MHz
- Works in short communication range – consumes a lot of power
- Requires minimal energy – constrains protocols
- Have batteries with a finite lifetime
- Passive devices provide little energy

1.3 WIRELESS SENSOR NETWORKS FOR BIG DATA SYSTEMS

When constructing a big data system, data collection, storage, processing, analysis, and visualization are steps that need to be followed in the said order. In ongoing research on big data systems, the research communities focus on fundamental aspects of dealing with big data: specific platforms, technology, beneficial applications, standards, and best practices (for applications in social web, financial issues, and so on). Moreover, there are many platforms and tools that can implement these functions in the real world. Thus, data-intensive applications are now being developed to benefit from them.

A WSN consists of a large number of sensor nodes that monitor and record the physical conditions of an environment, and the sensor data are collected at a so-called sink node. WSNs are used to measure environmental conditions: temperature, sound, pollution levels, humidity, wind, and so on. However, the limited capacity of a single node and a narrow wireless link (compared to typical networks) cause problems with delivering the sensor data to the sink node. Nevertheless, an effective data aggregation and in-network processing are beneficial to big data systems. Therefore, there is a need of analyzing research studies that link WSN and big data systems while overcoming the deficiencies of WSN and improving system performance.

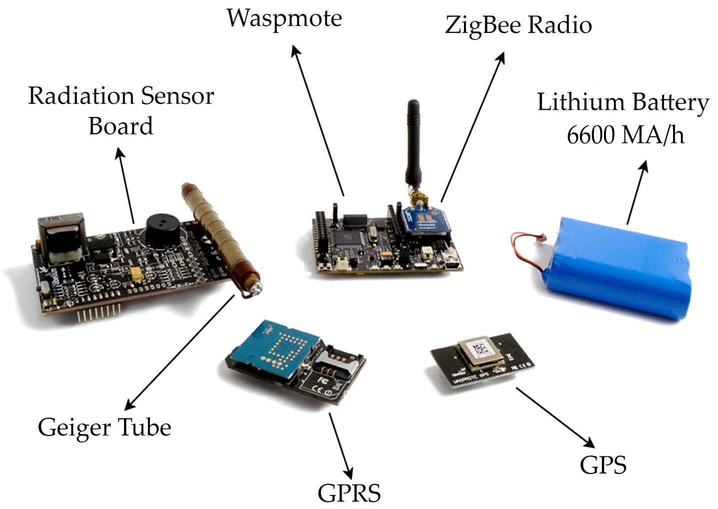


Figure 8 illustrates a basic structure of a big data system based on a WSN as an example of a fundamental system architecture. As shown in Figure 8, a sink node collects the data from sensor nodes and then delivers the data to a temporary storage for consequent data aggregation. After this step, the aggregated data can be manipulated by a big data framework using the main storage. The transformed data are handled by big data platforms and applications.

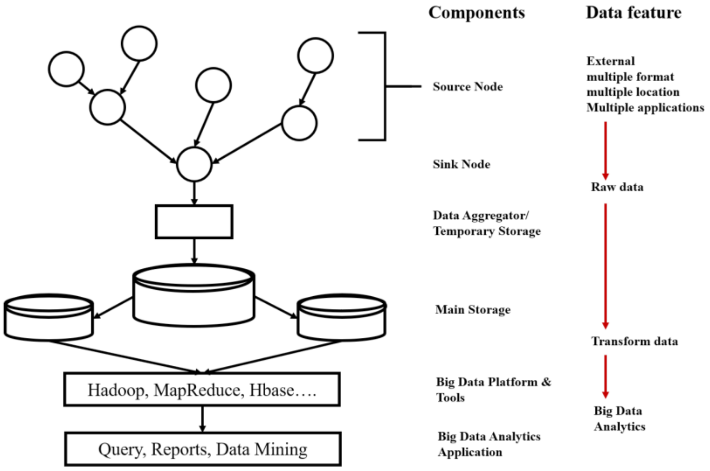


Figure 8. Big data system based on a WSN.

Convergence problems between WSNs/internet-of-things (IoT) and big data are reviewed and analyzed, wherein the following open issues are mentioned: convergence process, security, management of data, interoperability, and hardware/architecture challenges. From the point of view of data usage, data collected from IoT contribute to context-aware computing, such as ubiquitous and pervasive computing. Thus, big data issues in ambient intelligence and WSNs need to be explored well. Instead of describing the entire system, the process focused on wireless infrastructure such as data-aided transmission, data-driven network optimization, and novel applications under layered architecture (as shown in application, network, transmission, and data layers in **Figure 9**). Further, they discussed three potential application areas: smart grids, internet-of-things (IoT), and drones/unmanned aerial vehicles (UAV), as illustrated in **Figure 9**.

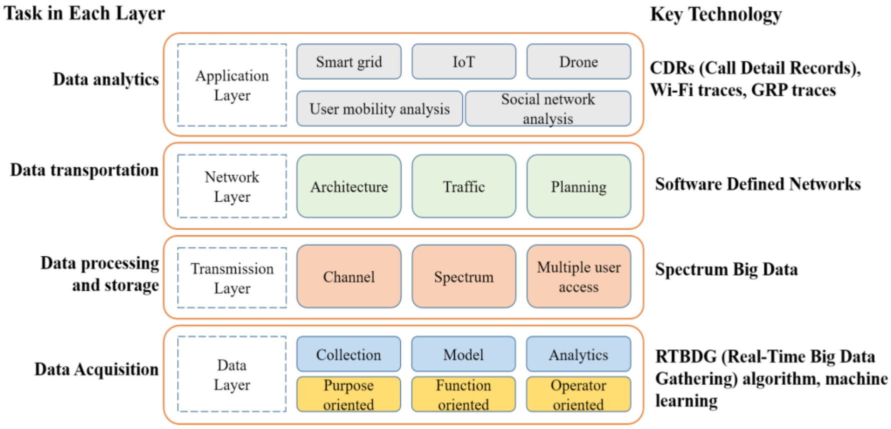


Figure 9. Protocol layering of wireless big data systems.

1.3.1 Applications of WSN-Based Big Data Systems

Before starting a detailed analysis of related work, it is highly desirable to understand which big data applications can be implemented and deployed through WSNs. Since a WSN is usually built to meet application-specific requirements, it is reasonable to review big data applications prior to addressing their technical issues.

The following monitoring applications can benefit from using WSNs: smart grids, monitoring human body, and monitoring the environment.

In case of smart grids, smart sensor networks are introduced in big data systems for energy management. These systems run smart grid applications that include power monitoring, demand-side energy management, coordination of distributed storage, and integration of renewable energy generators. Additionally, techniques used to manage big data generated by sensors and meters are proposed. Moreover, feasible recommendations and practices for smart grid are discussed.

The next example is monitoring the human body: wireless body area networks (WBAN). Collection of a vast amount of health and medical data via body sensor networks is presented for big data systems. To implement body sensor networks, an activity recognition application is required to implement the following functionality: feature extraction and selection, classification, supporting software platforms, and sensor and user authentication. This typical activity recognition procedure is classified and illustrated in **Figure 10**. Another interesting application of monitoring abnormal conditions of heart rate, with ZigBee and big data analysis based pulse monitoring system. To avoid missing the pulse signal, two following methods were proposed: (1) the photo-electricity based dynamic continuous heart rate monitoring methods, and (2) comprehensive anti-jamming methods. Using these two methods, a training model based on big data is proposed to improve the physical training level and to create training plans. With regard to data classification and detecting atypical events (anomaly detection).

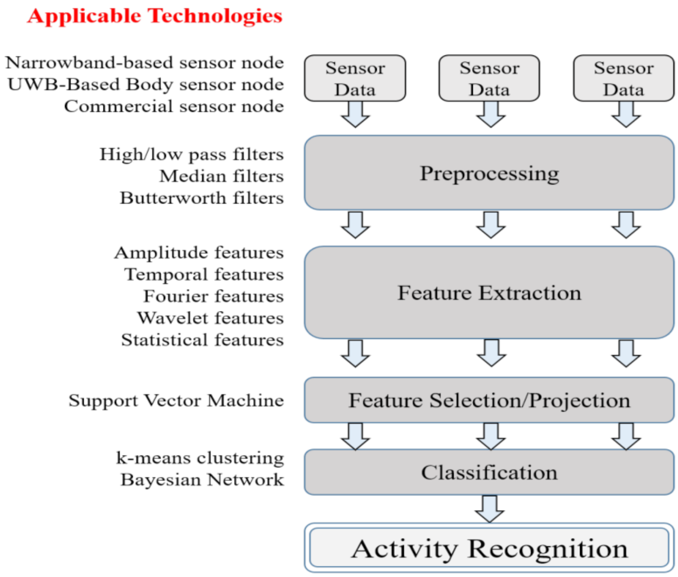


Figure 10. Typical activity recognition process for inferring activities from WBAN sensors.

Next, we consider examples of environmental monitoring, such as big data systems that monitor air quality in industrial workplace buildings. A case study of a big data system that monitors air quality collected by WSN. Locations include two workshops that are part of a large on-shore logistics base of a regional shipping industry in Norway. The study is conducted to prove the efficiency of data analytics and visualization.

Substantially, the case study reveals the possibility to monitor worker safety in other high-risk industries, as well as the quality of goods in supply chain management by integrating WSNs and big data systems. Moreover, as a possible monitoring application, cooperative fire security system using human agent robot machine sensor (CFS2H) message protocol for a firefight system is presented to provide fast communication and stable collaboration. The stationary WSN node is responsible of generating the data, while a big data center controls the whole system's work in the suggested architecture.

In this analysis, the data is converted into a shape similar to a map (that could be paper maps or digital ones) of e-government services. Usually, e-government services are known to improve efficiency and citizen satisfaction due to their implementation of spatial data infrastructure (SDI). Therefore, it is proven that e-government based on big data and WSN improves the access to e-services. Moreover, it could be used to overcome challenges of managing limited resources.

1.3.2 Technical Approaches to WSN-Based Big Data Systems

Wireless Sensor Network (WSN) is an infrastructure-less wireless network that is deployed in a large number of wireless sensors in an ad-hoc manner that is used to monitor the system, physical or environmental conditions. Sensor nodes are used in WSN with the onboard processor that manages and monitors the environment in a particular area. They are connected to the Base Station which acts as a processing unit in the WSN System. Base Station in a WSN System is connected through the Internet to share data.

Big Data through WSN

Data generated by the sensors grow exponentially. Conventional information technologies for data processing, storage, and reporting (such as servers and relational databases) are too expensive to deal with these data. Moreover, they cannot cope with the processing needs that can be required for real-time processes. In addition, most of the events monitored at regular intervals are largely redundant or are minor variations leading to a large waste of data storage resources and communication energy at relay and sensor nodes. This implies that much of these data are of no interest, meaningless, and redundant. Thus, unlike the case of typical WSNs, it is essential to gather and transmit a large amount of data while minimizing data latency in WSN-based big data systems. Moreover, it is required to efficiently eliminate data redundancy and improve energy efficiency. The overlap between big data systems and WSNs lies in the use of in-network data processing techniques. For the WSNs side, it would save their limited resources. At the same time, receiving a clean, non-redundant, and relevant data would reduce the excessive data volume at the side of the big data system. Thus, it would reduce overload by discovering values from these data rapidly.

Categorization

Research on big data systems based on WSNs is largely categorized into two main areas (**Figure 11**): network systems and data systems. The former is focused on network systems that deliver the sensor data to the big data system, while the latter is focused on data processing. Each research area has multiple subcategories according to the research objective.

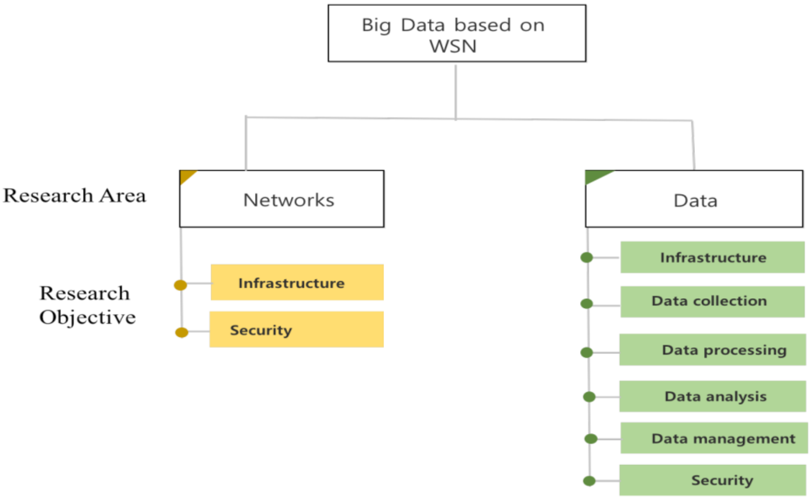


Figure 11. Classification of related work.

Network System

Infrastructure

In this research area, most work is conducted by proposing either network architectures or communication protocols. First, it propose structural construction of the WSN based on big data processing called service-oriented architecture and virtualization cloud for WSN (SVC4WSN) and simulates it numerically through comparison with multi-hop direct forwarding for local wireless sensor network (MDF4LWSN) architecture. The SVC4WSN consists of four layers: a large-scale WSN, gateway, cloud center, and users (as shown in **Figure 12**). In this architecture, there are two critical issues: congestion caused by the big data, and communication latency. To deal with these issues, flexible and multi-layer data processing and storage models based on cloud computing are proposed.

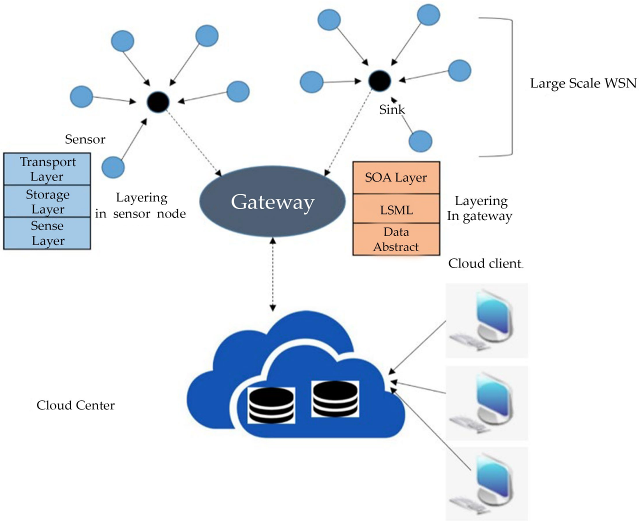


Figure 12. Architecture of SVC4WSN.

Another challenging task for gathering big data in a densely distributed sensor network with high energy efficiency. The method can be used to determine the sink node's trajectory and data-gathering through clustering. Unlike the typical clustering scheme, K-medoid clustering in six steps is proposed to keep energy consumption balanced in continuous iterations. In their work, the mobility of a sink node was selected as an effective solution to manage big data collection. Moreover, a framework to leverage the correlation between sets of active sensor nodes was presented. Another approach to utilize a mobile data collector, where two different approaches are suggested: data collection using data mule (MULE) and sensor network with mobile access point (SENMA). These approaches are characterized by the number of hops that are required to handle unexpected network partition during mobile data collection. It aims to reduce network congestion, in order to improve the reliability of data transmission, as well as reduce packet loss rate. To achieve these goals simultaneously, an efficient traffic load balancing algorithm is proposed to ensure balanced energy consumption within the network.

In addition to data collection, the data aggregation scheme for big data through the information-centric networking (ICN) approach, where data are retrieved by names and in-network caching. The proposed framework operates according to the following steps: (1) the network is initialized and the communication nodes are clustered using low-energy adoptive clustering hierarchy (LEACH) protocol; (2) collected data is aggregated to the cluster head memory; (3) an aggregately name-based routing (ANBR) method retrieves the data and forwards it to the data center. From another point of view, another architecture for WSN is proposed to prevent excessive energy consumption on a sensor node in case of the high redundancy of sensed data. A framework called structure fidelity data collection (SFDC) [**29**] leverages the spatial correlations between the nodes by reducing the number of active sensor nodes while maintaining a low structural distortion of the collected data. A node's duty cycle is controlled by a structural distortion depending on the image quality assessment approach. Thus, the data fidelity in terms of structural similarity in the continuous sensing applications for WSNs can be accomplished by SFDC.

As an example of specific-purpose routing protocol, a new routing protocol is proposed to assign a dynamic priority according to the requirements of quality-of-service (QoS) as well as achieve load distribution by involving a larger number of sensor nodes in the path. In particular, high energy efficiency is achieved by selecting a next hop according to the available resources and the required energy cost.

Consequently, the advantages and disadvantages of each method in the research area are compared in Table 1, where energy efficiency and management/maintenance cost are the major metrics.

Table 1. Comparison of advantages and disadvantages for infrastructure

Key Feature	Advantage	Disadvantage
Multi-layer model	Good lifecycle	High management cost
K-medoid clustering	High energy efficiency	High maintenance cost
Mobile sinks	Good for network partition	High energy consumption in few clusters
QoS provisioning	Balanced energy consumption	High complexity
Name and in-network caching	High energy efficiency	Low energy consumption in many clusters
Spatial correlation	High energy efficiency	No consideration for temporal correlation

Security

During data-gathering, WSN performs both data capture and transport, and it is important to accomplish these two tasks in a secure manner. Then, they propose a new architecture: trusted big data capture and transportation. In addition, misleading or forged data-gathering may occur in WSNs. Therefore, sensitive and critical data transmission through secure communication is required. While considering constraints on nodes, symmetric cryptography is very applicable to WSN due to its efficiency. However, symmetric cryptography should work with key management for distribution.

Data System

While the network system focuses on delivery of sensed data in WSN, the data system focuses on efficient processing of the data that are transmitted via WSN. The research objectives in data systems are more diverse than in network systems: they include infrastructure, data collection, processing, analysis, management, and security.

Infrastructure

Because of application-specific requirements and usage, manipulation, and exploitation of the data generated by WSN in the relevant data system for big data are in high demand.

First, an infrastructure may include big data tools for gathering, storage, and analysis of data generated by a WSN that monitors air pollution levels in a city. The proposed framework combines Hadoop and Storm for data processing, storage, and analysis, and Arduino-based kits for constructing unique sensor prototypes. In terms of components, the proposed system is composed of three main modules: data acquisition module (DAM), data processing module (DPM), and a messaging tool between DAM and DPM.

In addition to models of complete systems, there are models of specific components. For example, as for distributed data-centric storage in WSN. It aims at estimating the additional overhead for the real distribution of sensor nodes. As a technical enhancement, data redundancy among neighbor nodes and a simple routing protocol are proposed and evaluated in the experiments. The main contribution of their work is to reduce the energy consumption and improve the retrieval efficiency with the aid of proximity and routing algorithms, without the aid of GPS. The microcontroller, Smartphone, and host server tier are responsible for streamlining the sensor data transmission, forming the patterns in sensor data sets, and providing human expertise associated with the patterns. The proposed architecture leads to low-lost data transmission, early time-critical data mining, and urgent response for medical as well as healthcare applications. Additionally, it is interesting to introduce a Smartphone device in the tier. Furthermore, a detailed algorithm and approach for data mining are proposed to achieve the goal.

New requirements for integrating a WSN system and its associated services into a big data system. To implement this system, a holistic architecture is proposed to consider the flows of the data from sensors. Specifically, a constrained application protocol (CoAP) and a Linux service to integrate Hadoop with HBase datastore are employed over the core architecture on Linux. In the proposed architecture, multiple layers for node possess different capabilities depending on their roles.

In summary, major approaches in this research category are compared in **Table 2** in terms of advantages and disadvantages. Because most of these schemes are evaluated only for prototypes, further validation is required.

Table 2. Comparison of advantages and disadvantages of data systems

Key Feature	**Advantage**	**Disadvantage**
Framework based on Hadoop and Storm	Open-source based implementation	Low reliability
Distributed data-centric storage	High energy efficiency	Deployment issues in real-world scenarios
Three-tier data mining	Urgent response	Experiments in few scenarios

Data Collection

Although several approaches to data collection were already introduced in the terms of network system infrastructure, different algorithms. This implies that data collection through static and mobile sink are not changed in the data system. However, new models and procedures are defined according to the requirements of the system.

First, context-aware data mules (CADAMULE) are proposed as a solution for smart data collection in WSN. This information is used to capture the context information. Moreover, the proposed context-aware data mule aims at delivering the data to the sink. Consequently, another approach to introduce a mobile collector. This research determines the mobile sink node’s trajectory by introducing an M-mobile collector based on a clustering algorithm. In the proposed scheme, mobile collectors traverse a fixed path to collect data from cluster centroids and sensors in the clusters by multi-hop routing. MCDP operation is largely divided into network clustering, route planning, route combination, and data collecting. During these procedures, mobile sinks need to visit all the source nodes along the constrained routes while minimizing energy consumption. However, because this problem is NP-hard, two heuristic algorithms are proposed, accordingly. The former algorithm builds routes by adding a link to the partially formed routes between two end nodes based on a measure of cost savings, while the latter algorithm follows the sequential route building algorithm. To demonstrate suitability of the proposed scheme, the performance of a mobile collector; their main objective is to evaluate expectation maximization (EM) based clustering scheme as a function of the number of mobile nodes.

In addition to a mobile collector, another main issue of data collection for big data systems is energy efficiency. A new mobile sink routing and data-gathering method through the network clustering based on a modified expectation-maximization technique is suggested. Additionally, an optimal number of clusters to minimize the energy consumption through connectivity and data request flooding model are derived. Apart from energy efficiency, the problem of a long latency caused by the mobile collector. For this problem, Further, the amount of traffic is adjusted to prevent overwhelming by the predetermined threshold value. In the aspects of communications protocol, energy-efficient routing protocols to gather real-time data. Cluster header in BDEG is determined depending on the level of both the received signal strength indicator and the residual energy of the sensor nodes. Each cluster header (CH) within a cluster and relay node (RN) is connected to the cluster coordinator (CCO) nodes for inter-cluster communication. The proposed algorithm builds an energy-efficient route by balancing the load on the cluster headers, cluster coordinators, and relay nodes for gathering big data in an effective manner. Finally, adaptive distributed data-gathering (ADiDaG) technique to save energy in periodic WSN applications. Depending on operations in every round, ADiDaG takes data-gathering, sampling decision, and transmission phases. The main decision algorithm for each phase depends on the longest common subsequence similarity and grouping approach, as well as adaptive sensor sampling rate. Finally, a special case for data collection in indoor WSN. The proposed algorithm is performed according to the requirements of the risk analysis under the clustering architecture, which is built by using received signal strength indicator and residual energy information. Simulation results are given to demonstrate the suitability of RTBDG for industrial environments.

Table 3. Comparison of advantages and disadvantage for data collection

Key Feature	**Advantage**	**Disadvantage**
Situation and event awareness	Cost-efficient data collection	Simple weight based computation
M-mobile collector	High energy efficiency	Few scenarios for simulation
MDCP	High energy efficiency	Assumption of infinite storage memory
Mobile collector	High energy efficiency	Flooding based operation
Local data collector	Reduced data-gathering latency	Too much dependency on threshold value

Data Processing

It describes the-network processing in WSN: data aggregation and fusion technologies. They emphasized that processing sensor data inside the network (in-network) before any further processing helps save limited resources and prevent excessive data duplication.

In addition to a comprehensive survey, a specific algorithm called high-dimensional data aggregation control algorithm for big data (HDAC) In this study, information to eliminate the dimension not matching with the specified requirements is the main source. While handling this, the principal components method to analyze the remaining dimension is employed. In the process of data aggregation, the self-adaptive data aggregation mechanism is used to reduce the phenomenon of network delay. The simulation results evaluate node energy consumption and data delay through the HDAC. Recently, a comprehensive survey of data aggregation for big data in WSN. It covers research challenges in the big data area and proposes a new classification of these challenges, accordingly to the necessities of WSN. The detailed data aggregation strategies in WSNs are also discussed. They include: (1) distributed compressive data aggregation in large-scale WSN; (2) sensor data aggregation in a multi-layer big data framework; (3) lifting wavelet compression based data aggregation in big data WSN; (4) scalable privacy-preserving big data aggregation mechanism; and (5) a cluster-based data fusion technique to analyze big data in wireless multi-sensor systems.

Data Management

Big data tools and frameworks are introduced to evaluate performance of query processing and data collection. As the major challenge in data management in WSNs, and focus on energy preservation. In the aspects of energy efficiency, and emphasize decentralization, which is one of the promising ways to achieve energy preservation by distributing the computation tasks among sensor nodes.

The main contribution of this thesis is to propose an adaptive sampling approach for periodic data collection by allowing each sensor node to adjust its sampling rates in parallel with the physical changing dynamics. Additionally, periodic data aggregation on sensor node data is proposed as a preprocessing phase for an efficient and scalable data mining. Specifically, a new data mining method based on the existing K-means clustering is proposed.

Security

To detect intrusion efficiently and correctly, a WSN is usually a good solution: it can defend against the insider attacks using a relevant trust-based mechanism. However, due to excessive data, effectiveness of trust computation can be degraded significantly.

Security in multimedia big data applications for smart city in trust-assist sensor cloud (TASC). Two types of TASC: TASC-S (TASC with a single trust value threshold), and TASC-M (TASC with multiple trust value thresholds), are proposed and evaluated through extensive simulation results in the aspect of throughput. The major contribution of this work is to evaluate the trust value of a public key according to both the number of supporters and the trust degree of the public key.

SUMMARY

- Wireless sensor networks (WSNs) refer to networks of spatially dispersed and dedicated sensors that monitor and record the physical conditions of the environment and forward the collected data to a central location.
- Wireless Sensor Networks (WSNs) can be defined as a self-configured and infrastructure-less wireless networks to monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutants and to cooperatively pass their data through the network to a main location or sink where the data can be observed and analyzed.
- Wireless sensor networks (WSNs) enable new applications and require non-conventional paradigms for protocol design due to several constraints
- Wireless sensor networks have gained considerable popularity due to their flexibility in solving problems in different application domains and have the potential to change our lives in many different ways. WSNs have been successfully applied in various application domains.
- Sensor nodes are vulnerable and frequently deployed in dangerous environment. Nodes can fail due to hardware problems or physical damage or by exhausting their energy supply
- Sensor networks vary in scale from several nodes to potentially several hundred thousand. In addition, the deployment density is also variable.
- A mesh network allows transmitting data to one node to other node in the network that is within its radio transmission range.
- A star network is a communications topology where a single base station can send and/or receive a message to a number of remote nodes.
- A hybrid between the star and mesh network provides a robust and versatile communications network, while maintaining the ability to keep the wireless sensor nodes power consumption to a minimum.
- The sensor nodes are usually scattered in a sensor field as shown in Fig. 1. Each of these scattered sensor nodes has the capabilities to collect data and route data back to the sink and the end users.

REFERENCES

1. A. Cunha, M. Alves, and A. Koubaa, “Implementation Details of the Time Division BeaconFrame Scheduling Approach for Zigbee Cluster-Tree Networks,” IPP-HURRAY Technical Report TR070102 - http://www.open-zb.net, 2007.
2. Atmel Corp., “Low Power 2.4 Transceiver for ZigBee, IEEE 802.15.4, and ISM Applications,” www.atmel.com/dyn/resources/prod_documents/doc8111.pdf
3. Cirticom Corp, Securing Sensor Network, http://www.certicom.com, March 2006
4. Culler, D. E. and Hui, J. “IP on IEEE802.15.4 Low-Power Wireless Network,” Arch Rock Corporation, 2007
5. Cunha, A., Alves, M., and Koubà, A. “Implementation Details of the Time Division Beacon Scheduling Approach for ZigBee Cluster-Tree Networks,” www.hurray.isep.ipp.pt, 2007
6. Ron Ng W.L “Development of a Security System using Wireless FSK Communication” Thesis, University of Queensland, Australia, 2003.
7. W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan,”Energy-efficient Communication Protocols for Wireless Microsensor Networks,”Proceedings of the Hawaii International Conference on Systems Sciences, Hawai, 2000.

CHAPTER 2

WIRELESS TRANSMISSION TECHNOLOGY AND SYSTEMS

INTRODUCTION

The radio energy transmission transfers electrical energy from the transmitting end to the receiving end in the form of electromagnetic waves. Compared with the traditional cable transmission method, this technology not only eliminates the power loss in the cable transmission process, but also gets rid of the limitation of the transmission line, has broad development prospects, and has been paid attention to by many experts and scholars. From the end of the last century to the present, as scientists continue to deepen their research in this field, a series of scientific research results are presented to the world. At present, the theoretical development is relatively complete with the following forms of power transmission: electromagnetic induction wireless transmission; magnetically coupled resonant wireless transmission and wireless transmission under the microwave principle. Among them, electromagnetic induction wireless transmission is the most mature and has been put into commercial use. Wireless charging of small electronic devices (such as mobile phones and watches) that are common in our lives is based on this principle.

2.1 WIRELESS POWER TRANSMISSION TECHNOLOGY WITH APPLICATIONS

Nowadays electricity is considered as one of the basic needs of human beings. The conventional power transmission system uses transmission lines to carry the power from one place to another, but it is costlier in terms of cable costs and also there exists a certain transmission loss. One maintainable technology leading this charge is a wireless power transmission (WPT). It is also known as inductive power transfer (IPT).

Wireless Power Transmission

Wireless communication (or just wireless, when the context allows) is the transfer of information between two or more points that do not use an electrical conductor as a medium by which to perform the transfer. The most common wireless technologies use radio waves. With radio waves, intended distances can be short, such as a few meters for Bluetooth or as far as millions of kilometers for deep-space radio communications. It encompasses various types of fixed, mobile, and portable applications, including two-way radios, cellular telephones, personal digital assistants (PDAs), and wireless networking. Other examples of applications of radio wireless technology include GPS units, garage door openers, wireless computer mouse, keyboards and headsets, headphones, radio receivers, satellite television, broadcast television and cordless telephones. Somewhat less common methods of achieving wireless communications include the use of other electromagnetic wireless technologies, such as light, magnetic, or electric fields or the use of sound.

The term wireless has been used twice in communications history, with slightly different meaning. It was initially used from about 1890 for the first radio transmitting and receiving technology, as in wireless telegraphy, until the new word radio replaced it around 1920. Radios in the UK that were not portable continued to be referred to as wireless sets into the 1960s. The term was revived in the 1980s and 1990s mainly to distinguish digital devices that communicate without wires, such as the examples listed in the previous paragraph, from those that require wires or cables. This became its primary usage in the 2000s, due to the advent of technologies such as mobile broadband, Wi-Fi and Bluetooth.

Wireless operations permit services, such as mobile and interplanetary communications, that are impossible or impractical to implement with the use of wires. The term is commonly used in the telecommunications industry to refer to telecommunications systems (e.g. radio transmitters and receivers, remote controls, etc.) which use some form of energy (e.g. radio waves, acoustic energy,) to transfer information without the use of wires. Information is transferred in this manner over both short and long distances.

2.1.1 Wireless Power Transmission Technology

Wireless power transmission technology is not a new technology. In 1980, it was demonstrated by Nikola Telsa. There are three main systems used for wireless electricity transmission: solar cells, microwaves and resonance. In an electrical device, microwaves are used to transmit electromagnetic radiation from a source to a receiver. The name wireless power transmission states the transfer of electrical power from a source to an electrical device without the help of wires. Basically, it involves two coils: a transmitter and a receiver coil. The transmitter coil is powered by an AC current to produce a magnetic field, which in turn induces a voltage in the receiver coil.

The basics of WPT involve the inductive energy transmission from a transmitter to a receiver through an oscillating magnetic field. To get this DC current, that is supplied by a power source, it is converted into high -frequency AC current by the specially designed electronics built into the transmitter.

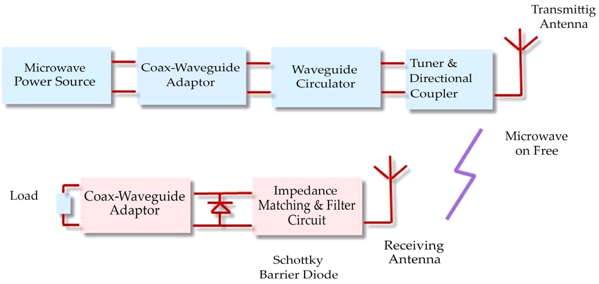
In the transmitter section, the AC current boosts a copper wire, which generates a magnetic field. Once a receiver coil is placed within the close vicinity of the magnetic field, the field can induce an AC current in the receiving coil. The electrons in the receiving device, then converts the AC current back into DC current, which becomes utilizable power.

2.1.2 Types of Wireless Power Transmission Methods

There are different types of wireless power transmission methods: microwave power transmission, inductive-coupling-power transmission and laser-power transmission methods.

1. *Microwave Power Transmission*

William C Brown, the pioneer in the WPT technology, has designed and exhibited to show how power can be transmitted through free space by microwaves. The concept of the WPT is explained with a functional block diagram which is shown below.



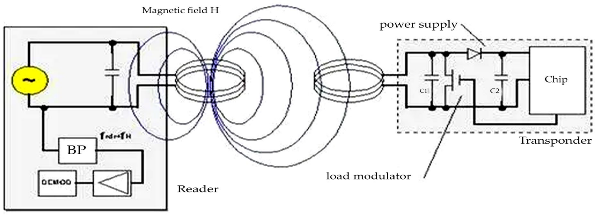
Microwave Power Transmission

The functional block diagram of WPT consists of two sections: transmitting section and receiving section. In the transmission section, the microwave power source generates microwave power which is controlled by the electronic control circuits. The waveguide circulator protects the microwave source from the reflected power, which is connected through the co-ax waveguide adaptor. The tuner contests the impedance between the microwave source and transmitting antenna. Then, based on the signal propagation direction, the attenuated signals are separated by the directional coupler. The transmitting antenna emits the power regularly through free space to the receiving antenna.

In the receiving section, the receiving antenna receives the transmitted power and converts the microwave power into DC power. The filter and impedance matching circuit is provided for setting the output impedance of a signal source which is equal to rectifying circuit. This circuit consists of Schottky barrier diodes which converts the received microwave power into DC power.

2. *Inductive Coupling Power Transmission:*

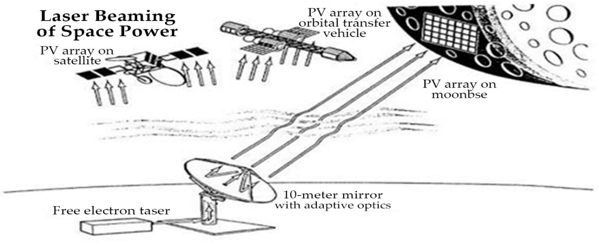
Inductive coupling method is the most important methods transferring energy wirelessly through inductive coupling. Basically, it is used for near -field power transmission. The power transmission takes place between the two conductive materials through mutual inductance. The general example of inductive coupling power transmission is a transformer.



Inductive Coupling Power Transmission

3. Laser Power Transmission:

In this type of power transmission method, a LASER is used to transfer power in the form of light energy, and the power is converted to electric energy at the receiver end. The LASER gets powered using different sources like sun, electricity generator or high-intensity-focused light. The size and shape of the beam are decided by a set of optics. The transmitted LASER light is received by the photo-voltaic cells, that converts the light into electrical signals. Usually, it uses optical-fiber cables for transmission.

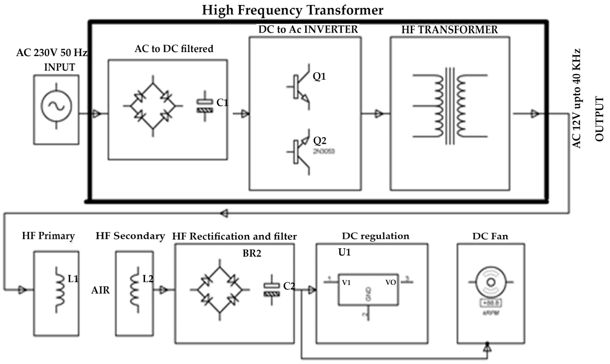

Laser Beaming of Space Power
PV array on satellite
PV array on orbital transfer vehicle
PV array on moonbse
10-meter mirror with adaptive optics
Free electron taser

A LASER power Transmission System

Laser Power Transmission System

2.1.3 Working Example of Wireless Power Transfer

The main intention of this project is to develop a system of wireless power transmission in 3D space.



Block Diagram of Wireless Power Transfer

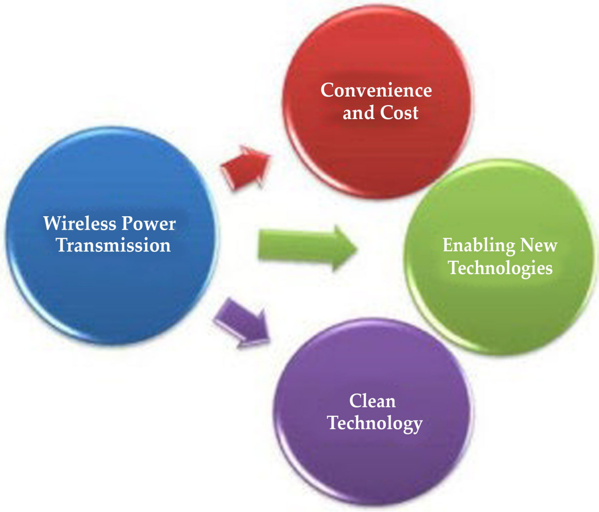
Hardware requirements include HF transformer, diode, rectifier, capacitors, transformers, lamp and air filled inductor coil. This project requires an AC 230V 50Hz to AC 20 KHz at 12V circuit. The AC, 50Hz is rectified by a BR1 and the DC is derived again, and then made to AC by transistors switching at 40KHz. This is connected to another HF transformer(high frequency) . The output of which is then fed to a resonating coil which acts as a primary of another air-core transformer. Then, the secondary high frequency ID is fed to a second rectifier to drive a DC load.

The main concept of this wireless power transfer in 3D space is, air core transformer operates at 40KHz and by the air core, one cannot transfer 50Hz AC power

The secondary coil magnetic resonance develops a voltage of 40KHz at about 12V while taking over the primary coil. But, the overall efficiency of the power transfer is below 70% for all weakly coupled series resonators that are used in this project

Benefits of WPT:

- WPT system completely reduces existing high-tension power transmission cables, substations and towers between the consumers and generating station.
- The cost of the distribution and transmission become less.
- The cost of the electrical energy to the consumers also reduces.
- The power could be transmitted to places to which the wired transmission is not possible.



2.1.4 Benefits of Wireless Power Transmission

Applications of Wireless Power Transmission:

- The largest application of the WPT is the production of power by placing satellites with giant solar arrays in Geosynchronous Earth Orbit and transmitting the power as microwaves to the earth known as Solar Power Satellites (SPS).
- WPT is used in moving targets like fuel-free-electric vehicles, fuel- free airplanes, fuel-free rockets and moving robots.
- The other applications of WPT are Wireless power source or Ubiquitous Power Source, RF Power Adaptive Rectifying Circuits and Wireless sensors.

In this way, one can design a wireless-power-transfer system for simple electronic devices like mobile charges, mobile phones which not only reduces the risk of shock, but also the efforts to plug repeatedly into the sockets.

2.1.5 Wireless Technology

Wireless technologies, in the simplest sense, enable one or more devices to communicate without physical connections—without requiring network or peripheral cabling. Wireless technologies use radio frequency transmissions as the means for transmitting data, whereas wired technologies use cables.

Wireless technologies range from complex systems, such as Wireless Local Area Networks (WLAN) and cell phones to simple devices such as wireless headphones, microphones, and other devices that do not process or store information. They also include infrared (IR) devices such as remote controls, some cordless computer keyboards and mice, and wireless hi-fi stereo headsets, all of which require a direct line of sight between the transmitter and the receiver to close the link.

Wireless Networks

Wireless network is a network set up by using radio signal frequency to communicate among computers and other network devices. Sometimes it is also referred to as WiFi network or WLAN. This network is getting popular nowadays due to easy to setup feature and no cabling involved. You can connect computers anywhere in your home without the need for wires. Here is simple explanation of how it works, let say you have 2 computers each equipped with wireless adapter and you have set up wireless router. When the computer send out the data, the binary data will be encoded to radio frequency and transmitted via wireless router. The receiving computer will then decode the signal back to binary data.

It does not matter you are using broadband cable/DSL modem to access internet, both ways will work with wireless network. If you heard about wireless hotspot that means that location is equipped with wireless devices for you and others to join the network.

The two main components are wireless router or access point and wireless clients. If you have not set up any wired network, then just get a wireless routerand attach it to cable or DSL modem. You then set up wireless client by adding wireless card to each computer and form a simple wireless network.

Wireless LANs

A wireless local area network (WLAN or wireless LAN) consists of two or more computers that communicate wirelessly via radio waves. This is contrasted to a wired LAN, in which each computer in the network is physically tethered with an Ethernet cable to the server's network switch or hub.

The basic structure of all networks consists of a main computer or server, along with connected machines known as clients. The server typically has two Ethernet network interface cards (NICs) installed and software that can support the network. Microsoft Windows operating systems have featured built-in networking capability since Windows 98 Special Edition (SE), but third party networking software is also available. In the case of a simple home wireless LAN, a desktop might be the server while a laptop could be the client.

Let's assume that the desktop has a digital subscriber line (DSL) connection -- high-speed Internet service. In order to share the Internet connection via a wireless LAN, the desktop will be physically connected to a wireless DSL modem. The wireless DSL modem might also have a built-in network switch or router. These two devices keep data flowing to the proper machine on the network. If they are not built into the modem, they will have to be purchased separately

The connections from the desktop server to the DSL modem, switch and router are made with physical Ethernet cables. The clients, however, do not require cabling. Instead, each machine must have a wireless NIC installed. This might be a PCMCIA NIC, a USB device or even an internal wireless NIC. Many, but not all, wireless NICs feature a small antenna.

Once the wireless LAN has been set up on the server and client, the machines can communicate by sending and receiving data via radio waves. This makes a wireless LAN very convenient because the client can remain mobile anywhere within the broadcasting range of the network. One can work on a laptop in any room in the house -- even the backyard in most cases -- and still share the network connection from the server.

At the office, a wireless LAN provides instant connectivity to mobile personnel. It also avoids the costly expense of running Ethernet cable throughout a building, providing easy, effortless desktop connectivity between clients. Because there are no wires running to the clients, one of the main advantages of a wireless LAN is easy installation. Wireless LANs also provide more flexibility than wired LANs and are less expensive.

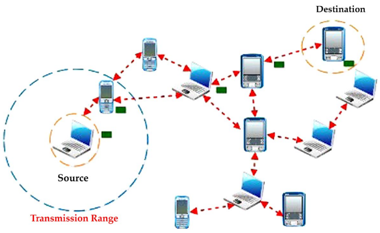
Two disadvantages of the wireless LAN are that it requires a few more steps to secure it from intrusion; and it can be slower and, if interference is present, less reliable than a wired network. However, dependency speeds are always improving as technology improves. Some configurations of wireless LANs can compete with wired networks.

Standards of wireless technology are indicated by letter designations. The wireless "g" standard delivers speeds of 54 megabits per second (mbps). With augmented technology like the latest varieties of multiple input multiple output (MIMO), rates can reach 100 mbps.

Ad Hoc Networks

Since the inception of wireless networking there have been two types of wireless networks: the infrastructure network, including some local area networks (LANs), and the ad hoc network. Ad hoc is Latin meaning "for this purpose." Ad hoc networks therefore refer to networks created for a particular purpose. They are often created on-the-fly and for one-time or temporary use. Often, these networks are comprised of a group of workstations or other wireless devices which communicate directly with each other to exchange information. Think of these connections as spontaneous networks, available to whomever is in a given area.

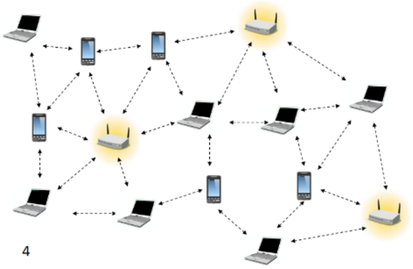
An ad hoc network is one where there are no access points passing information between participants. Infrastructure networks pass information through a central information hub which can be a hardware device or software on a computer. Office networks, for example, generally use a server to which company workstations connect to receive their information. Ad hoc networks, on the other hand, do not go through a central information hub.



These networks are generally closed in that they do not connect to the Internet and are typically created between participants. But, if one of the participants has a connection to a public or private network, this connection can be shared among other members of the network. This will allow other users on the spontaneous ad hoc network to connect to the Internet as well.

Ad hoc networks are common for portable video game systems like the Sony PSP or the Nintendo DS because they allow players to link to each other to play video games wirelessly. Some retail stores even create networks within them to allow customers to obtain new game demos via the store's own ad hoc network.

An ad hoc network can be thought of as a peer-to-peer network for the wireless age. Peer-to-peer or workgroup style networks were used to create a network environment for early Windows computers. This allowed these early computers to connect to each other to exchange information, usually in a smaller office environment without the need for domains and the additional management and overhead that comes with them.



Wireless Devices

A wireless device can refer to any kind of communications equipment that does not require a physical wire for relaying information to another device. Wireless headphones fitted with a receiver use either radio frequency (RF) or infrared technology to communicate with a transmitter that is connected to the sound source, say a television. In most cases, however, when someone refers to a wireless device, they are speaking of a networking device that can pass data to other wireless network gear without being physically connected.

In today's world, where people put a premium on staying connected to the Internet and to each other, there are several types of wireless technologies. In the home and office, wireless routers with built-in modems, hubs and switches broadcast a local area network (LAN) for computers in the area to join. Broadcasting distance varies widely depending on many factors, but a LAN generally spans 300 feet (91.44 m) or more. Any computer on the network can share resources that are connected to the network, including a high-speed Internet connection, printer or other office equipment.

In order to join a wireless LAN (WLAN), a computer must have a wireless network card or adapter installed. A network card is an internal wireless device manufactured to use the same language or protocol that wireless routers use. These protocols periodically evolve into new standards, however, causing compatibility issues in the interim. If a router uses a protocol that is not supported by an internal wireless device, an external wireless adapter can be used in an external port. The most common type is a USB dongle, but wireless network adapters are also available in Express Card® formats, giving laptop users a choice as to which port they would rather use.

Another type of wireless device might be part of a Personal Area Network (PAN). A PAN is created with Bluetooth® technology, designed to connect personal digital devices over very short distances of just a few feet, though the standard extends to 30 feet (9.14 m).

Bluetooth® is a very flexible and convenient type of network. It can be used to send print jobs from a laptop to a nearby printer without the hassle of setting up shared resources over a LAN. It is also used to connect Bluetooth®-enabled cell phones, personal digital assistants (PDAs), or Apple products to each other or to other Bluetooth®-enabled equipment including headsets, external speakers, or computers. Since Bluetooth® uses a different frequency range than LANs, you can use a Bluetooth® network “within” a LAN without interference.

2.2 BLUETOOTH

Bluetooth is a short-range wireless technology standard that is used for exchanging data between fixed and mobile devices over short distances using UHF radio waves in the ISM bands, from 2.402 GHz to 2.48 GHz, and building personal area networks (PANs). It was originally conceived as a wireless alternative to RS-232 data cables. It is mainly used as an alternative to wire connections, to exchange files between nearby portable devices and connect cell phones and music players with wireless headphones. In the most widely used mode, transmission power is limited to 2.5 milliwatts, giving it a very short range of up to 10 meters (30 feet).

Bluetooth is managed by the Bluetooth Special Interest Group (SIG), which has more than 35,000 member companies in the areas of telecommunication, computing, networking, and consumer electronics. The IEEE standardized Bluetooth as IEEE 802.15.1, but no longer maintains the standard. The Bluetooth SIG oversees development of the specification, manages the qualification program, and protects the trademarks. A manufacturer must meet Bluetooth SIG standards to market it as a Bluetooth device. A network of patents apply to the technology, which are licensed to individual qualifying devices. As of 2009, Bluetooth integrated circuit chips ship approximately 920 million units annually. By 2017, there were 3.6 billion Bluetooth devices shipping annually and the shipments were expected to continue increasing at about 12% a year.

Uses

Bluetooth is a standard wire-replacement communications protocol primarily designed for low power consumption, with a short range based on low-cost transceiver microchips in each device.

Ranges of Bluetooth devices by class

Class	Max. permitted power		Typ. range (m)
	(mW)	(dBm)	
1	100	20	~100
1.5	10	10	~20
2	2.5	4	~10
3	1	0	~1
4	0.5	–3	~0.5

Because the devices use a radio (broadcast) communications system, they do not have to be in visual line of sight of each other; however, a *quasi optical* wireless path must be viable. Range is power-class-dependent, but effective ranges vary in practice. See the table "Ranges of Bluetooth devices by class".

Officially Class 3 radios have a range of up to 1 metre (3 ft), Class 2, most commonly found in mobile devices, 10 metres (33 ft), and Class 1, primarily for industrial use cases,100 metres (300 ft). Bluetooth Marketing qualifies that Class 1 range is in most cases 20–30 metres (66–98 ft), and Class 2 range 5–10 metres (16–33 ft). The actual range achieved by a given link will depend on the qualities of the devices at both ends of the link, as well as the air conditions in between, and other factors.

The effective range varies depending on propagation conditions, material coverage, production sample variations, antenna configurations and battery conditions. Most Bluetooth applications are for indoor conditions, where attenuation of walls and signal fading due to signal reflections make the range far lower than specified line-of-sight ranges of the Bluetooth products.

Most Bluetooth applications are battery-powered Class 2 devices, with little difference in range whether the other end of the link is a Class 1 or Class 2 device as the lower-powered device tends to set the range limit. In some cases the effective range of the data link can be extended when a Class 2 device is connecting to a Class 1 transceiver with both higher sensitivity and transmission power than a typical Class 2 device. Mostly, however, the Class 1 devices have a similar sensitivity to Class 2 devices. Connecting two Class 1 devices with both high sensitivity and high power can allow ranges far in excess of the typical 100m, depending on the throughput required by the application. Some such devices allow open field ranges of up to 1 km and beyond between two similar devices without exceeding legal emission limits.

The Bluetooth Core Specification mandates a range of not less than 10 metres (33 ft), but there is no upper limit on actual range. Manufacturers' implementations can be tuned to provide the range needed for each case.

2.2.1 Bluetooth Profile

To use Bluetooth wireless technology, a device must be able to interpret certain Bluetooth profiles, which are definitions of possible applications and specify general behaviors that Bluetooth-enabled devices use to communicate with other Bluetooth devices. These profiles include settings to parameterize and to control the communication from the start. Adherence to profiles saves the time for transmitting the parameters anew before the bi-directional link becomes effective. There are a wide range of Bluetooth profiles that describe many different types of applications or use cases for devices.

A typical Bluetooth mobile phone headset

List of applications

- Wireless control and communication between a mobile phone and a handsfree headset. This was one of the earliest applications to become popular.
- Wireless control of and communication between a mobile phone and a Bluetooth compatible car stereo system (and sometimes between the SIM card and the car phone).
- Wireless communication between a smartphone and a smart lock for unlocking doors.
- Wireless control of and communication with iOS and Android device phones, tablets and portable wireless speakers.
- Wireless Bluetooth headset and Intercom. Idiomatically, a headset is sometimes called "a Bluetooth".
- Wireless streaming of audio to headphones with or without communication capabilities.
- Wireless streaming of data collected by Bluetooth-enabled fitness devices to phone or PC.
- Wireless networking between PCs in a confined space and where little bandwidth is required.
- Wireless communication with PC input and output devices, the most common being the mouse, keyboard and printer.
- Transfer of files, contact details, calendar appointments, and reminders between devices with OBEX[a] and sharing directories via FTP.
- Replacement of previous wired RS-232 serial communications in test equipment, GPS receivers, medical equipment, bar code scanners, and traffic control devices.

- For controls where infrared was often used.
- For low bandwidth applications where higher USB bandwidth is not required and cable-free connection desired.
- Sending small advertisements from Bluetooth-enabled advertising hoardings to other, discoverable, Bluetooth devices.
- Wireless bridge between two Industrial Ethernet (e.g., PROFINET) networks.
- Seventh and eighth generation game consoles such as Nintendo's Wii, and Sony's PlayStation 3 use Bluetooth for their respective wireless controllers.
- Dial-up internet access on personal computers or PDAs using a data-capable mobile phone as a wireless modem.
- Short-range transmission of health sensor data from medical devices to mobile phone, set-top box or dedicated telehealth devices.
- Allowing a DECT phone to ring and answer calls on behalf of a nearby mobile phone.
- Real-time location systems (RTLS) are used to track and identify the location of objects in real time using "Nodes" or "tags" attached to, or embedded in, the objects tracked, and "Readers" that receive and process the wireless signals from these tags to determine their locations.
- Personal security application on mobile phones for prevention of theft or loss of items. The protected item has a Bluetooth marker (e.g., a tag) that is in constant communication with the phone. If the connection is broken (the marker is out of range of the phone) then an alarm is raised. This can also be used as a man overboard alarm. A product using this technology has been available since 2009.
- Calgary, Alberta, Canada's Roads Traffic division uses data collected from travelers' Bluetooth devices to predict travel times and road congestion for motorists.
- Wireless transmission of audio (a more reliable alternative to FM transmitters)
- Live video streaming to the visual cortical implant device by Nabeel Fattah in Newcastle university 2017.
- Connection of motion controllers to a PC when using VR headsets

2.2.2 Bluetooth vs Wi-Fi (IEEE 802.11)

Bluetooth and Wi-Fi (Wi-Fi is the brand name for products using IEEE 802.11 standards) have some similar applications: setting up networks, printing, or transferring files. Wi-Fi is intended as a replacement for high-speed cabling for general local area network access in work areas or home. This category of applications is sometimes called wireless local area networks (WLAN). Bluetooth was intended for portable equipment and its applications. The category of applications is outlined as the wireless personal area network (WPAN). Bluetooth is a replacement for cabling in various personally carried applications in any setting and also works for fixed location applications such as smart energy functionality in the home (thermostats, etc.).

Wi-Fi and Bluetooth are to some extent complementary in their applications and usage. Wi-Fi is usually access point-centered, with an asymmetrical client-server connection with all traffic routed through the access point, while Bluetooth is usually symmetrical, between two Bluetooth devices. Bluetooth serves well in simple applications where two devices need to connect with a minimal configuration like a button press, as in headsets and speakers.

Devices

A Bluetooth USB dongle with a 100 m range

Bluetooth exists in numerous products such as telephones, speakers, tablets, media players, robotics systems, laptops, and console gaming equipment as well as some high definition headsets, modems, hearing aids and even watches. Given the variety of devices which use the Bluetooth, coupled with the contemporary deprecation of headphone jacks by Apple, Google, and other companies, and the lack of regulation by the FCC, the technology is prone to interference. Nonetheless Bluetooth is useful when transferring information between two or more devices that are near each other in low-bandwidth situations. Bluetooth is commonly used to transfer sound data with telephones (i.e., with a Bluetooth headset) or byte data with hand-held computers (transferring files).

Bluetooth protocols simplify the discovery and setup of services between devices. Bluetooth devices can advertise all of the services they provide. This makes using services easier, because more of the security, network address and permission configuration can be automated than with many other network types.

2.3 IEEE 802.11A/B/G/N SERIES OF WIRELESS LANS

IEEE 802.11 is part of the IEEE 802 set of local area network (LAN) technical standards, and specifies the set of media access control (MAC) and physical layer (PHY) protocols for implementing wireless local area network (WLAN) computer communication. The standard and amendments provide the basis for wireless network products using the Wi-Fi brand and are the world's most widely used wireless computer networking standards. IEEE 802.11 is used in most home and office networks to allow laptops, printers, smartphones, and other devices to communicate with each other and access the Internet without connecting wires.

The standards are created and maintained by the Institute of Electrical and Electronics Engineers (IEEE) LAN/MAN Standards Committee (IEEE 802). The base version of the standard was released in 1997 and has had subsequent amendments. While each amendment is officially revoked when it is incorporated in the latest version of the standard, the corporate world tends to market to the revisions because they concisely denote the capabilities of their products. As a result, in the marketplace, each revision tends to become its own standard.

IEEE 802.11 uses various frequencies including, but not limited to, 2.4 GHz, 5 GHz, 6 GHz, and 60 GHz frequency bands. Although IEEE 802.11 specifications list channels that might be used, the radio frequency spectrum availability allowed varies significantly by regulatory domain. The protocols are typically used in conjunction with IEEE 802.2, and are designed to interwork seamlessly with Ethernet, and are very often used to carry Internet Protocol traffic.

The 802.11 family consists of a series of half-duplex over-the-air modulation techniques that use the same basic protocol. The 802.11 protocol family employs carrier-sense multiple access with collision avoidance whereby equipment listens to a channel for other users (including non 802.11 users) before transmitting each frame (some use the term “packet”, which may be ambiguous: “frame” is more technically correct).

802.11-1997 was the first wireless networking standard in the family, but 802.11b was the first widely accepted one, followed by 802.11a, 802.11g, 802.11n, and 802.11ac. Other standards in the family (c–f, h, j) are service amendments that are used to extend the current scope of the existing standard, which amendments may also include corrections to a previous specification.

802.11b and 802.11g use the 2.4-GHz ISM band, operating in the United States under Part 15 of the U.S. Federal Communications Commission Rules and Regulations. 802.11n can also use that 2.4-GHz band. Because of this choice of frequency band, 802.11b/g/n equipment may occasionally suffer interference in the 2.4-GHz band from microwave ovens, cordless telephones, and Bluetooth devices. 802.11b and 802.11g control their interference and susceptibility to interference by using direct-sequence spread spectrum (DSSS) and orthogonal frequency-division multiplexing (OFDM) signaling methods, respectively.

802.11a uses the 5 GHz U-NII band--which, for much of the world, offers at least 23 non-overlapping, 20-MHz-wide channels--rather than the 2.4-GHz, ISM-frequency band--which offers only three non-overlapping, 20-MHz-wide channels--where other adjacent channels overlap (list of WLAN channels). Better or worse performance with higher or lower frequencies (channels) may be realized, depending on the environment. 802.11n can use either the 2.4 GHz or 5 GHz band; 802.11ac uses only the 5 GHz band. The segment of the radio frequency spectrum used by 802.11 varies between countries. In the US, 802.11a and 802.11g devices may be operated without a license, as allowed in Part 15 of the FCC Rules and Regulations. Frequencies used by channels one through six of 802.11b and 802.11g fall within the 2.4 GHz amateur radio band. Licensed amateur radio operators may operate 802.11b/g devices under Part 97 of the FCC Rules and Regulations, allowing increased power output but not commercial content or encryption.

2.3.1 Generations

The Wi-Fi Alliance began using a consumer-friendly generation numbering scheme for the publicly used 802.11 protocols. Wi-Fi generations 1–6 refer to the 802.11b, 802.11a, 802.11g, 802.11n, 802.11ac, and 802.11ax protocols, in that order.

Wi-Fi Generations

Generation/IEEE Standard	Maximum Linkrate	Adopted	Frequency
WiFi 6E (802.11ax)	600 to 9608 Mbit/s	2020	6 GHz
WiFi 6 (802.11ax)	600 to 9608 Mbit/s	2019	2.4/5 GHz
WiFi 5 (802.11ac)	433 to 6933 Mbit/s	2014	5 GHz
WiFi 4 (802.11n)	72 to 600 Mbit/s	2008	2.4/5 GHz
(Wi-Fi 3)* 802.11g	6 to 54 Mbit/s	2003	2.4 GHz
(Wi-Fi 2)* 802.11a	6 to 54 Mbit/s	1999	5 GHz
(Wi-Fi 1)* 802.11b	1 to 11 Mbit/s	1999	2.4 GHz
(Wi-Fi 0)* 802.11	1 to 2 Mbit/s	1997	2.4 GHz

802.11a (OFDM Waveform)

802.11a, published in 1999, uses the same data link layer protocol and frame format as the original standard, but an OFDM based air interface (physical layer) was added. It was later relabeled Wi-Fi 1, by the Wi-Fi Alliance, relative to Wi-Fi 2 (802.11b).

It operates in the 5 GHz band with a maximum net data rate of 54 Mbit/s, plus error correction code, which yields realistic net achievable throughput in the mid-20 Mbit/s. It has seen widespread worldwide implementation, particularly within the corporate workspace.

Since the 2.4 GHz band is heavily used to the point of being crowded, using the relatively unused 5 GHz band gives 802.11a a significant advantage. However, this high carrier frequency also brings a disadvantage: the effective overall range of 802.11a is less than that of 802.11b/g. In theory, 802.11a signals are absorbed more readily by walls and other solid objects in their path due to their smaller wavelength, and, as a result, cannot penetrate as far as those of 802.11b. In practice, 802.11b typically has a higher range at low speeds (802.11b will reduce speed to 5.5 Mbit/s or even 1 Mbit/s at low signal strengths). 802.11a also suffers from interference, but locally there may be fewer signals to interfere with, resulting in less interference and better throughput.

802.11b

The 802.11b standard has a maximum raw data rate of 11 Mbit/s (Megabits per second) and uses the same media access method defined in the original standard. 802.11b products appeared on the market in early 2000, since 802.11b is a direct extension of the modulation technique defined in the original standard. The dramatic increase in throughput of 802.11b (compared to the original standard) along with simultaneous substantial price reductions led to the rapid acceptance of 802.11b as the definitive wireless LAN technology.

Devices using 802.11b experience interference from other products operating in the 2.4 GHz band. Devices operating in the 2.4 GHz range include microwave ovens, Bluetooth devices, baby monitors, cordless telephones, and some amateur radio equipment. As unlicensed intentional radiators in this ISM band, they must not interfere with and must tolerate interference from primary or secondary allocations (users) of this band, such as amateur radio.

802.11g

In June 2003, a third modulation standard was ratified: 802.11g. This works in the 2.4 GHz band (like 802.11b), but uses the same OFDM based transmission scheme as 802.11a. It operates at a maximum physical layer bit rate of 54 Mbit/s exclusive of forward error correction codes, or about 22 Mbit/s average throughput. 802.11g hardware is fully backward compatible with 802.11b hardware, and therefore is encumbered with legacy issues that reduce throughput by ~21% when compared to 802.11a.

The then-proposed 802.11g standard was rapidly adopted in the market starting in January 2003, well before ratification, due to the desire for higher data rates as well as reductions in manufacturing costs. By summer 2003, most dual-band 802.11a/b products became dual-band/tri-mode, supporting a and b/g in a single mobile adapter card or access point. Details of making b and g work well together occupied much of the lingering technical process; in an 802.11g network, however, the activity of an 802.11b participant will reduce the data rate of the overall 802.11g network.

Like 802.11b, 802.11g devices also suffer interference from other products operating in the 2.4 GHz band, for example, wireless keyboards.

802.11-2007

In 2003, task group TGma was authorized to “roll up” many of the amendments to the 1999 version of the 802.11 standard. REVma or 802.11ma, as it was called, created a single document that merged 8 amendments (802.11a, b, d, e, g, h, i, j) with the base standard. Upon approval on 8

March 2007, 802.11REVma was renamed to the then-current base standard IEEE 802.11-2007.

802.11n

802.11n is an amendment that improves upon the previous 802.11 standards; its first draft of certification was published in 2006. The 802.11n standard was retroactively labelled as **Wi-Fi 4** by the Wi-Fi Alliance. The standard added support for multiple-input multiple-output antennas (MIMO). 802.11n operates on both the 2.4 GHz and the 5 GHz bands. Support for 5 GHz bands is optional. Its net data rate ranges from 54 Mbit/s to 600 Mbit/s. The IEEE has approved the amendment, and it was published in October 2009. Prior to the final ratification, enterprises were already migrating to 802.11n networks based on the Wi-Fi Alliance›s certification of products conforming to a 2007 draft of the 802.11n proposal.

2.3.2 Security of 802.11 Wireless LANs

Due to the RF signal nature of the wireless network, it is very difficult to control which computers or devices are receiving the wireless network signal. Therefore, the wireless relies on software link-level protection, specifically implementing cryptography to protect from eavesdropping and other network attacks. The original 802.11 standard only offers WEP to secure the wireless network.

Security Features of 802.11 Wireless LANs per the Standard

The three basic security services defined by IEEE for the WLAN environment are as follows:

- Authentication: A primary goal of WEP was to provide a security service to verify the identity of communicating client stations. This provides access control to the network by denying access to client stations that cannot authenticate properly. This service addresses the question, “Are only authorized persons allowed to gain access to my network?”
- Confidentiality: Confidentiality, or privacy, was a second goal of WEP. It was developed to provide “privacy achieved by a wired network.” The intent was to prevent information compromise from casual eavesdropping (passive attack). This service, in general,

addresses the question, “Are only authorized persons allowed to view my data?”

- Integrity: Another goal of WEP was a security service developed to ensure that messages are not modified in transit between the wireless clients and the access point in an active attack. This service addresses the question, “Is the data coming into or exiting the network trustworthy—has it been tampered with?”

It is important to note that the standard did not address other security services such as audit, authorization, and nonrepudiation. The security services offered by 802.11 are described in greater detail below.

Authentication

The IEEE 802.11 specification defines two means to “validate” wireless users attempting to gain access to a wired network: open-system authentication and shared-key authentication. One means, shared-key authentication, is based on cryptography, and the other is not. The open-system authentication technique is not truly authentication; the access point accepts the mobile station without verifying the identity of the station. It should be noted also that the authentication is only one-way: only the mobile station is authenticated. The mobile station must trust that it is communicating to a real AP. A taxonomy of the techniques for 802.11 is depicted in Figure 1.

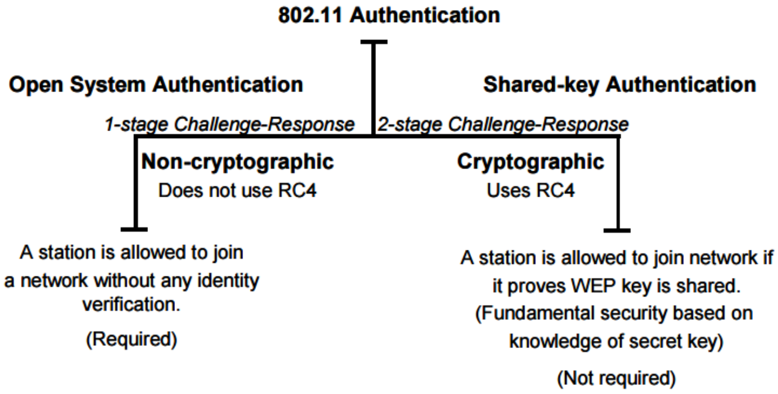

802.11 Authentication
Open System Authentication
Shared-key Authentication
1-stage Challenge-Response
2-stage Challenge-Response
Non-cryptographic
Does not use RC4
Cryptographic
Uses RC4
A station is allowed to join a network without any identity verification.
(Required)
A station is allowed to join network if it proves WEP key is shared.
(Fundamental security based on knowledge of secret key)
(Not required)

Figure 1: Taxonomy of 802.11 Authentication Techniques

With Open System authentication, a client is authenticated if it simply responds with a MAC address during the two-message exchange with an access point. During the exchange, the client is not truly validated but simply responds with the correct fields in the message exchange. Obviously, without cryptographic validatedation, open-system authentication is highly vulnerable to attack and practically invites unauthorized access. Open-system authentication is the only required form of authentication by the 802.11 specification.

Shared key authentication is a cryptographic technique for authentication. It is a simple "challengeresponse" scheme based on whether a client has knowledge of a shared secret. In this scheme, as depicted conceptually in Figure 2, a random challenge is generated by the access point and sent to the wireless client. The client, using a cryptographic key that is shared with the AP, encrypts the challenge (or "nonce," as it is called in security vernacular) and returns the result to the AP. The AP decrypts the result computed by the client and allows access only if the decrypted value is the same as the random challenge transmitted. The algorithm used in the cryptographic computation and for the generation of the 128-bit challenge text is the RC4 stream cipher developed by Ron Rivest of MIT. It should be noted that the authentication method just described is a rudimentary cryptographic technique, and it does not provide mutual authentication. That is, the client does not authenticate the AP, and therefore there is no assurance that a client is communicating with a legitimate AP and wireless network. It is also worth noting that simple unilateral challenge-response schemes have long been known to be weak. They suffer from numerous attacks including the infamous "man-in-the-middle" attack. Lastly, the IEEE 802.11 specification does not require shared-key authentication.

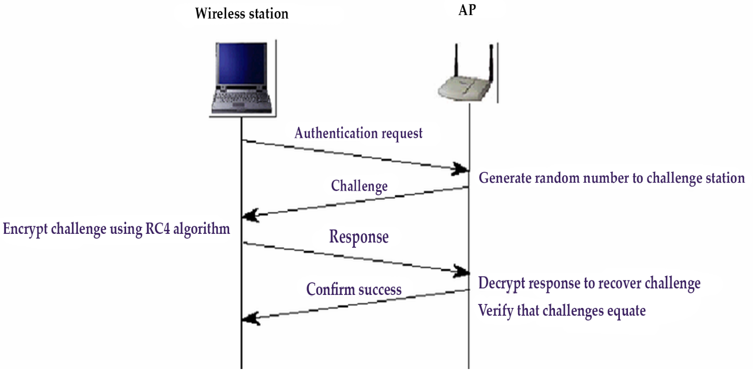

Wireless station
AP
Authentication request
Generate random number to challenge station
Challenge
Encrypt challenge using RC4 algorithm
Response
Decrypt response to recover challenge
Confirm success
Verify that challenges equate

Figure 2: Shared-key Authentication Message Flow

Privacy

The 802.11 standard supports privacy (confidentiality) through the use of cryptographic techniques for the wireless interface. The WEP cryptographic technique for confidentiality also uses the RC4 symmetrickey, stream cipher algorithm to generate a pseudo-random data sequence. This “key stream” is simply added modulo 2 (exclusive-OR-ed) to the data to be transmitted. Through the WEP technique, data can be protected from disclosure during transmission over the wireless link. WEP is applied to all data above the 802.11 WLAN layers to protect traffic such as Transmission Control Protocol/Internet Protocol (TCP/IP), Internet Packet Exchange (IPX), and Hyper Text Transfer Protocol (HTTP).

As defined in the 802.11 standard, WEP supports only a 40-bit cryptographic keys size for the shared key. However, numerous vendors offer nonstandard extensions of WEP that support key lengths from 40 bits to 104 bits. At least one vendor supports a keysize of 128 bits. The 104-bit WEP key, for instance, with a 24- bit Initialization Vector (IV) becomes a 128-bit RC4 key. In general, all other things being equal, increasing the key size increases the security of a cryptographic technique. However, it is always possible for flawed implementations or flawed designs to prevent long keys from increasing security. Research has shown that key sizes of greater than 80-bits, for robust designs and implementations, make brute-force cryptanalysis (code breaking) an impossible task. For 80-bit keys, the number of possible keys—a keyspace of more than 10^{26}—exceeds contemporary computing power. In practice, most WLAN deployments rely on 40-bit keys. Moreover, recent attacks have shown that the WEP approach for privacy is, unfortunately, vulnerable to certain attacks regardless of keysize. However, the cryptographic, standards, and vendor WLAN communities have developed enhanced WEP, which is available as a prestandard vendor-specific implementations. The WEP privacy is illustrated conceptually in Figure 3.

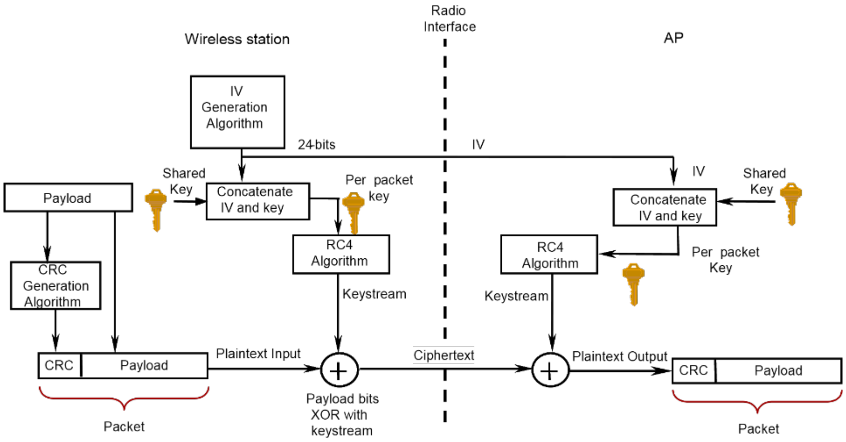


Figure 3: WEP Privacy Using RC4 Algorithm

Integrity

The IEEE 802.11 specification also outlines a means to provide data integrity for messages transmitted between wireless clients and access points. This security service was designed to reject any messages that had been changed by an active adversary "in the middle." This technique uses a simple encrypted Cyclic Redundancy Check (CRC) approach. As depicted in the diagram above, a CRC-32, or frame check sequence, is computed on each payload prior to transmission. The integrity-sealed packet is then encrypted using the RC4 key stream to provide the cipher-text message. On the receiving end, decryption is performed and the CRC is recomputed on the message that is received. The CRC computed at the receiving end is compared with the one computed with the original message. If the CRCs do not equal, that is, "received in error," this would indicate an integrity violation (an active message spoofer), and the packet would be discarded. As with the privacy service, unfortunately, the 802.11 integrity is vulnerable to certain attacks regardless of key size. In summary, the fundamental flaw in the WEP integrity scheme is that the simple CRC is not a "cryptographically secure" mechanism such as a hash or message authentication code.

The IEEE 802.11 specification does not, unfortunately, identify any means for key management (life cycle handling of cryptographic keys and related material). Therefore, generating, distributing, storing, loading, escrowing, archiving, auditing, and destroying the material is left to those deploying WLANs. Key management (probably the most critical aspect of a cryptographic system) for 802.11 is left largely as an exercise for the users of the 802.11 network. As a result, many vulnerabilities could be introduced into the WLAN environment. These vulnerabilities include WEP keys that are non-unique, never changing, factory-defaults, or weak keys (all zeros, all ones, based on easily guessed passwords, or other similar trivial patterns). Additionally, because key management was not part of the original 802.11 specification, with the key distribution unresolved, WEP-secured WLANs do not scale well. If an enterprise recognizes the need to change keys often and to make them random, the task is formidable in a large WLAN environment. For example, a large campus may have as many as 15,000 APs. Generating, distributing, loading, and managing keys for an environment of this size is a significant challenge. It is has been suggested that the only practical way to distribute keys in a large dynamic environment is to publish it. However, a fundamental tenet of cryptography is that cryptographic keys remain secret. Hence we have a major dichotomy. This dichotomy exists for any technology that neglects to elegantly address the key distribution problem.

Problems with the IEEE 802.11 Standard Security

This vulnerability in the standardized security of the 802.11 WLAN standard. As mentioned above, the WEP protocol is used in 802.11-based WLANs. WEP in turn uses a RC4 cryptographic algorithm with a variable length key to protect traffic. Again, the 802.11 standard supports WEP cryptographic keys of 40-bits. However, some vendors have implemented products with keys 104-bit keys and even 128-bit keys. With the addition of the 24-bit IV, the actual key used in the RC4 algorithm is 152 bits for the 128 bits WEP key. It is worthy to note that some vendors generate keys after a keystroke from a user, which, if done properly, using the proper random processes, can result in a strong WEP key. Other vendors, however, have based WEP keys on passwords that are chosen by users; this typically reduces the effective key size.

Several groups of computer security specialists have discovered security problems that let malicious users compromise the security of WLANs. These include passive attacks to decrypt traffic based on statistical analysis, active attacks to inject new traffic from unauthorized mobile stations (i.e., based on known plain text), active attacks to decrypt traffic (i.e., based on tricking the access point), and dictionary-building attacks. The dictionary building attack is possible after analyzing enough traffic on a busy network.

Security problems with WEP include the following:

The use of static WEP keys—many users in a wireless network potentially sharing the identical key for long periods of time, is a well-known security vulnerability. This is in part due to the lack of any key management provisions in the WEP protocol. If a computer such as a laptop were to be lost or stolen, the key could become compromised along with all the other computers sharing that key. Moreover, if every station uses the same key, a large amount of traffic may be rapidly available to an eavesdropper for analytic attacks, such as 2 and 3 below.

The IV in WEP, as shown in Figure 3, is a 24-bit field sent in the clear text portion of a message. This 24-bit string, used to initialize the key stream generated by the RC4 algorithm, is a relatively small field when used for cryptographic purposes. Reuse of the same IV produces identical key streams for the protection of data, and the short IV guarantees that they will repeat after a relatively short time in a busy network. Moreover, the 802.11 standard does not specify how the IVs are set or changed, and individual wireless NICs from the same vendor may all generate the same IV sequences, or some wireless NICs may possibly use a constant IV. As a result, hackers can record network traffic, determine the key stream, and use it to decrypt the cipher-text.

The IV is a part of the RC4 encryption key. The fact that an eavesdropper knows 24-bits of every packet key, combined with a weakness in the RC4 key schedule, leads to a successful analytic attack, that recovers the key, after intercepting and analyzing only a relatively small amount of traffic. This attack is publicly available as an attack script and open source code.

WEP provides no cryptographic integrity protection. However, the 802.11 MAC protocol uses a noncryptographic Cyclic Redundancy Check (CRC) to check the integrity of packets, and acknowledge packets with the correct checksum. The combination of noncryptographic checksums with stream ciphers is dangerous and often introduces vulnerablities, as is the case for WEP. There is an active attack that permits the attacker to decrypt any packet by systematically modifying the packet and CRC sending it to the AP and noting whether the packet is acknowledged. These kinds of attacks are often subtle, and it is now considered risky to design encryption protocols that do not include cryptographic integrity protection, because of the possibility of interactions with other protocol levels that can give away information about cipher text.

Note that only one of the four problems listed above depends on a weakness in the cryptographic algorithm. Therefore, these problems would not be improved by substituting a stronger stream cipher. For example, the third problem listed above is a consequence of a weakness in the implementation of the RC4 stream cipher that is exposed by a poorly designed protocol.

2.4 ZIGBEE

Zigbee is a wireless technology developed as an open global standard to address the unique needs of low-cost, low-power wireless IoT networks. The Zigbee standard operates on the IEEE 802.15.4 physical radio specification and operates in unlicensed bands including 2.4 GHz, 900 MHz and 868 MHz. The 802.15.4 specification upon which the Zigbee stack operates gained ratification by the Institute of Electrical and Electronics Engineers (IEEE) in 2003. The specification is a packet-based radio protocol intended for low-cost, battery-operated devices. The protocol allows devices to communicate in a variety of network topologies and can have battery life lasting several years. Zigbee is an IEEE 802.15.4-based specification for a suite of high-level communication protocols used to create personal area networks with small, low-power digital radios, such as for home automation, medical device data collection, and other low-power low-bandwidth needs, designed for small scale projects which need wireless connection. Hence, Zigbee is a low-power, low data rate, and close proximity (i.e., personal area) wireless ad hoc network.

The technology defined by the Zigbee specification is intended to be simpler and less expensive than other wireless personal area networks (WPANs), such as Bluetooth or more general wireless networking such as Wi-Fi. Applications include wireless light switches, home energy monitors, traffic management systems, and other consumer and industrial equipment that requires short-range low-rate wireless data transfer.

Its low power consumption limits transmission distances to 10–100 meters line-of-sight, depending on power output and environmental characteristics. Zigbee devices can transmit data over long distances by passing data through a mesh network of intermediate devices to reach more distant ones. Zigbee is typically used in low data rate applications that require long battery life and secure networking. (Zigbee networks are secured by 128 bit symmetric encryption keys.) Zigbee has a defined rate of 250 kbit/s, best suited for intermittent data transmissions from a sensor or input device.

Zigbee was conceived in 1998, standardized in 2003, and revised in 2006. The name refers to the waggle dance of honey bees after their return to the beehive.

Zigbee is a low-cost, low-power, wireless mesh network standard targeted at battery-powered devices in wireless control and monitoring applications. Zigbee delivers low-latency communication. Zigbee chips are typically integrated with radios and with microcontrollers. Zigbee operates in the industrial, scientific and medical (ISM) radio bands: 2.4 GHz in most jurisdictions worldwide; though some devices also use 784 MHz in China, 868 MHz in Europe and 915 MHz in the US and Australia, however even those regions and countries still use 2.4 GHz for most commercial Zigbee devices for home use. Data rates vary from 20 kbit/s (868 MHz band) to 250 kbit/s (2.4 GHz band).

Zigbee builds on the physical layer and media access control defined in IEEE standard 802.15.4 for low-rate wireless personal area networks (WPANs). The specification includes four additional key components: network layer, application layer, *Zigbee Device Objects* (ZDOs) and manufacturer-defined application objects. ZDOs are responsible for some tasks, including keeping track of device roles, managing requests to join a network, as well as device discovery and security.

The Zigbee network layer natively supports both star and tree networks, and generic mesh networking. Every network must have one coordinator device. Within star networks, the coordinator must be the central node. Both trees and meshes allow the use of Zigbee routers to extend communication at the network level. Another defining feature of Zigbee is facilities for carrying out secure communications, protecting establishment and transport of cryptographic keys, ciphering frames, and controlling device. It builds on the basic security framework defined in IEEE 802.15.4.

2.4.1 History

Zigbee-style self-organizing ad hoc digital radio networks were conceived in the 1990s. The IEEE 802.15.4-2003 Zigbee specification was ratified on December 14, 2004. The Zigbee Alliance announced availability of Specification 1.0 on June 13, 2005, known as the *ZigBee 2004 Specification*.

Cluster library

In September 2006, the *Zigbee 2006 Specification* was announced, obsoleting the 2004 stack· The 2006 specification replaces the message and key–value pair structure used in the 2004 stack with a *cluster library*. The library is a set of standardised commands, attributes and global artifacts organised under groups known as clusters with names such as Smart Energy, Home Automation, and Zigbee Light Link.

In January 2017, Zigbee Alliance renamed the library to *Dotdot* and announced it as a new protocol to be represented by an emoticon (||). They also announced it will now additionally run over other network types using Internet Protocol and will interconnect with other standards such as Thread. Since its unveiling, Dotdot has functioned as the default application layer for almost all Zigbee devices.

Zigbee Pro

Zigbee Pro, also known as Zigbee 2007, was finalized in 2007. A Zigbee Pro device may join and operate on a legacy Zigbee network and vice versa. Due to differences in routing options, Zigbee Pro devices must become non-routing Zigbee end devices (ZEDs) on a legacy Zigbee network, and legacy Zigbee devices must become ZEDs on a Zigbee Pro network. It operates using the 2.4 GHz ISM band, and adds a sub-GHz band.

2.4.2 Use Cases

Zigbee protocols are intended for embedded applications requiring low power consumption and tolerating low data rates. The resulting network will use very little power—individual devices must have a battery life of at least two years to pass certification.

Typical application areas include:

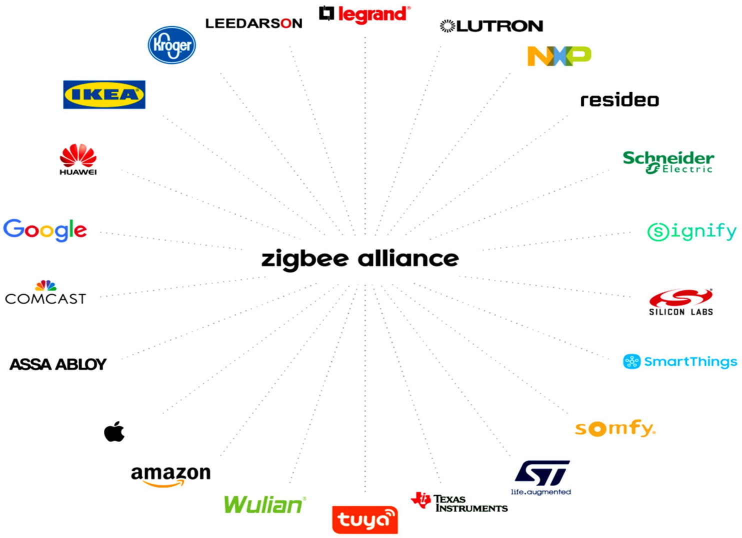
- Home automation
- Wireless sensor networks
- Industrial control systems
- Embedded sensing
- Medical data collection
- Smoke and intruder warning
- Building automation
- Remote wireless microphone configuration

Zigbee is not for situations with high mobility among nodes. Hence, it is not suitable for tactical ad hoc radio networks in the battlefield, where high data rate and high mobility is present and needed.

2.4.3 Zigbee Alliance

Established in 2002, the Zigbee Alliance is a group of companies that maintain and publish the Zigbee standard. The name Zigbee is a registered trademark of this group, and is not a single technical standard. The organization publishes application profiles that allow multiple OEM vendors to create interoperable products. The relationship between IEEE 802.15.4 and Zigbee is similar to that between IEEE 802.11 and the Wi-Fi Alliance.

Over the years, the Alliance's membership has grown to over 500 companies, including the likes of Comcast, Ikea, Legrand, Samsung SmartThings, and Amazon. The Zigbee Alliance has three levels of membership: adopter, participant, and promoter. The adopter members are allowed access to completed Zigbee specifications and standards, and the participant members have voting rights, play a role in Zigbee development, and have early access to specifications and standards for product development.



The requirements for membership in the Zigbee Alliance cause problems for free-software developers because the annual fee conflicts with the GNU General Public Licence. The requirements for developers to join the Zigbee Alliance also conflict with most other free-software licenses. The Zigbee Alliance board of directors has been asked to make their license compatible with GPL, but refused. Bluetooth has GPL-licensed implementations.

2.4.4 Application Profiles

The first Zigbee application profile, Home Automation, was announced November 2, 2007. Additional application profiles have since been published.

The Zigbee Smart Energy 2.0 specifications define an Internet Protocol-based communication protocol to monitor, control, inform, and automate the delivery and use of energy and water. It is an enhancement of the Zigbee Smart Energy version 1 specifications. It adds services for plug-in electric vehicle charging, installation, configuration and firmware download, prepay services, user information and messaging, load control, demand response and common information and application profile interfaces for wired and wireless networks. It is being developed by partners including:

- HomeGrid Forum responsible for marketing and certifying ITU-T G.hn technology and products
- HomePlug Powerline Alliance
- International Society of Automotive Engineers SAE International

- IPSO Alliance
- SunSpec Alliance
- Wi-Fi Alliance

Zigbee Smart Energy relies on Zigbee IP, a network layer that routes standard IPv6 traffic over IEEE 802.15.4 using 6LoWPAN header compression.

In 2009, the Radio Frequency for Consumer Electronics Consortium (RF4CE) and Zigbee Alliance agreed to deliver jointly a standard for radio frequency remote controls. Zigbee RF4CE is designed for a broad range of consumer electronics products, such as TVs and set-top boxes. It promised many advantages over existing remote control solutions, including richer communication and increased reliability, enhanced features and flexibility, interoperability, and no line-of-sight barrier. The Zigbee RF4CE specification uses a subset of Zigbee functionality allowing to run on smaller memory configurations in lower-cost devices, such as remote control of consumer electronics.

2.4.5 Radio Hardware

The radio design used by Zigbee has few analog stages and uses digital circuits wherever possible. Products that integrate the radio and microcontroller into a single module are available.

The Zigbee qualification process involves a full validation of the requirements of the physical layer. All radios derived from the same validated semiconductor mask set would enjoy the same RF characteristics. Zigbee radios have very tight constraints on power and bandwidth. An uncertified physical layer that malfunctions can increase the power consumption of other devices on a Zigbee network. Thus, radios are tested with guidance given by Clause 6 of the 802.15.4-2006 Standard.

This standard specifies operation in the unlicensed 2.4 to 2.4835 GHz (worldwide), 902 to 928 MHz (Americas and Australia) and 868 to 868.6 MHz (Europe) ISM bands. Sixteen channels are allocated in the 2.4 GHz band, spaced 5 MHz apart, though using only 2 MHz of bandwidth each. The radios use direct-sequence spread spectrum coding, which is managed by the digital stream into the modulator. Binary phase-shift keying (BPSK) is used in the 868 and 915 MHz bands, and offset quadrature phase-shift keying (OQPSK) that transmits two bits per symbol is used in the 2.4 GHz band.

The raw, over-the-air data rate is 250 kbit/s per channel in the 2.4 GHz band, 40 kbit/s per channel in the 915 MHz band, and 20 kbit/s in the 868 MHz band. The actual data throughput will be less than the maximum specified bit rate due to the packet overhead and processing delays. For indoor applications at 2.4 GHz transmission distance is 10–20 m, depending on the construction materials, the number of walls to be penetrated and the output power permitted in that geographical location. The output power of the radios is generally 0–20 dBm (1–100 mW).

2.4.6 Device Types and Operating Modes

There are three classes of Zigbee devices:

- *Zigbee coordinator (ZC)*: The most capable device, the coordinator forms the root of the network tree and may bridge to other networks. There is precisely one Zigbee coordinator in each network since it is the device that started the network originally (the Zigbee LightLink specification also allows operation without a Zigbee coordinator, making it more usable for off-the-shelf home products). It stores information about the network, including acting as the trust center and repository for security keys.
- *Zigbee router (ZR)*: As well as running an application function, a router can act as an intermediate router, passing data on from other devices.
- *Zigbee end device (ZED)*: Contains just enough functionality to talk to the parent node (either the coordinator or a router); it cannot relay data from other devices. This relationship allows the node to be asleep a significant amount of the time thereby giving long battery life. A ZED requires the least amount of memory and thus can be less expensive to manufacture than a ZR or ZC.

The current Zigbee protocols support beacon-enabled and non-beacon-enabled networks. In non-beacon-enabled networks, an unslotted CSMA/CA channel access mechanism is used. In this type of network, Zigbee routers typically have their receivers continuously active, requiring additional power. However, this allows for heterogeneous networks in which some devices receive continuously while others transmit when necessary. The typical example of a heterogeneous network is a wireless light switch: The Zigbee node at the lamp may constantly receive since it is reliably powered by the mains supply to the lamp, while a battery-powered light switch would remain asleep until the switch is thrown. In which case, the switch wakes up, sends a command to the lamp, receives an acknowledgment, and returns to sleep. In such a network the lamp node will be at least a Zigbee router, if not the Zigbee coordinator; the switch node is typically a Zigbee end device. In beacon-enabled networks, Zigbee routers transmit periodic beacons to confirm their presence to other network nodes. Nodes may sleep between beacons, thus extending their battery life. Beacon intervals depend on data rate; they may range from 15.36 milliseconds to 251.65824 seconds at 250 kbit/s, from 24 milliseconds to 393.216 seconds at 40 kbit/s and from 48 milliseconds to 786.432 seconds at 20 kbit/s. Long beacon intervals require precise timing, which can be expensive to implement in low-cost products.

In general, the Zigbee protocols minimize the time the radio is on, so as to reduce power use. In beaconing networks, nodes only need to be active while a beacon is being transmitted. In non-beacon-enabled networks, power consumption is decidedly asymmetrical: Some devices are always active while others spend most of their time sleeping.

Except for Smart Energy Profile 2.0, Zigbee devices are required to conform to the IEEE 802.15.4-2003 Low-rate Wireless Personal Area Network (LR-WPAN) standard. The standard specifies the lower protocol layers—the physical layer (PHY), and the media access control portion of the data link layer. The basic channel access mode is carrier-sense multiple access with collision avoidance (CSMA/CA). That is, the nodes communicate in a way somewhat analogous to how humans converse: a node briefly checks to see that other nodes are not talking before it starts. CSMA/CA is not used in three notable exceptions:

- Message acknowledgments
- Beacons are sent on a fixed-timing schedule.
- Devices in beacon-enabled networks that have low-latency, real-time requirements may also use guaranteed time slots.

Network layer

The main functions of the network layer are to assure correct use of the MAC sublayer and provide a suitable interface for use by the next upper layer, namely the application layer. The network layer deals with network functions such as connecting, disconnecting, and setting up networks. It can establish a network, allocate addresses, and add and remove devices. This layer makes use of star, mesh and tree topologies.

The data entity of the transport layer creates and manages protocol data units at the direction of the application layer and performs routing according to the current topology. The control entity handles the configuration of new devices and establishes new networks. It can determine whether a neighboring device belongs to the network and discovers new neighbors and routers.

The routing protocol used by the network layer is AODV. To find a destination device, AODV is used to broadcast a route request to all of its neighbors. The neighbors then broadcast the request to their neighbors and onward until the destination is reached. Once the destination is reached, a route reply is sent via unicast transmission following the lowest cost path back to the source. Once the source receives the reply, it updates its routing table with the destination address of the next hop in the path and the associated path cost.

Application layer

The application layer is the highest-level layer defined by the specification and is the effective interface of the Zigbee system to its end users. It comprises the majority of components added by the Zigbee specification: both ZDO (Zigbee device object) and its management procedures, together with application objects defined by the manufacturer, are considered part of this layer. This layer binds tables, sends messages between bound devices, manages group addresses, reassembles packets and also transports data. It is responsible for providing service to Zigbee device profiles.

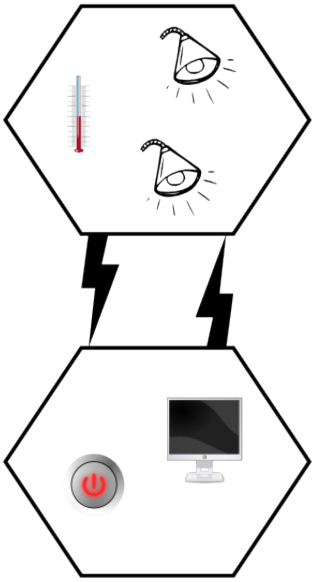
Main components

The *ZDO* (Zigbee device object), a protocol in the Zigbee protocol stack, is responsible for overall device management, security keys, and policies. It is responsible for defining the role of a device as either coordinator or end device, as mentioned above, but also for the discovery of new devices on the network and the identification of their offered services. It may then go on to establish secure links with external devices and reply to binding requests accordingly.

The application support sublayer (APS) is the other main standard component of the stack, and as such it offers a well-defined interface and control services. It works as a bridge between the network layer and the other elements of the application layer: it keeps up-to-date binding tables in the form of a database, which can be used to find appropriate devices depending on the services that are needed and those the different devices offer. As the union between both specified layers, it also routes messages across the layers of the protocol stack.

Communication models

An application may consist of communicating objects which cooperate to carry out the desired tasks. Tasks will typically be largely local to each device, for instance, the control of each household appliance. The focus of Zigbee is to distribute work among many different devices which reside within individual Zigbee nodes which in turn form a network.



Zigbee high-level communication model

The objects that form the network communicate using the facilities provided by APS, supervised by ZDO interfaces. Within a single device, up to 240 application objects can exist, numbered in the range 1–240. 0 is reserved for the ZDO data interface and 255 for broadcast; the 241-254 range is not currently in use but may be in the future.

Two services are available for application objects to use (in Zigbee 1.0):

- The *key-value pair service* (KVP) is meant for configuration purposes. It enables description, request and modification of object attribute through a simple interface based on get, set and event primitives, some allowing a request for a response. Configuration uses XML.
- The *message service* is designed to offer a general approach to information treatment, avoiding the necessity to adapt application protocols and potential overhead incurred by KVP. It allows arbitrary payloads to be transmitted over APS frames.

Addressing is also part of the application layer. A network node consists of an IEEE 802.15.4-conformant radio transceiver and one or more device descriptions (collections of attributes that can be polled or set, or can be monitored through events). The transceiver is the basis for addressing, and devices within a node are specified by an *endpoint identifier* in the range 1 to 240.

Communication and device discovery

For applications to communicate, their comprising devices must use a common application protocol (types of messages, formats and so on); these sets of conventions are grouped in *profiles*. Furthermore, binding is decided upon by matching input and output cluster identifiers, unique within the context of a given profile and associated to an incoming or outgoing data flow in a device. Binding tables contain source and destination pairs.

Depending on the available information, device discovery may follow different methods. When the network address is known, the IEEE address can be requested using unicast communication. When it is not, petitions are broadcast (the IEEE address being part of the response payload). End devices will simply respond with the requested address while a network coordinator or a router will also send the addresses of all the devices associated with it.

This extended discovery protocol permits external devices to find out about devices in a network and the services that they offer, which endpoints can report when queried by the discovering device (which has previously obtained their addresses). Matching services can also be used.

The use of cluster identifiers enforces the binding of complementary entities using the binding tables, which are maintained by Zigbee coordinators, as the table must always be available within a network and coordinators are most likely to have a permanent power supply. Backups, managed by higher-level layers, may be needed by some applications. Binding requires an established communication link; after it exists, whether to add a new node to the network is decided, according to the application and security policies.

Communication can happen right after the association. Direct addressing uses both radio address and endpoint identifier, whereas indirect addressing uses every relevant field (address, endpoint, cluster, and attribute) and requires that they are sent to the network coordinator, which maintains associations and translates requests for communication. Indirect addressing is particularly useful to keep some devices very simple and minimize their need for storage. Besides these two methods, *broadcast* to all endpoints in a device is available, and group addressing is used to communicate with groups of endpoints belonging to a set of devices.

2.4.7 Security Services

As one of its defining features, Zigbee provides facilities for carrying out secure communications, protecting establishment and transport of cryptographic keys, cyphering frames, and controlling devices. It builds on the basic security framework defined in IEEE 802.15.4. This part of the architecture relies on the correct management of symmetric keys and the correct implementation of methods and security policies.

Basic security model

The basic mechanism to ensure confidentiality is the adequate protection of all keying material. Trust must be assumed in the initial installation of the keys, as well as in the processing of security information. For an implementation to globally work, its general conformance to specified behaviors is assumed.

Keys are the cornerstone of the security architecture; as such their protection is of paramount importance, and keys are never supposed to be transported through an insecure channel. A momentary exception to this rule occurs during the initial phase of the addition to the network of a previously unconfigured device. The Zigbee network model must take particular care of security considerations, as ad hoc networks may be physically accessible to external devices. Also the state of the working environment cannot be predicted.

Within the protocol stack, different network layers are not cryptographically separated, so access policies are needed, and conventional design assumed. The open trust model within a device allows for key sharing, which notably decreases potential cost. Nevertheless, the layer which creates a frame is responsible for its security. If malicious devices may exist, every network layer payload must be ciphered, so unauthorized traffic can be immediately cut off. The exception, again, is the transmission of the network key, which confers a unified security layer to the grid, to a new connecting device.

Security Architecture

Zigbee uses 128-bit keys to implement its security mechanisms. A key can be associated either to a network, being usable by both Zigbee layers and the MAC sublayer, or to a link, acquired through pre-installation, agreement or transport. Establishment of link keys is based on a master key which controls link key correspondence. Ultimately, at least, the initial master key must be obtained through a secure medium (transport or pre-installation), as the security of the whole network depends on it. Link and master keys are only visible to the application layer. Different services use different one-way variations of the link key to avoid leaks and security risks.

Key distribution is one of the most important security functions of the network. A secure network will designate one special device which other devices trust for the distribution of security keys: the trust center. Ideally, devices will have the center trust address and initial master key preloaded; if a momentary vulnerability is allowed, it will be sent as described above. Typical applications without special security needs will use a network key provided by the trust center (through the initially insecure channel) to communicate.

Thus, the trust center maintains both the network key and provides point-to-point security. Devices will only accept communications originating from a key supplied by the trust center, except for the initial master key. The security architecture is distributed among the network layers as follows:

- The MAC sublayer is capable of single-hop reliable communications. As a rule, the security level it is to use is specified by the upper layers.
- The network layer manages routing, processing received messages and being capable of broadcasting requests. Outgoing frames will use the adequate link key according to the routing if it is available; otherwise, the network key will be used to protect the payload from external devices.
- The application layer offers key establishment and transport services to both ZDO and applications.

The security levels infrastructure is based on CCM*, which adds encryption- and integrity-only features to CCM. According to the German computer e-magazine *Heise Online*, Zigbee Home Automation 1.2 is using fallback keys for encryption negotiation which are known and cannot be changed. This makes the encryption highly vulnerable

2.4.8 Radio-Frequency Identification (RFID)

Radio Frequency Identification (RFID) refers to a wireless system comprised of two components: tags and readers. The reader is a device that has one or more antennas that emit radio waves and receive signals back from the RFID tag. Tags, which use radio waves to communicate their identity and other information to nearby readers, can be passive or active. Passive RFID tags are powered by the reader and do not have a battery. Active RFID tags are powered by batteries.

RFID tags can store a range of information from one serial number to several pages of data. Readers can be mobile so that they can be carried by hand, or they can be mounted on a post or overhead. Reader systems can also be built into the architecture of a cabinet, room, or building.

Radio-frequency identification (RFID) uses electromagnetic fields to automatically identify and track tags attached to objects. An RFID system consists of a tiny radio transponder, a radio receiver and transmitter. When triggered by an electromagnetic interrogation pulse from a nearby RFID reader device, the tag transmits digital data, usually an identifying inventory number, back to the reader. This number can be used to track inventory goods.

Passive tags are powered by energy from the RFID reader's interrogating radio waves. Active tags are powered by a battery and thus can be read at a greater range from the RFID reader, up to hundreds of meters. Unlike a barcode, the tag does not need to be within the line of sight of the reader, so it may be embedded in the tracked object. RFID is one method of automatic identification and data capture (AIDC).

RFID tags are used in many industries. For example, an RFID tag attached to an automobile during production can be used to track its progress through the assembly line, RFID-tagged pharmaceuticals can be tracked through warehouses, and implanting RFID microchips in livestock and pets enables positive identification of animals. Tags can also be used in shops to expedite checkout, and to prevent theft by customers and employees.

Since RFID tags can be attached to physical money, clothing, and possessions, or implanted in animals and people, the possibility of reading personally-linked information without consent has raised serious privacy concerns. These concerns resulted in standard specifications development addressing privacy and security issues.

2.4.9 Uses

RFID systems use radio waves at several different frequencies to transfer data. In health care and hospital settings, RFID technologies include the following applications:

- Inventory control
- Equipment tracking
- Out-of-bed detection and fall detection
- Personnel tracking
- Ensuring that patients receive the correct medications and medical devices
- Preventing the distribution of counterfeit drugs and medical devices
- Monitoring patients
- Providing data for electronic medical records systems

The FDA is not aware of any adverse events associated with RFID. However, there is concern about the potential hazard of electromagnetic interference (EMI) to electronic medical devices from radio frequency transmitters like RFID. EMI is a degradation of the performance of equipment or systems (such as medical devices) caused by an electromagnetic disturbance.

RFID Use-Case Example

One of the more common uses of RFID technology is through the microchipping of pets or pet chips. These microchips are implanted by veterinarians and contain information pertaining to the pet including their name, medical records, and contact information for their owners. If a pet goes missing and is turned into a rescue or shelter, the shelter worker scans the animal for a microchip. If the pet has a microchip, the shelter worker will only be a quick phone call or internet search away from being able to contact the pet's owners. Pet chips are thought to be more reliable than collars, which can fall off or be removed.

With the rise of accessibility of the technology, most veterinarians and shelters now have the technology to read these microchips. Universal scanners and national databases for storing owner information are also rising in popularity, making it easier than ever for microchipping pets to be a successful way to get lost pets reunited with their owners. One downside of the device is that the records must be kept up to date. The information is only as reliable as what is being imputed by the person setting up the microchip.

Information for Health Care Professionals

Because this technology continues to evolve and is more widely used, it is important to keep in mind its potential for interference with pacemakers, implantable cardioverter defibrillators (ICDs), and other electronic medical devices.

Physicians should stay informed about the use of RFID systems. If a patient experiences a problem with a device, ask questions that will help determine if RFID might have been a factor, such as when and where the episode occurred, what the patient was doing at the time, and whether or not the problem resolved once the patient moved away from that environment. If you suspect that RFID was a factor, device interrogation might be helpful in correlating the episode to the exposure. Report any suspected medical device malfunctions to MedWatch, FDA's voluntary adverse event reporting system.

FDA Actions

The FDA has taken steps to study RFID and its potential effects on medical devices including:

- Working with manufacturers of potentially susceptible medical devices to test their products for any adverse effects from RFID and encouraging them to consider RFID interference when developing new devices.
- Working with the RFID industry to better understand, where RFID can be found, what power levels and frequencies are being used in different locations, and how to best mitigate potential EMI with pacemakers and ICDs.
- Participating in and reviewing the development of RFID standards to better understand RFID's potential to affect medical devices and to mitigate potential EMI.
- Working with the Association for Automatic Identification and Mobility (AIM) to develop a way to test medical devices for their vulnerability to EMI from RFID systems.
- Collaborating with other government agencies, such as the Federal Communications Commission (FCC), the National Institute for Occupational Safety and Health (NIOSH) and the Occupational

Safety and Health Administration (OSHA) to better identify places where RFID readers are in use.

2.5 CLUSTERING IN WIRELESS MULTIMEDIA SENSOR NETWORKS

Recent developments in wireless communication and embedded technology have made the wireless sensor network (WSN) possible. Wireless sensor networks are constituted of large number of low-cost, low-power and less communication bandwidth tiny sensor nodes. The sensors, which are randomly deployed in an environment, are required to collect data from their surroundings, process the data and finally send it to the sink through multi hops. Traditional WSNs collects the scalar data such as temperature, pressure, etc. and transmit it to the sink. WSN has potential to design many new applications for handling emergency, military and disaster relief operations that require real time information for efficient coordination and planning.

Wireless multimedia sensor network (WMSN) uses cheap CMOS (Complementary Metal Oxide Semiconductor) camera and microphone sensors which can acquire multimedia information. WMSN consists of camera sensors as well as scalar sensors. The multimedia content has the potential to enhance the level of information collected, compared with scalar data. Multimedia content produces immense amount of data to transmit over WMSN, which is limited in terms of power supply, communication bandwidth, memory, etc. In a large-scale network, if all the nodes have to communicate their data to their respective destination, it will deplete their energy quickly due to the long-distance, large volume of data and multi-hopnature of the communication. This will also lead to network contention. The clustering is a standard approach for achieving efficient and scalable control in these networks.

Clustering results in a number of benefits. It facilitates distribution of control over the network. It saves energy and reduces network contention by enabling locality of communication. Nodes communicate their data over shorter distances to their respective cluster head (CH). The cluster head aggregates these data into a smaller set of meaningful information. Not all nodes, but only the cluster heads need to communicate with their neighbouring cluster heads and sink/base station. Figure 4 shows the clustering of nodes in a general WSN. You have utilized spectral graph partitioning (SGP) technique based upon eigenvalues proposed by

Fiedler to form clustering in WMSN. SGP method has been used in many applications such as image segmentation, social networks, etc.

The spectral graph partitioning (SGP) algorithm is based on second highest eigenvalues of particular graph.

The second highest eigenvalue of the Laplacian matrix corresponding to different eigenvectors, is used to partition the graph into two parts. Within a cluster, a node with highest eigenvalue is selected as cluster head. In case of WMSN, large volume of sensed data is generated, therefore, such clustering can be utilized to reduce the volume and number of data transmissions through data aggregation. Simulation experiments have been performed to evaluate the performance of proposed method and compare it with the existing technique.

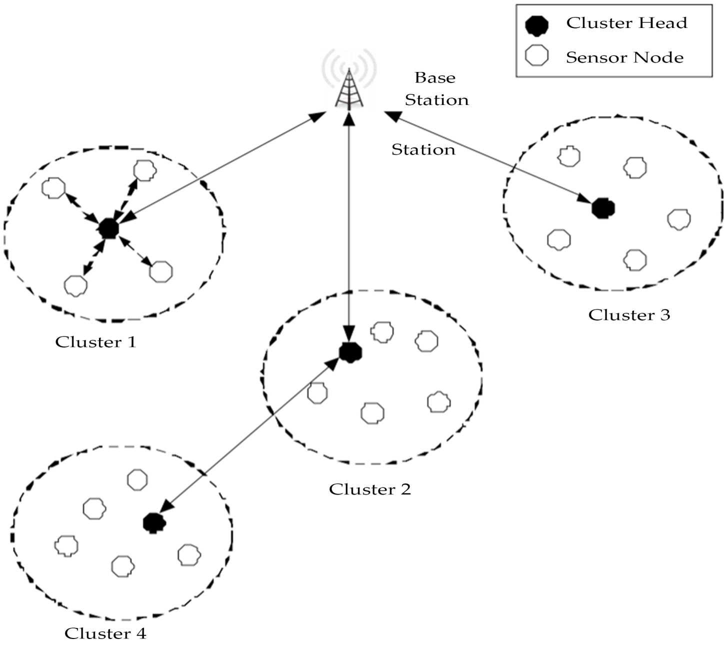


Figure 4. Clustering of SNs in WSN.

The second highest eigenvalue of the Laplacian matrix corresponding to different eigenvectors, is used to partition the graph into two parts. Within a cluster, a node with highest eigenvalue is selected as cluster head. In case of WMSN, large volume of sensed data is generated, therefore, such clustering can be utilized to reduce the volume and number of data transmissions through data aggregation. Simulation experiments have been performed to evaluate the performance of proposed method and compare it with the existing technique.

2.5.1 Related Work

The nodes are often grouped together into disjoint and mostly non-overlapping groups are called clusters. Clusters are used to minimize communication latency and improve energy efficiency. Leader of every cluster is often called the cluster head (CH) and generally has to perform more functions as compared to normal sensor node.

The suggested voting-based clustering algorithm (VCA) that enhances the criteria for cluster selection and combines load balancing consideration together with topology and energy information. VCA addresses inefficient cluster formation using a voting scheme, which enables the nodes to exchange information about their local network view. This method assumes synchronization among the nodes. Similar to WCA, the time required for the nodes to gather information about all other nodes depends on the network size and is not constant. The suggested spectral classification based on near optimal clustering in wireless sensor networks (SCNOPC-WSN) algorithm. This algorithm deals with the clustering problem in WSN. Energy aware adaptive clustering protocol is used for the bi-partitioning spectral classification and it guarantees robust clustering. SCNOPC-WSN also deals with the optimization of the energy dissipated in the network.

The suggested hierarchical clustering algorithm based on geometric properties of the wireless network. A number of cluster properties such as cluster size and the degree of overlap, which are useful for the management and scalability of the hierarchy, are also considered while grouping the nodes. In the proposed scheme, any node in the WSN can initiate the cluster formation process. Initiator with least node ID will take precedence, if multiple nodes started cluster formation process at the same time.

Bandyopadhyay and Coyle proposed EEHC which is a distributed, randomized clustering algorithm for WSNs with the objective of maximizing the network lifetime. CHs collect the sensor reading in their individual clusters and send an aggregated report to the base station. Their technique is based on two stages—initial and extended. The hot-spot problem in multihop networks is solved using cluster with unequal size. CHs that are closed to the base station tend to die faster, because they relay much more traffic than remote nodes. Setting smaller cluster sizes to the close CHs preserves their energy. Using a separation between the data gathering and aggregation task and the forwarding task. Spectral graph partitioning algorithm partitions the graph using the eigenvectors of the matrix obtained from the graph. SGP obtains data representation in the low-dimensional space that can be easily clustered. Eigenvalues and eigenvectors provide a penetration into the connectivity of the graph.

2.5.2 SGP for Cluster Formation

Spectral graph partitioning technique is based on eigenvalues and eigenvector of the adjacency matrix of graph to partition the graph. The methods are called spectral, because they make use of the spectrum of the adjacency matrix of the data to cluster the points. Spectral methods are widely applied for graph partitioning. Spectral graph partitioning is a powerful technique and also is being used in image segmentation and social network analysis. SGP divides the graph into two disjoint groups, based on eigenvectors corresponding to the second smallest eigenvalue of the Laplacian matrix.

Let G (V, E) is an undirected graph where V represents the set of vertices (sensor nodes) and E represents the set of edges connecting these vertices. Each vertex is identified by an index $i \in \{1, 2, \cdots, N\}$. The edge between node i and node j is represented by e_{ij}. The graph can be represented as an adjacency matrix. The adjacency matrix A of graph G having N nodesis the N × N matrix where the non-diagonal entry a_{ij} is the number of edges from node i to node j, and the diagonal entry a_{ii} is the number of loops at node i. The adjacency matrix is symmetric for undirectedgraphs.

The adjacency matrix A is defined as

$$A = \left[a_{ij} \right] = \begin{cases} 1 & \text{edge weight between node } i \text{ and node } j \\ 0 & \text{otherwise} \end{cases}$$

We also define degree matrix D for graph G. The degree matrix of a graph gives the number of edges between node i to another node. The degree matrix is a diagonal matrix which contains information about the degree of each node. It is helpful to construct the Laplacian matrix of a graph. The degree matrix D for G is a N × N square matrix and is defined as

$$D = \left[\deg_{ij} \right] = \begin{cases} \text{total weight of edges incident to node } i \\ 0 \end{cases}$$

The Laplacian matrix is formed from adjacency matrix and the degree matrix. The Laplacian matrix of the graph G having N vertices is N × N square matrix and is represented as

L=D-A

The normalized form of Laplacian matrix can be written as

$$\Upsilon(i,j)=\begin{cases}1 & \text{if } i=j \text{ and } \deg_j \neq 0\\ -\dfrac{1}{\sqrt{\deg_i \deg_j}} & \text{if } i \text{ and } j \text{ are adjacent}\\ 0 & \text{otherwise}\end{cases}$$

The eigenvalues of matrix ϒ are denoted by, λ_i i= Λ N such that $\lambda_1 \leq \lambda_2 \leq \ldots\ldots \leq \lambda_n$ Laplacian matrix has the property $\Upsilon \cdot X = \lambda \cdot X$ where X is the eigenvector of the matrix and λ is the eigenvalue of the matrix. Laplacian matrix plays important role in spectral graph theory. λ_1 represents the number of subgraphs in the network. The second smallest eigenvalue λ_2 is referred to the algebraic connectivity and its corresponding eigenvector is usually referred to as the Fiedler Vector.

We choose the eigenvector values corresponding to the second highest eigenvalue λ_2. Second highest eigenvalue (λ_2) divides the graph into two subgraphs. G is divided into two subgraphs G^- and G^+, where G^+ and are the set of vertices related to the new subgraphs. contains nodes corresponding to positive eigenvalues and G^- contains nodes corresponding to negative eigenvalues. The set of vertices is defined by

$$N = N^+ \bigcup N^-$$

and

$$N^+ \bigcap N^- = \phi$$

where $N = |G|$, $N^+ = |G^+|$ and $N^- = |G^-|$.

2.5.3 Cluster Formation In WMSN

SGP technique can be used for dividing the network into clusters. SGP has many advantages as compared to other clustering algorithms. SGP partitions the graph on the basis of eigenvalues and eigenvector adjacency matrix. If the graph is partitioned into more than two subgraphs, apply SGP technique recursively. These properties make SGP technique a better option for multimedia data clustering where large volume of data is transmitted between nodes and CH.

In our proposed method, clustering of WMSN has been done on the basis of Spectral Graph Partitioning technique. Each node sends short message to sink which contains the location information of the node. On the basis of this information, the sink constructs the adjacency matrix and degree matrix and then constructs the Laplacian matrix. The eigenvector corresponding to second smallest eigenvalue (called Fiedler Vector) is used to partition the WMSN. The location of each node may be found by GPS or any other localization method.

2.5.4 Steps for Clustering

1) Construct a graph G of the given sensor network.
2) Construct the normalized Laplacian matrix as

$$\Upsilon(i,j) = \begin{cases} 1 & \text{if } i = j \text{ and } \deg_j \neq 0 \\ -\dfrac{1}{\sqrt{\deg_i \deg_j}} & \text{if } i \text{ and } j \text{ are adjacent} \\ 0 & \text{otherwise} \end{cases}$$

Where $\deg_i$ is the degree of node i

3) From the Laplacian matrix ϒ of the graph compute the eigenvalues and eigenvector of Laplacian matrix.
4) Select the second smallest eigenvalue λ_2 of Laplacian matrix ϒ.
5) Choose eigenvector value corresponding to the eigenvalue λ_2.

Divide the graph G into two subgraphs G^+ and G^- where G^+ contains nodes corresponding to positive eigenvalues and G^- , contains nodes corresponding to negative eigenvalues.

After the first iteration of above method, the whole network is divided into two clusters based on the eigen values of the node. After first iteration cluster 1 contains all the nodes with positive eigenvector values and another cluster 2 contains nodes having negative eigenvector values. Cluster 1 has five nodes with positive value of eigenvector and the nodes are A, B, C, D and F. Cluster 2 has five nodes that have negative eigenvector values and the nodes are E, G, H, I and J.

Only two clusters are formed in first iteration. The larger size clusters can be further divided into two different clusters by applying the algorithm recursively. This process continues until maximum intra-node distance within a cluster is less than $R/2\sqrt{2}$ where R is the transmission range of the sensor node. When intranode distance is remaining $R/2\sqrt{2}$ two nodes in neighbouring clusters can communicate in one hop. After applying the algorithm recursively, the given network is divided into four clusters

It has been observed that both the clusters have higher intra-node distance than $R/2\sqrt{2}$, so apply the algorithm to both the clusters. After applying the algorithm cluster 1 is portioned into two different clusters. This algorithm is also applied to cluster 2

2.5.5 Cluster Head Election

The clustering algorithm divides the whole network into clusters. The next step is election of cluster head for each cluster. As per the property of SGP, the least eigenvector value of node signifies that the node is well connected to the other nodes within the cluster as well as it is connected to cluster.

For initial cluster head election, we chose the least eigenvector value among the nodes within cluster, Table 4 represents the eigenvector values of the cluster and Table 5 shows the elected cluster heads in different clusters on the basis of eigenvector values. Therefore, we compare the eigenvector values of the cluster and choose the least eigenvector node as a cluster head.

$$\text{Cluster Head} = \text{Least}\left|\text{Eigenvector}\right|$$

Cluster head rotation must take place when residual energy (E_{res}) of the cluster head node falls below the threshold value (E_{th}). The present cluster head declares the election process by sending a message that contains its E_{res} to all the cluster members. The cluster members whose residual energy is greater than E_{res} responds to this message by sending the residual energy to the cluster head.

The new cluster head is elected based upon CH Candidacy Factor (CF) defined as

$$CF_i = \frac{E^i_{res}}{D_i}$$

Where E^i_{res} is the residual energy of node i, Di is the distance between node i and current cluster head. If (x_{ch}, y_{ch}) and (x_i, y_i) are the location coordinates of current cluster head and node i, respectably, then

$$D_i = \sqrt{(x_{ch} - x_i) + (y_{ch} - y_i)}$$

A node with highest value of CF is elected as next cluster head.

SUMMARY

- The conventional power transmission system uses transmission lines to carry the power from one place to another, but it is costlier in terms of cable costs and also there exists a certain transmission loss.
- Wireless communication (or just wireless, when the context allows) is the transfer of information between two or more points that do not use an electrical conductor as a medium by which to perform the transfer.
- Inductive coupling method is the most important methods transferring energy wirelessly through inductive coupling.
- The largest application of the WPT is the production of power by placing satellites with giant solar arrays in Geosynchronous Earth Orbit and transmitting the power as microwaves to the earth known as Solar Power Satellites (SPS).
- Wireless network is a network set up by using radio signal frequency to communicate among computers and other network devices. Sometimes it is also referred to as WiFi network or WLAN.
- Bluetooth is a short-range wireless technology standard that is used for exchanging data between fixed and mobile devices over short distances using UHF radio waves in the ISM bands, from 2.402 GHz to 2.48 GHz, and building personal area networks (PANs).
- The Wi-Fi Alliance began using a consumer-friendly generation numbering scheme for the publicly used 802.11 protocols.
- Zigbee is a wireless technology developed as an open global standard to address the unique needs of low-cost, low-power wireless IoT networks.
- Wireless multimedia sensor network (WMSN) uses cheap CMOS (Complementary Metal Oxide Semiconductor) camera and microphone sensors which can acquire multimedia information.

REFERENCES

1. A. Bertrand and M. Moonen, "Distributed Computation of the Fiedler Vector with Application to Topology Inference in Ad Hoc Networks," Internal Report KU Leuven ESAT-SCD, 2012.
2. A. Savvides, C. Han and M. B. Srivastva, "Dynamic FineGrained Localization in Ad Hoc Networks of Sensors," 7th International Conference on Mobile Computing and Networking (MOBICOM), 2001, pp. 166-179.
3. A. Savvides, C. Han and M. B. Srivastva, "Dynamic FineGrained Localization in Ad Hoc Networks of Sensors," 7th International Conference on Mobile Computing and Networking (MOBICOM), 2001, pp. 166-179.
4. B. Elbhiri, S. El Fkihi, R. Saadane and D. Aboutajdine, "Clustering in Wireless Sensor Network Based on Near Optimal Bi-partitions," 6th EURO-NF Conference on Next Generation Internet (NGI), 2010, pp. 1 -6.
5. C. Li, M. Ye, G. Chen and J. Wu, "An Energy Efficient Unequal Clustering Mechanism for Wireless Sensor Networks," 2nd IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS), 2005, pp. 125-132.
6. F. Akyldiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, "Wireless Sensor Networks: A Survey," Computer Networks, Vol. 38, No. 4, 2002, pp. 393-422.
7. I. F. Akyildiz, T. Melodia and K. R. Chowdhury, "A Survey on Wireless Multimedia Sensor Networks," Computer Networks, Vol. 51, No. 4, 2007, pp. 921-960.
8. M. Chatterjee, S. K. Das and D. Turgut, "WCA: A Weighted Clustering Algorithm for Mobile Ad hoc Networks," Journal of Cluster Computing (Special Issue on Mobile Ad hoc Networks), Vol. 5, No. 2, 2002, pp. 193-204.
9. M. Demirbas, A. Arora, V. Mittal and V. Kulathumani, "A Fault-Local Self-Stabilizing Clustering Service for Wireless Ad Hoc Networks," IEEE Transactions on Parallel and Distributed Systems, Vol. 17, No. 9, 2006, pp. 912- 922.
10. M. Qin and R. Zimmermann, "VCA: An Energy Efficient Voting Based Clustering Algorithm for Sensor Networks," Journal of Universal Computer Science, Vol. 13, No. 1, 2007, pp. 87-109.

CHAPTER 3

SENSOR-NODE ARCHITECTURE

INTRODUCTION

A sensor node, also known as a mote, is a node in a sensor network that is capable of performing some processing, gathering sensory information and communicating with other connected nodes in the network. A mote is a node but a node is not always a mote.

A Wireless Sensor Network is one kind of wireless network that includes a large number of circulating, self-directed, minute, low powered devices named sensor nodes called motes. These networks certainly cover a huge number of spatially distributed, little, battery-operated, embedded devices that are networked to caringly collect, process, and transfer data to the operators, and it has controlled the capabilities of computing & processing. Nodes are tiny computers, which work jointly to form networks.

3.1 THE CONCEPT OF SENSOR NODE ARCHITECTURE

A sensor network is made up of the following parts, namely a set of sensor nodes which are distributed in a sensor field, a sink which communicates with the task manager via Internet interfacing with users. A set of sensor nodes is the basic component of a sensor network. Many researchers are currently engaged in developing pervasive sensor nodes due to the great promise and potential with applications shown by various wireless remote sensor networks.

Sensing units are usually made up of application specific sensors and ADCs (analog to digital converters), which digitalize the analog signals produced by the sensors when they sensed particular phenomenon. In some cases, an actuator is also needed.

Obviously sensors play a key role in a sensor network which are the very front end connecting our physical world to the computational world and the Internet. Although MEMS technology has been making steady progress in the past decades, there is still large space for the further development of smart front end sensors. Among them, various chemical and biochemical sensors remain one of the most challenging sensor groups to be explored and developed, e.g. sensors to detect toxic or explosive trace in public areas, sensors for diagnostic analysis and sensors used under extreme conditions. New sensing principle, new sensing material and new sensor design need to be invented and adopted.

The processing unit is usually associated with an embedded operating system, a microcontroller and a storage part. It manages data acquisition, analyzes the raw sensing data and formulates answers to specific user requests. It also controls the communication and performs a wide variety of application specified tasks. Energy and cost are two key constraints for processing components. Nodes may have different types of processors for certain specific tasks. For example, a video sensor node may need a more powerful processor to run than a common temperature sensor. A small embedded operation system such as Berkeley's TinyOS is another key issue for an embedded system. Besides the basic ability for process management and resource management, it may also possess the capability for software tailor and real time management, the ability to provide support for embedded middleware, network protocols and embedded database.

The transceiver connects the sensor node to the network. Usually each of the sensor nodes has the capability to transmit data to and receive data from another node and the sink. The latter may further communicate with the task manager via Internet (or Satellite) and information reaches the end user. A transceiver is the most power consuming component of the node. Thus the study of multi-hop communications and complex power saving modes of operation, e.g. having multiple different sleep states, is crucial in this content.

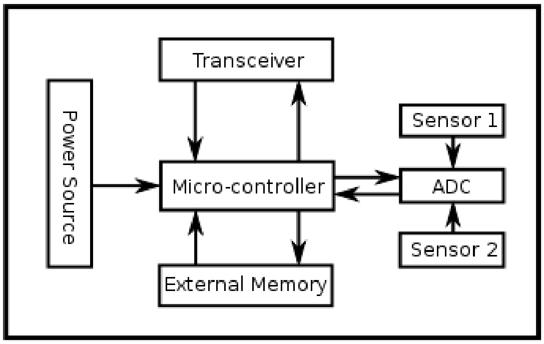
The power unit delivers power to all the working parts of the node. Because of the limited capacity of the power unit, e.g. the limited lifetime of a battery, the development of the power unit itself and the design of a power saving working mode of the sensor network remain some of the most important technical issues. For some applications, a solar battery may be used. Additionally, a sensor node may have application dependent functional subunits such as a location finder, a mobilizer, a power generator and other special-purpose sensors. The nature or number of such subunits may vary, depending on the application needs. It is a very interesting area to be continuously exploited.

3.1.1 History

Although wireless sensor nodes have existed for decades and used for applications as diverse as earthquake measurements to warfare, the modern development of small sensor nodes dates back to the 1998 Smartdust project and the NASA Sensor Web One of the objectives of the Smartdust project was to create autonomous sensing and communication within a cubic millimeter of space. Though this project ended early on, it led to many more research projects. They include major research centres in Berkeley NEST and CENS. The researchers involved in these projects coined the term *mote* to refer to a sensor node. The equivalent term in the NASA Sensor Webs Project for a physical sensor node is *pod*, although the sensor node in a Sensor Web can be another Sensor Web itself. Physical sensor nodes have been able to increase their capability in conjunction with Moore's Law. The chip footprint contains more complex and lower powered microcontrollers. Thus, for the same node footprint, more silicon capability can be packed into it. Nowadays, motes focus on providing the longest wireless range (dozens of km), the lowest energy consumption (a few uA) and the easiest development process for the user.

3.1.2 Components

The main components of a sensor node are a microcontroller, transceiver, external memory, power source and one or more sensors.



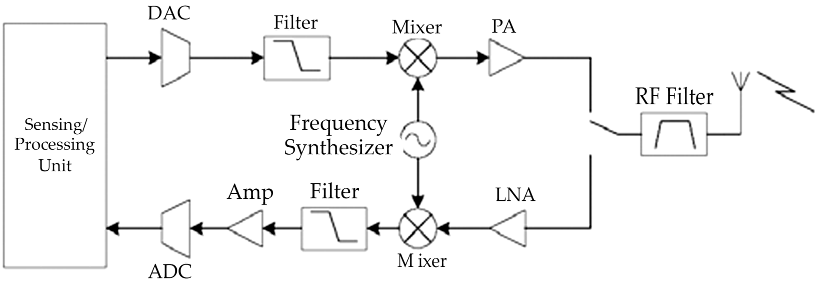
Controller

The controller performs tasks, processes data and controls the functionality of other components in the sensor node. While the most common controller is a microcontroller, other alternatives that can be used as a controller are: a general purpose desktop microprocessor, digital signal processors, FPGAs and ASICs. A microcontroller is often used in many embedded systems such as sensor nodes because of its low cost, flexibility to connect to other devices, ease of programming, and low power consumption. A general purpose microprocessor generally has a higher power consumption than a microcontroller, therefore it is often not considered a suitable choice for a sensor node. Digital Signal Processors may be chosen for broadband wireless communication applications, but in Wireless Sensor Networks the wireless communication is often modest: i.e., simpler, easier to process modulation and the signal processing tasks of actual sensing of data is less complicated. Therefore, the advantages of DSPs are not usually of much importance to wireless sensor nodes. FPGAs can be reprogrammed and reconfigured according to requirements, but this takes more time and energy than desired.

Transceiver

Sensor nodes often make use of ISM band, which gives free radio, spectrum allocation and global availability. The possible choices of wireless transmission media are radio frequency (RF), optical communication (laser) and infrared. Lasers require less energy, but need line-of-sight for communication and are sensitive to atmospheric conditions. Infrared, like lasers, needs no antenna but it is limited in its broadcasting capacity. Radio frequency-based communication is the most relevant that fits most of the WSN applications. WSNs tend to use license-free communication frequencies: 173, 433, 868, and 915 MHz; and 2.4 GHz. The functionality of both transmitter and receiver are combined into a single device known as a transceiver. Transceivers often lack unique identifiers. The operational states are transmit, receive, idle, and sleep. Current generation transceivers have built-in state machines that perform some operations automatically.

Most transceivers operating in idle mode have a power consumption almost equal to the power consumed in receive mode. Thus, it is better to completely shut down the transceiver rather than leave it in the idle mode when it is not transmitting or receiving. A significant amount of power is consumed when switching from sleep mode to transmit mode in order to transmit a packet.



External Memory

From an energy perspective, the most relevant kinds of memory are the on-chip memory of a microcontroller and Flash memory—off-chip RAM is rarely, if ever, used. Flash memories are used due to their cost and storage capacity. Memory requirements are very much application dependent. Two categories of memory based on the purpose of storage are: user memory used for storing application related or personal data, and program memory used for programming the device. Program memory also contains identification data of the device if present.

Power Source

A wireless sensor node is a popular solution when it is difficult or impossible to run a mains supply to the sensor node. However, since the wireless sensor node is often placed in a hard-to-reach location, changing the battery regularly can be costly and inconvenient. An important aspect in the development of a wireless sensor node is ensuring that there is always adequate energy available to power the system. The sensor node consumes power for sensing, communicating and data processing. More energy is required for data communication than any other process. The energy cost of transmitting 1 Kb a distance of 100 metres (330 ft) is approximately the same as that used for the execution of 3 million instructions by a 100 million instructions per second/W processor. Power is stored either in batteries or capacitors. Batteries, both rechargeable and non-rechargeable, are the main source of power supply for sensor nodes. They are also classified according to electrochemical material used for the electrodes such as NiCd (nickel-cadmium), NiZn (nickel-zinc), NiMH (nickel-metal hydride), and lithium-ion. Current sensors are able to renew their energy from solar sources, Radio Frequency(RF), temperature differences, or vibration. Two power saving policies used are Dynamic Power Management (DPM) and Dynamic Voltage Scaling (DVS). DPM conserves power by shutting down parts of the sensor node which are not currently used or active. A DVS scheme varies the power levels within the sensor node depending on the non-deterministic workload. By varying the voltage along with the frequency, it is possible to obtain quadratic reduction in power consumption.

Sensors

Sensors are used by wireless sensor nodes to capture data from their environment. They are hardware devices that produce a measurable response to a change in a physical condition like temperature or pressure. Sensors measure physical data of the parameter to be monitored and have specific characteristics such as accuracy, sensitivity etc. The continual analog signal produced by the sensors is digitized by an analog-to-digital converter and sent to controllers for further processing. Some sensors contain the necessary electronics to convert the raw signals into readings which can be retrieved via a digital link (e.g. I2C, SPI) and many convert to units such as °C. Most sensor nodes are small in size, consume little energy, operate in high volumetric densities, be autonomous and operate unattended, and be adaptive to the environment. As wireless sensor nodes are typically very small electronic devices, they can only be equipped with a limited power source of less than 0.5-2 ampere-hour and 1.2-3.7 volts.

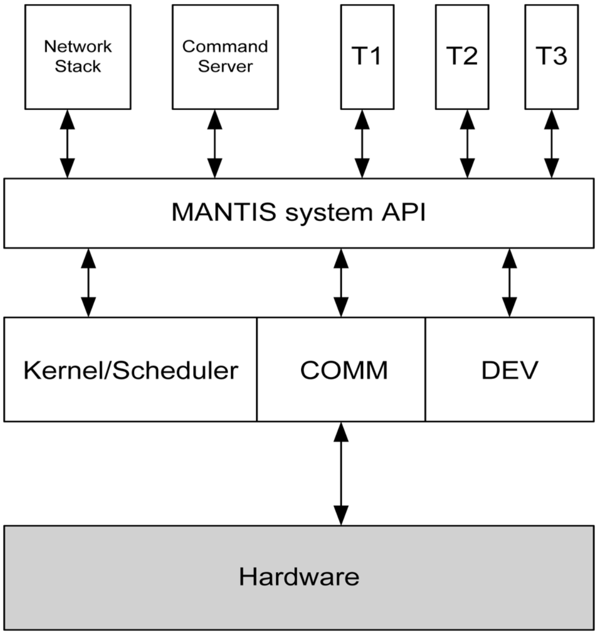
Sensors are classified into three categories: passive, omnidirectional sensors; passive, narrow-beam sensors; and active sensors. Passive sensors sense the data without actually manipulating the environment by active probing. They are self powered; that is, energy is needed only to amplify their analog signal. Active sensors actively probe the environment, for example, a sonar or radar sensor, and they require continuous energy from a power source. Narrow-beam sensors have a well-defined notion of direction of measurement, similar to a camera. Omnidirectional sensors have no notion of direction involved in their measurements.

Most theoretical work on WSNs assumes the use of passive, omnidirectional sensors. Each sensor node has a certain area of coverage for which it can reliably and accurately report the particular quantity that it is observing. Several sources of power consumption in sensors are: signal sampling and conversion of physical signals to electrical ones, signal conditioning, and analog-to-digital conversion. Spatial density of sensor nodes in the field may be as high as 20 nodes per cubic meter.

3.2 SENSOR NETWORK OPERATING SYSTEMS

Sensor networks have severe resource constraints in terms of processing power, memory size and energy, while operating in a communication-rich environment that interfaces both with the physical world and with other sensor network nodes. The operating system must efficiently manage the constrained resources while providing a programming interface, i.e. allow system developers to create resource-efficient software.

An operating system multiplexes hardware resources and provides an abstraction of the underlying hardware to make application programs simpler and more portable. Unlike general-purpose computers, which have settled for a number of semi-standardized hardware architectures, sensor network hardware is extremely diverse in terms of processor architectures, communication hardware and sensor devices. This makes operating system design for sensor networks a challenge.



In the sensor network research community, several operating systems have been developed, with each offering a different solution for the fundamental problems. TinyOS and Contiki are perhaps the two most well-known systems. TinyOS defines its own programming language called nesC, an extension to the C programming language, whereas Contiki uses standard C. Mantis, SOS and LiteOS are also widely cited sensor network operating systems.

Operating systems for sensor networks share some characteristics with real-time operating systems for embedded systems. Like sensor network nodes, embedded systems also often have severe resource constraints. But unlike embedded systems, sensor network nodes must interact both with the physical world and with each other: sensor networks are highly communication-intensive systems. This communication intensity adds additional challenges in terms of resource management and operating system structure.

Fundamental Problems

The fundamental problem that an operating system addresses is that of resource allocation. A sensor network node has a limited set of resources in terms of processor time, memory, storage, communication bandwidth and energy. The role of the operating system is to efficiently manage the available resources.

The operating system also provides a system programming interface to developers. This interface must be easy to use for system developers while providing efficiency. This results in additional constraints to the way the operating system can be designed.

Sensor Network Node Hardware

A sensor node (figure 1) consists of sensors and actuators, which interact with the physical world around the sensor node; a microcontroller, which interacts with the components and executes the software; a communication device, which typically is a radio; and a power source, which often is a battery but which also can be an energy-scavenging device such as a solar cell. Additionally, the sensor node may also contain secondary storage, such as on-board flash memory. Unlike general purpose computers, sensor network nodes do not have support for memory hierarchies, multiple protection domains or multi-level caches.

Typical hardware platforms for sensor network nodes have processing speeds of the order of a few megahertz, memory size of the order of hundreds of kilobytes and must run with less than 1 mW of power. Although Moore’s Law has somewhat relaxed the resource limitations over the past 10 years, it has primarily driven the hardware development in the direction of smaller, less expensive hardware platforms and lower power draws. With many sensor network applications requiring extremely low-cost devices, hardware development is unlikely to yield any extensive improvements in resources for the foreseeable future.

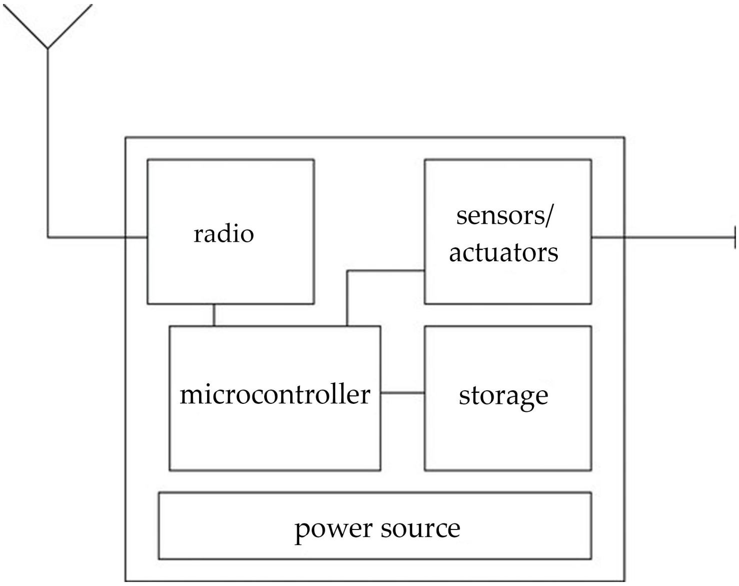


Figure 1. The hardware of a sensor node consists of a communication device, typically a radio, a microcontroller, a set of sensors and actuators and a storage device, typically a flash chip.

All hardware devices draw power when active, but their activity patterns are different. The microcontroller draws power when it is executing instructions and the sensors draw power when sensing physical phenomena. The communication device draws power both when it is transmitting and receiving data and when it is in idle mode, listening for messages from neighbors. The storage device typically draws power only when it is actively read from or written to and not when it is idle. To make the discussion concrete, table 1 contains the empirical power draws of the components of the Telos sensor node platform.

Table 1. The power consumption of hardware components of the Telos sensor network node.

component	**power draw (mW)**
microcontroller, sleeping	0.163
microcontroller, active	5.40
flash read	12.3
flash write	45.3
radio transmit	58.5
radio listen	65.4

Concurrency and Execution Models

An operating system must manage processor time so that each application gets its fair share. In a sensor network node, multiple activities may happen concurrently with respect to each other: sensor readings are collected from the on-board sensors; they are processed by the microcontroller and possibly stored in secondary storage and are transmitted over the communication device; communication from neighboring nodes is received and forwarded to other nodes, and timed events occur. The operating system must manage this concurrency in a way which is both resource efficient and easy to understand for the system developer.

The execution model of an operating system determines the way concurrent applications execute. Sensor network nodes operate in a highly concurrent environment and with severe resource constraints. Many sensor network operating systems therefore follow an event-driven execution model. In the event-driven execution model, the primary unit of execution is the event handler. An event handler is invoked in response to an external or internal event. Examples of external events are an incoming message from the communication device and a sensed phenomenon from a sensor. An example of an internal event is a timer that expires.

An alternative to the event-driven model is the multi-threaded model, which is also the most commonly used concurrency model in general purpose operating systems. Under the multi-threaded model, applications are defined as multiple threads that run concurrently. The threads block when waiting for external events. Operating systems such as Mantis make use of the multi-threaded model.

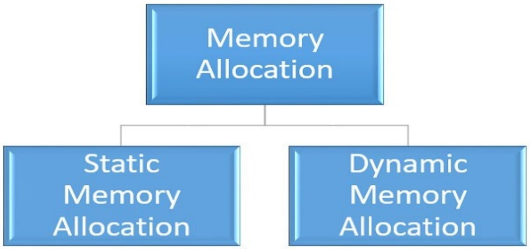
One of the primary benefits of the event-driven model over the multi-threaded model is memory efficiency. Since every event handler returns directly to the operating system, the system does not need to keep track of its state after the invocation is finished. By contrast, in the multi-threaded model, each thread must maintain memory for its stack. Much of this memory is unused, but must be kept free in case the thread needs to use it. Under the even-driven model, only one stack is needed, thereby reducing memory requirements.

When an event occurs in an event-driven system, the operating system finds the correct event handler to handle the event and invokes it. Event handlers have run-to-completion semantics: the event handler must quickly perform its action and return control back to the operating system. This approach works well for simple event handlers, which do their task quickly and return to the caller, but may be troublesome for more complex event handlers that need to wait for multiple events before continuing. Such event handlers must be split into multiple handlers since each event handler must run to completion. The developer must define a state machine that is driven by the event handlers. Research has shown that such state machines typically follow a set of simple patterns that correspond to how developers write programs under the multi-threaded paradigm.

The trade-off between memory efficiency and programmer complexity in the event-driven and the multi-threaded models has led to the development of several hybrid models. The protothread model provides a sequential flow of control, like the multi-threaded model, but without the overhead of multiple stacks. A protothread is a stackless type of thread that provides a conditional blocking wait primitive that allow programs to execute a blocking wait without a separate stack for each protothread. Another approach is to run multiple threads on top of an event-driven kernel, which allows the system developer to choose which execution model to use, depending on the needs of the application program. In the sensor network field, this hybrid threading model was first used in the Contiki operating system and was later improved upon in the TinyOS system.

Memory Allocation

On sensor nodes, the size of the memory is constrained on both physical and practical grounds. The size of the memory is determined by the number of transistors that hold the contents of the memory. This in turn affects both the power needed to maintain the memory contents and the manufacturing cost of the chip. Both limit the size of the memory used on sensor network nodes.



The memory is split into two parts: the static part, which contains the program code, and the dynamic part, which contains run-time variables, buffers, data and the stack. The static part is typically stored in read-only memory (ROM), whereas the dynamic part is held in random access memory (RAM). Because of the physical characteristics of existing microcontroller architectures, the RAM has a higher power draw than ROM and requires a larger physical chip area. For this reason, the RAM is typically smaller than the ROM. For example, the Telos platform has 48 kb of ROM and 10 kb of RAM. Moreover, unlike general purpose computer systems, sensor node microcontrollers do not have memory indirection or memory protection mechanisms.

The fundamental problem that memory allocation mechanisms must handle is memory fragmentation. Memory fragmentation is when unused memory is scattered across multiple memory regions that are not contiguous. When memory is fragmented, allocations may fail despite the total amount of unused memory being larger than the allocation.

To avoid fragmentation, operating systems for sensor nodes typically have avoided dynamic memory allocation. Instead, all memory has been statically allocated. For dynamic allocation needs, the system developer must pre-allocate static buffers, which may be used at runtime. This allows the system developer to understand the total memory requirements of the system beforehand and reduces the risk of the system running into a fatal fragmentation situation at runtime.

Energy

Sensor networks are typically battery operated. Since each battery has a fixed amount of energy, the power draw of each node effectively determines its lifetime. Energy is therefore a critical resource. The power draw of individual hardware components may differ by the order of magnitudes. Energy management is an essential service of the sensor node operating system.

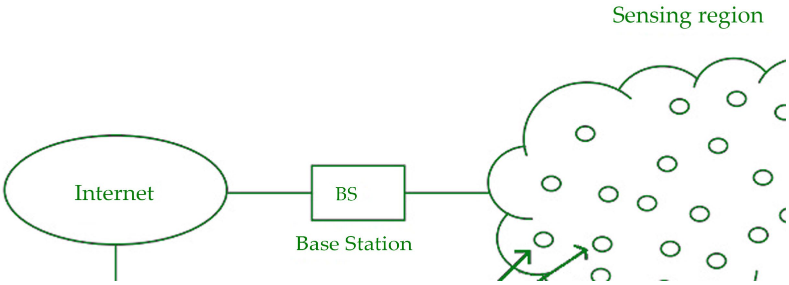
To reduce the power draw, the operating system must switch off unused components as often as it is possible to do so. The microprocessor is switched to sleep mode when no application is running. When an event occurs, such as a sensor reading taking place or a timer firing, the microcontroller is woken up. The operating system then invokes the appropriate application program. The communication component is difficult for the operating system to manage, as the component must be switched on for communication to occur. Communication energy management is therefore handled by a separate radio duty cycling mechanism.

Operating systems also may track energy consumption. For this, both hardware- and software-based approaches have been developed. Quanto uses a hardware-based energy meter coupled with a software-based power state and activity tracking system for TinyOS. The total time and energy measurements are dissected and attributed to hardware peripherals or logical activities. The Contiki and Pixie operating systems use an entirely software-based approach based on power state tracking, in which the system tracks the states of all components of the system. Their state determines the power draw of the device. The system collects this information into energy capsules that are attributed to activities such as individual packet transmissions or receptions. Based on the cumulative energy information in the energy capsules, a power profile can be determined.

The Eon system makes energy consumption a first-class abstraction and schedules application flows depending on the current energy profile. The Pixie operating system takes a different approach, in which the programmer articulates the energy requirements each application has and the operating system schedules tasks accordingly. By contrast with systems that leave energy management to the application layer, the integrated concurrency control and energy management architecture does automatic power management in the operating system, without the need for application involvement. This demonstrates that per-component energy management can be efficiently performed by the operating system without application-layer involvement.

Storage

In sensor networks, secondary storage takes the form of on-board flash ROM or secure digital cards. Storage systems for flash-based storage must deal with the physical storage semantics of flash memory. In flash memory, unlike RAM memory and magnetic disks, bits cannot be freely written: individual bits can only be flipped from 1 to 0. To reset bits from 0 to 1, an entire sector of bits must be erased. A sector typically contains many kilobytes of data. The storage system must be able to efficiently map data onto the sectors to make writing and erasing efficient. To make matters worse, individual sectors have a fixed number of erase cycles before they wear out. The storage system therefore must perform wear-levelling to spread the erasure load evenly across the memory to avoid wearing the memory out.



The traditional approach secondary storage overlays a file system over the storage. With the file system, named files can be created, written to, read from and deleted. The file system approach is general enough to underpin many mechanisms running on top of a file system and the approach has therefore been widely used.

Recognizing the often simple storage needs of early applications, the early work on file systems for sensor networks, such as Matchbox and efficient log-structured flash file system (ELF), used simplified file system models that only supported append operations and did not allow files to be overwritten. Evolving needs have led more recent systems, to provide a full file system interface that freely supports rewriting and deletion of files.

Other approaches to storage have also been proposed. Amnesiac storage is a technique in which sensor data stored in secondary storage are compressed over time. Recent data are compressed with low loss and, as data get older, the compression ratio is increased at the cost of loss of detail. The intuition behind the system is that, as data get older, the importance of detail decreases. Another model is the sensor-as-database model, which turns the on-board storage into a database, from which records can be retreived with SQL-like queries.

Communication Software Architectures

Application software running in sensor networks is often communication-bound. The sensor network operating system must make it easy for application programs to efficiently perform its communication tasks. Moreover, the operating system must make the underlying network protocols possible to implement efficiently. Each sensor network operating system provides a software framework in which network protocols can be implemented and efficiently executed. We call this the communication architecture of the operating system, and it performs memory allocation and management for message buffers, manages neighbour and address tables, and provides an interface for applications.

Traditional communication architectures follow a layered design in which different layers of the system solve an individual part of the communication problem. Early work in sensor networks challenged this traditional view, because of the novel resource constraints and application directions of sensor networks, and instead took the direction of rethinking layering and towards cross-layer optimization.

The TinyOS system uses a concept called Active Messages, where each message is tagged with an identifier that corresponds to an application at the receiver. When the operating system on the receiving node receives the message, it invokes the application that registered itself with the corresponding identifier. With an extension to the TinyOS system, Polastre *et al.* argued that the narrow waist of the sensor network stack should be placed at the link layer. The authors showed that abstracting the link layer allowed for generality in both neighbour management and neighbour sleep cycles. A later modular network layer added multi-hop functionality to this model, but more recent work has argued moving the narrow waist back to the network layer. The Contiki Rime communication architecture separates the protocol logic from construction and parsing of protocol headers, thereby making it possible to map the protocols across different underlying link layers and protocols, without sacrificing runtime performance.

Recently, the use of the Internet protocol (IP) architecture has become widespread in sensor networks. Contiki has long provided full IP-networking support through the uIP and uIPv6 stacks. Likewise, TinyOS provides IP-networking support through its Berkeley low-power Internet protocol (BLIP) stack. Because both systems are designed around the same underlying IP architecture, they share many of their design elements. This is a natural course of development as the community has progressed in addressing the fundamental problems in sensor network operating systems.

3.3 ENERGY OPTIONS FOR WIRELESS SENSOR NODES

Reduction in size and power consumption of consumer electronics has opened up many new opportunities for low power wireless sensor networks. Such networks have significant potential in a variety of applications, including monitoring of animal health and behaviour, structural monitoring for mining equipment and measuring water salinity levels of oceans and rivers. With these opportunities come a number of new challenges. Sensor nodes are usually battery powered, so as sensor networks increase in number and size, replacement of depleted batteries becomes time consuming and wasteful. Additionally, a battery that is large enough to last the life, say five years, of a sensor node would dominate the overall size of the node, and thus would not be very attractive or practical. Additionally, the battery chemistries often involve toxic heavy metals, and present disposal issues, regardless of rechargeable technology.

There is a clear need to explore novel alternatives to power sensor networks/nodes, as existing battery technology hinders the widespread deployment of these networks. By harvesting energy from their local environment, sensor networks can achieve much greater run-times, years not months, with potentially lower cost and weight.

Power for wireless sensor nodes can be split into two main technology categories: energy storage and energy harvesting. This paper reviews the state-of-the art technology in each of these fields, outlining different powering options for sensor nodes. These include energy storage utilizing batteries, capacitors, fuel cells, heat engines and betavoltaic systems and energy harvesting methodologies including photovoltaics, temperature gradients, fluid flow, pressure variations and motion harvesting. Energy storage is the basis of present technology and involves powering the sensor node from energy stored at the node; a key example of this is batteries. This energy may be stored in different forms ranging from electrical charge to hydrocarbon based fuels. By itself, energy storage cannot deliver energy indefinitely, as at some stage the energy will be depleted and need replenishing. The metric used for comparison of these devices is their average energy density, Joules per unit volume; typically this is J/cm^3.

Energy harvesting is a newer approach and relies on technology to gather energy from the surrounding environment using, for example, solar cells or fluid turbines. It involves converting the ambient energy inherent in the sensor node's environment into electrical energy. By doing so, a sensor node will have the opportunity to extend its life to a range determined by the failure of its own components rather than by its previously limited power supply. The metric used for comparison of energy harvesting devices differs from that used for energy storage as they don't have a fixed amount of energy intrinsic to their volume. Therefore, energy harvesting devices will be rated on their average power density or Watts per unit volume, W/cm^3, rather than their average energy density.

In general, energy harvesting will not directly power a sensor. This may be because the levels of power are too low, or it may be as a result of the power being in the wrong form. Typically, sensors and nodes require a voltage in the range 2 – 10 V and peak direct current of approximately 100 mA. Some energy harvesting techniques generate much higher voltages, produce AC power, or simply do not have sufficient power to run the node directly. The result of this is that electronics are required to condition the power for the device and, critically, secondary energy storage in the form of capacitors or rechargeable batteries will be required.

Many of the power options involve taking a technology which has been proven on large scale applications and scaling it down to dimensions suitable for the sensor node. This approach often runs into technical difficulties due to different effects which come into play at smaller scales. Some of these effects, thermal effects as a device's ratio of surface area to volume changes, viscosity issues involving fluid flow at smaller scale and problems related to increasing volume taken up by battery connectors, packaging and other essential hardware. However, through the persistent work of researchers, many technologies have overcome these obstacles and are nearing fruition.

It should be noted that in the context of this review, the term micro-scale is used to describe nodal elements with sizes of approximately 100 mm on a side, masses of less than 100 gm (not including batteries or associated transducers) and power requirements of less than 100 mW. To give examples of the energy requirements of sensor nodes, Table 2 shows a number of commercially available nodes and their various levels of power consumption.

Table 2. Consumption parameters for some example wireless sensor nodes

Node Name	Sleep Mode (µA)	Transmit Mode (mA)	Receive Mode (mA)	Duty Cycle (mA)	Operating Voltage (V)	Batteries Required	Battery Life (Days)
Fleck3	80	36.8	18.4	0.27	3.3	3	440
XBee^{TM}	10	45	50	0.51	2.8	2	230
MICAz^{TM}				0.70	2.7	2	170

The Fleck3 is a CSIRO product and a range of data was easily sourced, unlike the XBeeTM and MICAzTM which are both commercial products, for which full specifications were unavailable. The power consumption is very dependant on the various transmit/store duty cycle components and should not be interpreted as a measure of the efficiency of the device. Note that the nodes in Table 2 are often able to operate at lower power consumption. It is not the purpose of this review to compare individual nodes and manufacturers. Rather, the purpose of Table 2 is to indicate the relative amount of energy required for each node, for the following arbitrary duty cycle: In every 3 minute cycle, 1% (1.8 s) listening, 20 ms transmit time, and the remainder (178.18 s) sleeping.

The final two columns in Table 2 show the number of alkaline AA cells needed to power the node and the length of time they will last for the given duty cycle. Alkaline AA cells were chosen as alkaline chemistry is well established, has a reasonable shelf life, AA cells offer a good trade-off between capacity and size (2850 mAh from 8.3 cm^3) and that most readers would have some familiarity with them as they are commonly used in household devices. Note that the values quoted in Table 2 do not allow for powering sensors connected to the device. Typically, the power requirements of some sort of physical sensor to measure temperature or humidity, for example, need to be considered. The magnitude of this power could easily exceed the power requirement of the node itself, effectively halving the battery life.

3.3.1 Energy Storage

Batteries

The most common power sources for wireless sensor nodes are batteries. Batteries combine good energy density with a range of commercially available sizes while also supplying their energy at precisely the voltage levels required of modern electronics, eliminating the need for intermediate power conditioning electronics. A battery can store energy chemically and can release it as electricity through a chemical reaction which transfers electrons from its anode to its cathode. The power output of a particular battery is limited by a number of factors including: the relative potentials of the anode and cathode materials, and the surface area of the electrodes.

Batteries can be classed in two main categories, primary and secondary. Primary batteries are not easily recharged using electricity, while secondary batteries can reverse the chemical reaction through a recharging process whereby energy is delivered back into the battery and stored in the form of chemical bonds. When using primary batteries the lifetime of the sensor node is determined by the fixed amount of energy initially stored in the battery. The amount of energy stored depends on the energy density and volume of the battery. For sensor node applications, it is desirable to minimize the volume, and with improvements in battery energy density reaching a plateau, batteries are forcing a large trade-off between the node's lifetime and its volume.

The capacity of a battery is specified by the manufacturer and is achieved by the use of specific discharge rates. Each manufacturer can use their nominated methodology, of which there are many. As an example based on, the following is used for non-rechargeable batteries: A discharge rate of 25 mA is applied until the voltage reaches 0.8 V. The time in hours that is taken is then multiplied by the discharge rate (25 mA) to calculate a milliamp-hour (mAh) capacity. An alkaline battery that cannot be recharged has effectively reached the end of its life at 0.8 V. It still contains significant 'overhead' energy but this energy is unable to be used. When the battery is thrown out this energy is effectively wasted. This does not apply to rechargeable batteries, as they can be topped up hundreds of times.

For rechargeable batteries, a similar methodology is nominated by a manufacturer to achieve the rated capacity. For example bases their capacities on a 0.1 capacity charge followed by a 0.2 capacity discharge. So a rechargeable battery with a stated capacity of 1,000 mAh will only get that capacity if it is charged at a maximum of 100 mA (for 10 hrs) and then discharged at 200 mA (for 5 hrs). Other examples of discharge rates include a 1 hour rate and a 20 hour rate. Thus a 1,000 mAh battery will achieve this capacity using the 1 hour rate if discharged at 1,000 mA. The 20 hour rate is typically used on sealed lead acid batteries as they do not perform well at the 1 hour rate. Importantly, a battery will only achieve the nominated capacity if discharged at the nominated rate. A higher or lower discharge rate will result in a different capacity due to internal energy changes. Typically if a lower discharge rate is used then the battery will supply a slightly higher capacity. This is important for wireless nodes as they typically consume much less than the nominated discharge rate, and thus the battery should last slightly longer than predicted from the capacity.

Like primary batteries there are different types of secondary batteries whose characteristics are determined by their internal chemistries. Conventional chemistries such as Nickel-Zinc (NiZn), Nickel Metal Hydride (NiMH) and Nickel-Cadmium (NiCd), offer high energy densities and good discharge rates, but with the disadvantages of short cycle life and adverse "memory" effects. Lithium-ion batteries overcome these drawbacks, with a higher energy density and discharge rate, higher cell voltage, longer cycle life and elimination of "memory" effects. However their major disadvantage is the particular care required when recharging to avoid overheating and permanent damage. Figure 2 shows the relative strengths of the different battery chemistries in terms of their energy and power densities.

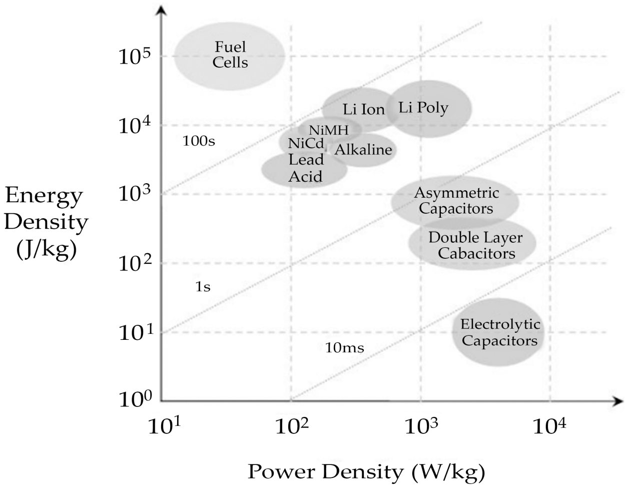


Figure 2. Ragone chart for capacitors, supercapacitors, batteries and fuel cells.

Some battery chemistries have problems with shelf life. Standard alkaline batteries have shelf lives of around seven years; while newer lithium based systems (both primary and secondary) have even longer lives. Other secondary (rechargeable) chemistries like Nickel Metal Hydride (NiMH) lose 1 − 2% of their capacity per day of storage.

Secondary batteries provide the option of extending the sensor node's lifetime, relative to that of a primary battery, through their recharging ability. However, this means they need to run in conjunction with another device capable of supplying power. This arrangement is usually desirable as quite often the device supplying the power does so intermittently. A battery can store these bursts of energy and provide the electronics with a stable constant energy interface. A robust system will require electronics to control the charging and discharging of the battery in a way that maximizes its life as incorrect charging profiles diminish the battery's usable life.

Two promising new fields of research in battery technology are micro-batteries and flexible batteries. Micro-batteries seek not only to reduce the size of the actual battery but also to improve integration with the electronics they are powering. The goal of micro-batteries is therefore to produce a battery on a chip. The main challenge is overcoming small power outputs due to the surface area limitations of micro-batteries, however work into three-dimensional surfaces seem promising. The second field involves a new breed of lightweight flexible batteries which can be moulded to any shape allowing them to serve a double purpose of acting as structural material, thus reducing the total volume of the sensor node.

Capacitors

Capacitors store energy in the electric field between a pair of oppositely charged conductors. They have significantly higher power density than batteries, as they are able to charge and discharge over much shorter periods of time. However, their energy density is two to three orders of magnitude lower.

This makes capacitors ideal for providing short bursts of high power with low duty cycles giving the capacitor time to recharge before the next burst of power is needed. This effect may mean a combination of capacitor and battery could solve the power requirement across a normal nodal duty cycle. A battery can be used to provide the low power requirements on sleep and receive mode, while a capacitor can provide the high power required for RF transmission on short duty cycles.

Continued research into capacitors strives to increase their energy density, with a new breed of supercapacitors. Figure 3 shows a charged supercapacitor. The critical difference between a supercapacitor and a standard capacitor is in the surface area supplied by the electrode and the thinness of the double layer formed at the electrode-electrolyte interface. In a standard capacitor the area is simply the surface area of a nominally flat plate. However, the use of porous materials such as carbon effectively increases the surface area of each electrode enormously. This allows capacitors with values of the order of 2000 F in packages approximating standard battery sizes.

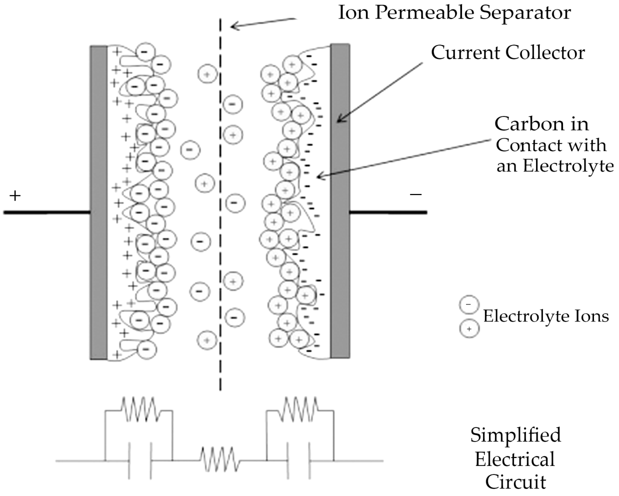

Ion Permeable Separator
Current Collector
Carbon in Contact with an Electrolyte
+
–
Electrolyte Ions
Simplified Electrical Circuit

Figure 3. Representation of a charged electrochemical double layer capacitor.

The simplified circuit shown in Figure 3 hints at a further improvement: Capacitors in series add such that the total capacitance of a cell is given by:

$$\frac{1}{C_{CELL}} = \frac{1}{C_1} + \frac{1}{C_2} \qquad (1)$$

Thus, for a supercapacitor both C_1 and C_2 are large and this leads to a C_{CELL} approximately half the size of C_1 or C_2. This has lead to the development of so called asymmetric capacitors, as seen in Figure 4. An asymmetric supercapacitor typically consists of a battery type electrode (usually a faradaic or intercalating metal oxide) and an electrochemical capacitor type electrode (high surface area carbon). In such an arrangement, the carbon electrode has a much greater capacity than the battery electrode. Thus C_{CELL} approaches the capacitance of the carbon electrode alone, resulting in a much larger energy storage capability of a comparable symmetric carbon based supercapacitor. This has lead to development of cells with capacitance values in excess of 8,000 F.

The increase in capacitance values has led to energy storage capabilities approaching that of some battery chemistries, such as lead-acid storage cells, and power storage capabilities and order of magnitude greater. Critically, the efficiency of capacitors exceeds 90% while batteries have typical values of 60-70%.

Although the specific application here is for hybrid cars, the technology should be applicable to sensor nodes. Figure 2 shows a comparison of the energy and power densities of the energy storage devices just discussed: capacitor, supercapacitor and battery. Some supercapacitors are capable of more than 500,000 charge cycles before noticeable deterioration (compared with about 1,000 for rechargeable batteries). This factor, along with short charging times and high power densities, make supercapacitors attractive as secondary power sources in place of rechargeable batteries in some wireless sensor network applications.

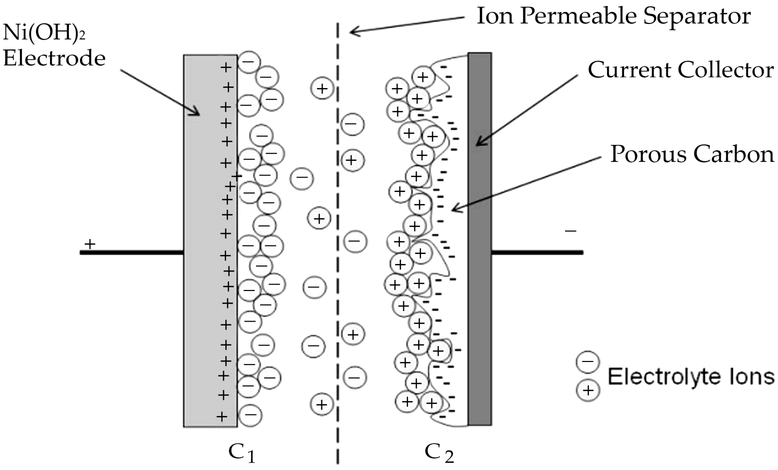


Figure 4. Schematic of an Ni(OH)$_2$/NiOOH–porous carbon asymmetric supercapacitor.

3.3.2 Micro-Fuel Cells

Like batteries, fuel cells convert stored chemical energy into electricity. Generally, liquid fuels have much higher energy density than battery chemistries. In the fuel cell, such as the one shown in Figure 5, a catalyst promotes the separation of the electrons from the protons of hydrogen atoms drawn from the fuel. The electrons are then available for use by an external circuit, while the protons diffuse through an electrolyte to recombine with the electrons and oxygen on the other side producing water molecules. This technology was pioneered for the NASA space program and has been used on large scales for decades but recent work has focused on reducing their size to replace consumer batteries.

As with batteries, the major performance restriction of micro-scale fuels cells results from the small electrode surface area. An opportunity may exist to combine the work of Hart et al. involving three dimensional surfaces in battery electrodes, with the noted shortcomings of fuel cell electrodes. Another hindrance is the plumbing for the fuel reservoir which at micro-scales is seen as a harder task than micro-fabricating the electrodes. The main issue here is due to flow considerations and ensuring that the fuel flows throughout the cell particularly to the finer tubing at the extremities.

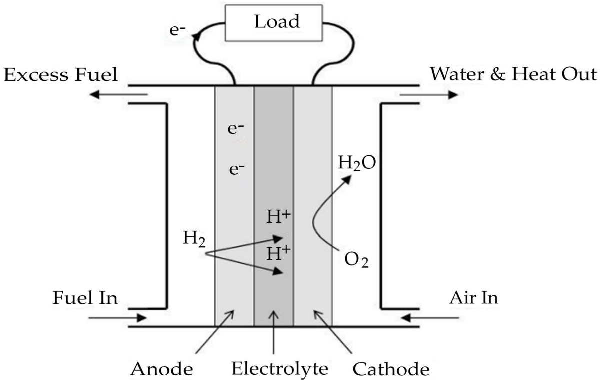


Figure 5. Example of a polymer electrolyte membrane (PEM) fuel cell.

Matsushita Battery has developed a direct methanol fuel cell (DMFC) incorporated with a lithium ion battery. This system is approximately 400 cm^3, with peak output of 20 W and an average of 13 W. This corresponds to an average power density of 0.03 W/cm^3. Angstrom Power has completed a six month test program using a hydrogen fuel cell. The fuel is supplied as hydrogen absorbed in a metal hydride. The volume of the fuel storage is around 6 cm^3, and the fuel cell itself can be made in many forms. The two presently available are a cylindrical, 1 W unit with a volume of 10 cm^3, and a rectangular 0.38 W unit with a volume of 2.5 cm^3. The average power densities for these, including the fuel storage, are 0.06 W/cm^3 and 0.04 W/cm^3, respectively.

Radioactive Power Sources

The use of radioactive materials as a power source is attractive due to their extremely high average energy densities, approximately 10^5 kJ/cm^3. Like many other power sources it has been used in the large scale for decades but has not yet fully transferred down to a scale useful for sensor nodes. The main technical reason for this is the lack of a high conversion efficiency mechanism at the microscale.

Early research into small scale radioactive energy conversion focussed on thermal heating using the kinetic energy of emitted particles. The heat could be converted into electricity using thermoelectric or thermionic techniques which require high temperatures (300 – 900 K) for efficient operation. This scheme works well for operations requiring power in the Watt to kilowatt range but doesn't scale down for micro-power applications since with reducing size, the surface-to-volume ratio increases, leading to high heat leakage to the surroundings, i.e. thermal heat management at the micro-scale is a tough engineering challenge.

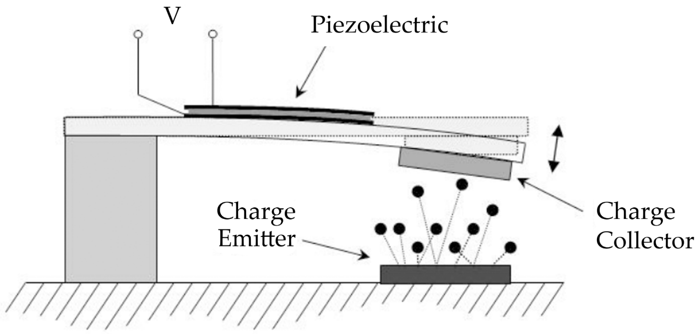


Figure 6. Radioisotope energy harvester.

To date the most promising work for applications in powering wireless sensor nodes is by Lal et al. where they have used a radioactive isotope to actuate a conductive cantilever. As shown in Figure 6 the emitted electrons collect on the cantilever which causes an electrostatic attraction forcing the cantilever to bend towards the source. When contact is made the charge differential is dissipated and the cantilever oscillates about its equilibrium position. A piezoelectric plate will convert the mechanical energy of the oscillation into electrical energy. They have demonstrated a power conversion efficiency of 2 – 3% using this radioactive-to-mechanical-to electrical conversion cycle with power outputs in the tens of microwatts, which could power low-power electronics or trickle charge a battery or capacitor.

The weakness of useable radioactive sources is their low power density. Typically, the longer the half-life of an element, the lower the power density. As such they do not by themselves offer a standalone solution to the powering of sensor nodes. However, they are an extremely consistent power source with long lifetimes governed by the half-life of the source which in some cases can be centuries. Because of this they are often put into the energy harvesting category, but strictly speaking they are in fact an energy storage source. Possible uses include extending the life of batteries, charging capacitors, or providing power to applications which need very low power. Due to safety concerns the use of radioactive material is a highly political and controversial topic. As Table 3 shows, although some groups of betavoltaics offer good power density, they also require extensive shielding.

Table 3. Decay sources

Element	Power Capacity (W/mole)	Power Capacity (W/gm)	Half Life (yrs)	Lead Shielding (mm/W)
Caesium-137	20	0.15	30.2	80
Cerium-144	3600	25	0.78	150
Nickel-63	0.42	0.0067	100	<5
Promethium-147	50	0.34	2.6	<1
Strontium-90	80	0.88	28	70
Thulium-170	2200	12.9	0.35	30
Thallium-204	160	0.78	3.8	30

3.4 PHYSICAL LAYER AND TRANSCEIVER DESIGN CONSIDERATIONS IN WSNS

The physical layer is mostly concerned with modulation and demodulation of digital data; this task is carried out by so-called transceivers. In sensor networks, the challenge is to find modulation schemes and transceiver architectures that are simple, low cost, but still robust enough to provide the desired service.

Some of the most crucial points influencing PHY design in wireless sensor networks are:

- Low power consumption.
- As one consequence: small transmit power and thus a small transmission range.
- As a further consequence: low duty cycle. Most hardware should be switched off or operated in a low-power standby mode most of the time.
- Comparably low data rates, on the order of tens to hundreds kilobits per second, required.
- Low implementation complexity and costs.
- Low degree of mobility.
- A small form factor for the overall node.

In general, in sensor networks, the challenge is to find modulation schemes and transceiver architectures that are simple, low-cost but still robust enough to provide the desired service.

3.4.1 Energy Usage Profile

The choice of a small transmit power leads to an energy consumption profile different from other wireless devices like cell phones.

First, the radiated energy is small, typically on the order of 0 dBm (corresponding to 1 mW). On the other hand, the overall transceiver (RF front end and baseband part) consumes much more energy than is actually radiated; Wang et al. [855] estimate that a transceiver working at frequencies beyond 1 GHz takes 10 to 100 mW of power to radiate 1 mW. Similar numbers are given for 2.4-GHz CMOS transceivers: For a radiated power of 0 dBm, the transmitter uses actually 32 mW, whereas the receiver uses even more, 38 mW. For the Mica motes, 21 mW are consumed in transmit mode and 15 mW in receive mode. These numbers coincide well with the observation that many practical transmitter designs have efficiencies below 10 % at low radiated power.

A second key observation is that for small transmit powers the transmit and receive modes consume more or less the same power; it is even possible that reception requires more power than transmission; depending on the transceiver architecture, the idle mode's power consumption can be less or in the same range as the receive power. To reduce average power consumption in a low-traffic wireless sensor network, keeping the transceiver in idle mode all the time would consume significant amounts of energy. Therefore, it is important to put the transceiver into sleep state instead of just idling. It is also important to explicitly include the received power into energy dissipation models, since the traditional assumption that receive energy is negligible is no longer true.

However, there is the problem of the startup energy/startup time, which a transceiver has to spend upon waking up from sleep mode, for example, to ramp up phase-locked loops or voltage controlled oscillators. During this startup time, no transmission or reception of data is possible. For example, the µAMPS-1 transceiver needs a startup time of 466 µs and a power dissipation of 58 mW. Therefore, going into sleep mode is unfavorable when the next wakeup comes fast. It depends on the traffic patterns and the behavior of the MAC protocol to schedule the transceiver operational state properly. If possible, not only a single but multiple packets should be sent during a wakeup period, to distribute the startup costs over more packets. Clearly, one can attack this problem also by devising transmitter architectures with faster startup times.

A third key observation is the relative costs of communications versus computation in a sensor node. Clearly, a comparison of these costs depends for the communication part on the BER requirements, range, transceiver type, and so forth, and for the computation part on the processor type, the instruction mix, and so on.

3.4.2 Choice of Modulation Scheme

A crucial point is the choice of modulation scheme. Several factors have to be balanced here: the required and desirable data rate and symbol rate, the implementation complexity, the relationship between radiated power and target BER, and the expected channel characteristics.

To maximize the time a transceiver can spend in sleep mode, the transmit times should be minimized. The higher the data rate offered by a transceiver/modulation, the smaller the time needed to transmit a given amount of data and, consequently, the smaller the energy consumption.

A second important observation is that the power consumption of a modulation scheme depends much more on the symbol rate than on the data rate. For example, power consumption measurements of an IEEE 802.11b Wireless Local Area Network (WLAN) card showed that the power consumption depends on the modulation scheme, with the faster Complementary Code Keying (CCK) modes consuming more energy than DBPSK and DQPSK; however, the relative differences are below 10 % and all these schemes have the same symbol rate. It has also been found that for the µAMPS-1 nodes the power consumption is insensitive to the data rate.

The desire for “high” data rates at “low” symbol rates calls for m-ary modulation schemes. However, there are trade-offs:

- m-ary modulation requires more complex digital and analog circuitry than 2-ary modulation, for example, to parallelize user bits into m-ary symbols.
- Many m-ary modulation schemes require for increasing m an increased E_b/N_0 ratio and consequently an increased radiated power to achieve the same target BER; others become less and less bandwidth efficient. This is exemplarily shown for coherently detected m-ary FSK and PSK in Table 4, where for different values of m, the achieved bandwidth efficiencies and the E_b/N_0 required to achieve a target BER of 10^{-6} are displayed. However, in wireless sensor network applications with only low to moderate bandwidth requirements, a loss in bandwidth efficiency can be more tolerable than an increased radiated power to compensate E_b/N_0 losses.

- It is expected that in many wireless sensor network applications most packets will be short, on the order of tens to hundreds of bits. For such packets, the startup time easily dominates overall energy consumption, rendering any efforts in reducing the transmission time by choosing m-ary modulation schemes irrelevant.

Table 4. Bandwidth efficiency ηBW and E_b/N_0[dB] required at the receiver to reach a BER of 10^{-6} over an AWGN channel for m-ary orthogonal FSK and PSK

m	2	4	8	16	32	64
m-ary PSK:η_{BW}	0.5	1.0	1.5	2.0	2.5	3.0
m-ary PSK:E_b/N_0	10.5	10.5	14.0	18.5	23.4	28.5
m-ary FSK:η_{BW}	0.40	0.57	0.55	0.42	0.29	0.18
m-ary FSK:E_b/N_0	13.5	10.8	9.3	8.2	7.5	6.9

Let us explore the involved trade-offs a bit further with the help of an example.

Example 4.1 (Energy efficiency of m-ary modulation schemes) Our goal is to transmit data over a distance of d = 10 m at a target BER of 10^{-6} over an AWGN channel having a path-loss exponent of $\gamma = 3.5$. We compare two families of modulations: coherently detected m-ary PSK and coherently detected orthogonal m-ary orthogonal FSK. For these two families we display in Table 4, the bandwidth efficiencies η_{BW} and the E_b/N_0 in dB required at the receiver to reach a BER of 10^{-6} over an AWGN channel.

The relationship between E_b/N_0 and the received power at a distance d is given as:

$$\frac{E_b}{N_0} = \mathrm{SNR} \cdot \frac{1}{R} = \frac{P_{\mathrm{rcvd}}(d)}{N_0} \cdot \frac{1}{R}$$
$$= \frac{1}{N_0 \cdot R} \cdot \frac{P_{\mathrm{tx}} \cdot G_t \cdot G_r \cdot \lambda^2}{(4\pi)^2 \cdot d_0^{\gamma} \cdot L} \cdot \left(\frac{d_0}{d}\right)^{\gamma},$$

which can be easily solved for Ptx given a required E_b/N_0 value and data rate R. We denote the solution as $P_{\mathrm{tx}}\left(\frac{E_b}{N_0}, R\right)$. One example: From Table 4 we obtain that 16-PSK requires an E_b/N_0 of 18.5 dB to reach the target BER. When fixing the parameters $G_t = G_r = L = 1$, $\lambda = 12.5$ cm (according to a 2.4 GHz transceiver), reference distance $d_0 = 1$ m, distance d = 10 m, a data rate of R = 1 Mbps, and a noise level of N0 = −180 dB this corresponds to P_{tx} (18.5 dB, R) ≈ 2.26 mW.

In this model, it is assumed that during the startup time mainly a frequency synthesizer is active, consuming energy PFS, while during the actual waveform transmission power is consumed by the frequency synthesizer, the modulator (using P_{MOD}), and the radiated energy $P_{tx}(\cdot,\cdot)$. The power amplifier is not explicitly considered. We assume PFS = 10 mW, P_{MOD} = 2 mW and a symbol rate of B = 1 M symbols/sec. The duration of the startup time is T_{start}. For the case of binary modulation, we assume the following energy model:

$$E_{\text{binary}}\left(\frac{E_b}{N_0}, B\right) = P_{\text{FS}} \cdot T_{\text{start}} + \left(P_{\text{MOD}} + P_{\text{FS}} + P_{\text{tx}}\left(\frac{E_b}{N_0}, B\right)\right) \cdot \frac{n}{B},$$

where n is the number of data bits to transmit in a packet. For the case of m-ary modulation, it is assumed that the power consumption of the modulator and the frequency synthesizer are increased by some factors $\alpha \geq 1$, $\beta \geq 1$, such that the overall energy expenditure is:

$$E_{m\text{-ary}}\left(\frac{E_b}{N_0}, B \cdot \log_2 m\right) = \beta \cdot P_{\text{FS}} \cdot T_{\text{start}} + \left(\alpha \cdot P_{\text{MOD}} + \beta \cdot P_{\text{FS}} + P_{\text{tx}}\left(\frac{E_b}{N_0}, B \cdot \log_2 m\right)\right) \cdot \frac{n}{B \cdot \log_2(m)}.$$

Accepting the value β = 1.75 for both PSK and FSK modulation, one can evaluate the ratio $\frac{E_{m\text{-ary}}(\cdot,\cdot)}{E_{\text{binary}}(\cdot,\cdot)}$ to measure the energy advantage or disadvantage of mary modulation over binary modulation. As an example, we show this ratio in Figure 7 for varying m ∈ {4, 8, 16, 32, 64}, with α = 2.0, a startup time of 466 μs, and two different packet sizes, 100 bits and 2000 bits. The two upper curves correspond to a packet size of 100 bits; the two lower curves correspond to the packet size of 2000 bits.

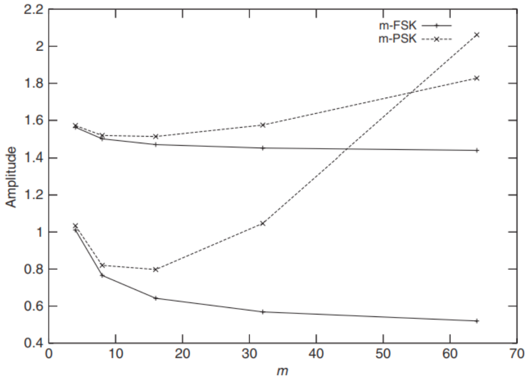


Figure 7. Comparison of the energy consumption of m-ary FSK/PSK to binary FSK/PSK for α = 2.0 and startup time of 466 μs.

Other results obtained with a shorter startup time of 100 μs or α = 3.0 look very similar. One can see that for large packet sizes m-ary FSK modulation is favorable, since the actual packet transmission times are shortened and furthermore the required E_b/N_0 decreases for increasing m, at the expense of a reduced bandwidth efficiency, which translates into a wider required spectrum (the FSK scheme is orthogonal FSK). For m-ary PSK, only certain values of m give an energy advantage; for larger m the increased E_b/N_0 requirements outweigh the gains due to reduced transmit times. For small packet sizes, the binary modulation schemes are more energy efficient for both PSK and FSK, because the energy costs are dominated by the startup time. If one reduces β to β = 1 (assuming no extra energy consumption of the frequency synthesizer due to m-ary modulation), then m-ary modulation would, for all parameters under consideration, be truly better than binary modulation. For shorter startup times also the packet lengths required to make m-ary modulation pay out are smaller.

Clearly, this example provides only a single point in the whole design space. The bottom line here is that the choice of modulation scheme depends on several interacting aspects, including technological factors (in the example: α, β), packet size, target error rate, and channel error model (A similar example is carried out for the case of Rayleigh fading). The optimal decision would have to properly balance the modulation scheme and other measures to increase transmission robustness, since these also have energy costs:

- With retransmissions, entire packets have to be transmitted again.
- With FEC coding, more bits have to be sent and there is additional energy consumption for coding and decoding. While coding energy can be neglected, and the receiver needs significant energy for the decoding process. This is especially cumbersome if the receiver is a power-constrained node.
- The cost of increasing the radiated power depends on the efficiency of the power amplifier, but the radiated power is often small compared to the overall power dissipated by the transceiver, and additionally this drives the PA into a more efficient regime.

A similar analysis as in our example has been carried out for m-ary QAM. Specifically, the energy-per-bit consumption (defined as the overall energy consumption for transmitting a packet of n bits divided by n) of different m-ary QAM modulation schemes has been investigated for different packet sizes, taking startup energy and the energy costs of power amplifiers as well as PHY and MAC packet overheads explicitly into account. For the particular setup used in this investigation, 16-QAM seems to be the optimum modulation schemes for all different sizes of the user data.

3.4.3 Dynamic Modulation Scaling

Even if it is possible to determine the optimal scheme for a given combination of BER target, range, packet sizes and so forth, such an optimum is only valid for short time; as soon as one of the constraints changes, the optimum can change, too. In addition, other constraints like delay or the desire to achieve high throughput can dictate to choose higher modulation schemes.

Therefore, it is interesting to consider methods to adapt the modulation scheme to the current situation. Such an approach, called dynamic modulation scaling. In particular, for the case of m-ary QAM and a target BER of 10^{-5}, a model has been developed that uses the symbol rate B and the number of levels per symbol m as parameters. This model expresses the energy required per bit and also the achieved delay per bit (the inverse of the data rate), taking into account that higher modulation levels need higher radiated energy. Extra startup costs are not considered. Clearly, the bit delay decreases for increasing B and m. The energy per bit depends much more on m than on B. In fact, for the particular parameters chosen, it is shown that both energy per bit and delay per bit are minimized for the maximum symbol rate. With modulation scaling, a packet is equipped with a delay constraint, from which directly a minimal required data rate can be derived. Since the symbol rate is kept fixed, the approach is to choose the smallest m that satisfies the required data rate and which thus minimizes the required energy per bit. Such delay constraints can be assigned either explicitly or implicitly. One approach explored in the paper is to make the delay constraint depend on the packet backlog (number of queued packets) in a sensor node: When there are no packets present, a small value for m can be used, having low energy consumption. As backlog increases, m is increased as well to reduce the backlog quickly and switch back to lower values of m.

3.4.4 Antenna Considerations

The desired small form factor of the overall sensor nodes restricts the size and the number of antennas. If the antenna is much smaller than the carrier's wavelength, it is hard to achieve good antenna efficiency, that is, with ill-sized antennas one must spend more transmit energy to obtain the same radiated energy.

Secondly, with small sensor node cases, it will be hard to place two antennas with suitable distance to achieve receive diversity. The antennas should be spaced apart at least 40–50 % of the wavelength used to achieve good effects from diversity. For 2.4 GHz, this corresponds to a spacing of between 5 and 6 cm between the antennas, which is hard to achieve with smaller cases.

In addition, radio waves emitted from an antenna close to the ground – typical in some applications – are faced with higher path-loss coefficients than the common value $\alpha = 2$ for free-space communication. Typical attenuation values in such environments, which are also normally characterized by obstacles (buildings, walls, and so forth), are about $\alpha = 4$. Moreover, depending on the application, antennas must not protrude from the casing of a node, to avoid possible damage to it. These restrictions, in general, limit the achievable quality and characteristics of an antenna for wireless sensor nodes.

Nodes randomly scattered on the ground, for example, deployed from an aircraft, will land in random orientations, with the antennas facing the ground or being otherwise obstructed. This can lead to nonisotropic propagation of the radio wave, with considerable differences in the strength of the emitted signal in different directions. This effect can also be caused by the design of an antenna, which often results in considerable differences in the spatial propagation characteristics (so-called lobes of an antenna).

SUMMARY

- A sensor node, also known as a mote, is a node in a sensor network that is capable of performing some processing, gathering sensory information and communicating with other connected nodes in the network. A mote is a node but a node is not always a mote.
- A sensor network is made up of the following parts, namely a set of sensor nodes which are distributed in a sensor field, a sink which communicates with the task manager via Internet interfacing with users. A set of sensor nodes is the basic component of a sensor network.
- The power unit delivers power to all the working parts of the node. Because of the limited capacity of the power unit, e.g. the limited lifetime of a battery, the development of the power unit itself and the design of a power saving working mode of the sensor network remain some of the most important technical issues.
- The controller performs tasks, processes data and controls the functionality of other components in the sensor node. While the most common controller is a microcontroller, other alternatives that can be used as a controller are: a general purpose desktop microprocessor, digital signal processors, FPGAs and ASICs.
- Sensor nodes often make use of ISM band, which gives free radio, spectrum allocation and global availability. The possible choices of wireless transmission media are radio frequency (RF), optical communication (laser) and infrared.
- A wireless sensor node is a popular solution when it is difficult or impossible to run a mains supply to the sensor node.
- Sensors are used by wireless sensor nodes to capture data from their environment. They are hardware devices that produce a measurable response to a change in a physical condition like temperature or pressure.
- Sensor networks have severe resource constraints in terms of processing power, memory size and energy, while operating in a communication-rich environment that interfaces both with the physical world and with other sensor network nodes.
- An operating system multiplexes hardware resources and provides an abstraction of the underlying hardware to make application programs simpler and more portable. Unlike general-purpose computers, which have settled for a number of semi-standardized

hardware architectures, sensor network hardware is extremely diverse in terms of processor architectures, communication hardware and sensor devices. This makes operating system design for sensor networks a challenge.

- The execution model of an operating system determines the way concurrent applications execute. Sensor network nodes operate in a highly concurrent environment and with severe resource constraints.

REFERENCES

1. Arici, T. and Altunbasak, Y.: Adaptive Sensing for Environment Monitoring using Wireless Sensor Networks. Proc. IEEE Wireless Communications and Networking Conference (WCNC), Atlanta, GA, (2004)

2. Davidson, J.; Knight, C.; Behrens, S. Scoping study - energy harvesting for wireless sensor networks. CSIRO internal report ET/IR 1017, February 2008.

3. Ferrari, M.; Ferrari, V.; Guizetti, M.; Marioli, D.; Taroni, A. Characterization of Thermoelectric Modules for Powering Autonomous Sensors. In Instrument and Measurement Technology Conference, Warsaw Poland, 2007.

4. Hart, R.W.; White, H.S.; Dunn, B.; Rolison, D.R. 3-D microbatteries. Electrochem. Commun. 2003, 5, 120–123.

5. Pandolfo, A.G.; Duffy, N.W. High energy Asymmetric Nickel-Carbon supercapacitors. In Proceedings of the Land Warfare Conference; Defence Science and Technology Organisation: Salisbury, 2006.

6. Paradiso, J.A.; Starner, T. Energy scavenging for mobile and wireless electronics. IEEE Pervasive Comput. 2005, 4, 18–27.

7. Roundy, S.; Wright, P.K.; Rabaey, J. A study of low level vibrations as a power source for wireless sensor nodes. Comput. Commun. 2003, 26, 1131–1144.

8. Roundy, S.; Wright, P.K.; Rabaey, J.M. Energy Scavenging for Wireless Sensor Networks: With Special Focus on Vibrations; Kluwer Academic Publishers: Norwell, MA, USA, 2004.

9. Schwiebert, L., Gupta, S.K.S., and Weinmann. J.: Research challenges in wireless networks of biomedical sensors. In Mobile Computing and Networking, (2001) 151-165

10. Taylor, G.; Burns, J.; Kammann, S.; Powers, W.; Welsh, T. The energy harvesting eel: a small subsurface ocean/river power generator. IEEE J. Oceanic Eng. 2001, 26, 539–547.

11. Venkatasubramanian, R.; Siivola, E.; Colpitts, T.; O'Quinn, B. Thin-film thermoelectric devices with high room-temperature figures of merit. Nature 2001, 597-602.

12. Wark, T.; Hu, W.; Corke, P.; Hodge, J.; Keto, A.; Mackey, B.; Foley, G.; Sikka, P.; Brunig, M. Springbrook: Challenges in Developing a Long-Term, Rainforest Wireless Sensor Network. In 4th International Conference on Intelligent Sensors, Sensor Networks and Information Processing; Sydney, Dec 2008.

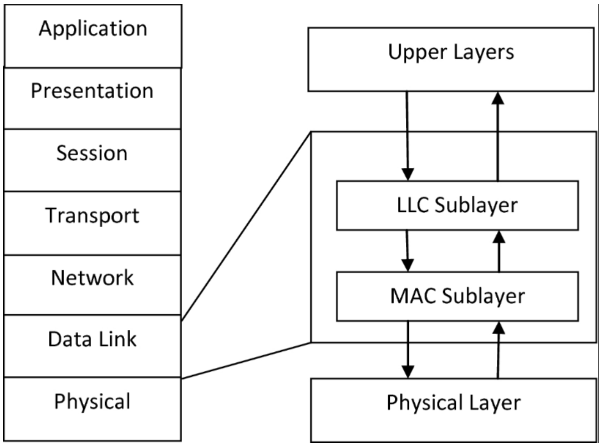
13. Weiling, L.; Shantung, T. Recent Developments of thermoelectric power generation. Chin. Sci. Bull. 2004, 49, 1212-1219.

CHAPTER 4

MEDIUM ACCESS CONTROL PROTOCOLS FOR WIRELESS SENSOR NETWORKS

INTRODUCTION

Wireless Sensor Networks (WSN) consist of a large number of battery-powered sensors capable of communicating wireless. They are distributed within an area of interest in order to track, measure and monitors various events. They are often deployed in an ad-hoc fashion, without careful planning. They must be organized so as to transmit measured data to the fusion center, which is usually done using multihop communication.



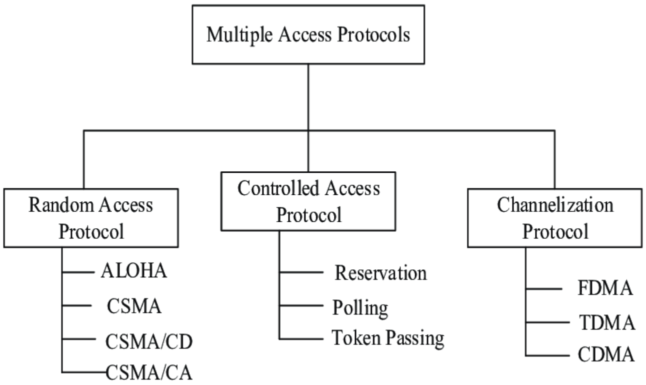
Protocols for these networks need to be extremely adaptable and scalable because of constant changes in network topology (caused by node movement and nature of wireless communication). If high energy-efficiency demands are also considered, it becomes clear that the design of MAC protocols for WSN is a difficult task.

4.1 MEDIUM ACCESS CONTROL (MAC) PROTOCOL

The Media Access Control (MAC) data communication Networks protocol sub-layer, also known as the Medium Access Control, is a sub-layer of the data link layer specified in the seven-layer OSI model. The medium access layer was made necessary by systems that share a common communications medium. Typically these are local area networks. The MAC layer is the "low" part of the second OSI layer, the layer of the "data link". In fact, the IEEE divided this layer into two layers "above" is the control layer the logical connection (Logical Link Control, LLC) and "down" the control layer the medium access (MAC).

In WSN, nodes usually have to share a common channel. Therefore, the MAC sublayer task is to provide fair access to channels by avoiding possible collisions. The main goal in MAC protocol design for WSN is energy efficiency in order to prolong the lifetimes of sensors. The reasons for the unnecessary energy waste in wireless communication are:

- Packet collision: It can occur when nodes don't listen to the medium before transmitting. Packets transmitted at the same time collide, become corrupted and must be retransmitted. This causes unnecessary energy waste.
- Overhearing: A node receives a packet which is addressed to another node.
- Control packet overhead: Control packets are necessary for successful data transmission. They don't, however, represent useful data. They are very short.
- Idle listening: The main reason for energy waste is when a node listens to an idle channel waiting to receive data.
- Over emitting: The node sends data when the recipient node is not ready to accept incoming transmission.



In order to satisfy WSN needs, the MAC protocols have to fulfill the following requirements:

- Energy efficiency: Most sensor nodes are battery powered and prolonging their lifetime is possible by designing energy-efficient protocols.
- Collision avoidance: The main goal is to reduce collisions as much as possible. This can be achieved either by listening to the channel (CSMA) or by using time (TDMA), frequency (FDMA) or code (CDMA) channel division access.
- Scalability and adaptability: The MAC protocol needs to be adaptable to changes in network topology caused by node movement and nature of wireless transmission.
- Latency: Latency represents the delay of a packet when sent through the network. The importance of latency in wireless sensor networks depends on the monitoring application.
- Throughput: Represents the amount of data within a period of time sent from the sender to the receiver through WSN.
- Fairness: The MAC protocol needs to provide fair medium access for all active nodes.

4.1.1 Issues in designing a MAC protocol for Ad hoc wireless Networks

Following are the main issues one should have in mind when considering designing a MAC protocol for ad hoc wireless networks.

Bandwidth Efficiency

The scarcity of bandwidth resources in these networks calls for its efficient usage. To quantify this, we could say that bandwidth efficiency is the ratio of the bandwidth utilized for data transmission to the total available bandwidth. In these terms, the target will be to maximize this value.

Quality of Service Support

Providing QoS in these networks is very difficult, due to the high mobility of the nodes comprising them. Once a node moves out of another node's reach, the reservation in it is lost. On the other hand, in these networks QoS is sometimes extremely important, for example in military environments. Therefore, QoS should be provided somehow, despite the characteristics of ad hoc networks.

Synchronization

Some mechanism has to be found in order to provide synchronization among the nodes. Synchronization is important for regulating the bandwidth reservation.

Hidden and Exposed Terminal Problems

The reason for these two problems is the broadcast nature of the radio channel, namely, all the nodes within a node's transmission range receive its transmission.

- Hidden terminal problem – two nodes that are outside each-other's range perform simultaneous transmission to a node that is within the range of each of them, hence, there is a packet collision.
- Exposed terminal problem – the node is within the range of a node that is transmitting, and it cannot transmit to any node.

Hidden nodes mean increased probability of collision at a receiver, whereas exposed nodes may be denied channel access unnecessarily, which means underutilization of the bandwidth resources.

Error-prone shared broadcast channel

In radio transmission, a node can listen to all traffic within its range. Therefore, when there is communication going on no other node should transmit, otherwise there would be interferences. Access to the physical medium should be granted only if there is no session going on. Nodes will often compete for the channel at the same time; therefore, there is high probability of collisions. The aim of a MAC protocol will be to minimize them, while maintaining fairness.

No central coordination

In ad hoc networks, there is no central point of coordination due to the mobility of the nodes. Therefore, the control of the access to the channel must be distributed among them. In order for this to be coordinated, the nodes must exchange information. It is the responsibility of the MAC protocol to make sure this overhead is not a burden for the scarce bandwidth.

Mobility of nodes

The mobility of the nodes is one of its key features. The QoS reservations or the exchanged information might become useless, due to node mobility. The MAC protocol must be such that mobility has as little influence as possible on the performance of the whole network.

Signal propagation delay

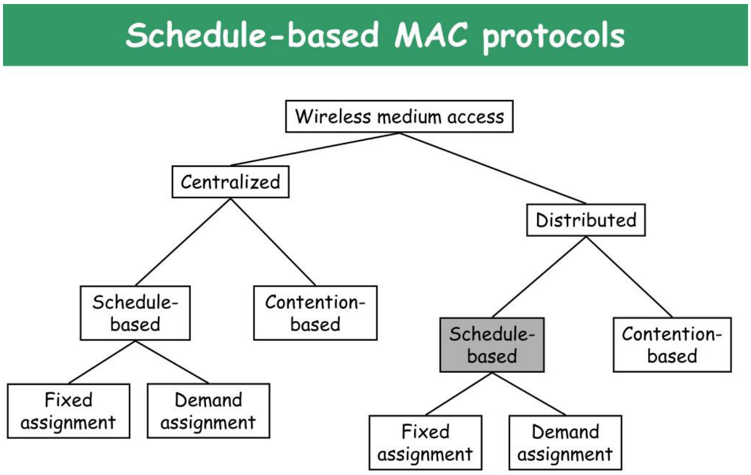
Signal propagation delay is the amount of time needed for the transmission to reach the receiver. If the value of this parameter is considerable, a node may start transmitting, when in fact, transmission from other nodes is taking place, but it has not reached the node yet. The ad hoc networks that utilize synchronization, therefore, will have to expand the time slot to accommodate the propagation delay.

Hardware Constraints

Most radio-receivers are designed in such a way that only halfduplex communication can take place. When a node is transmitting, the power level of the outgoing signal is higher than any received signal; therefore, the node receives its own transmission. Here, we can also add hardware switching time – time needed to shift from one mode to the other.

4.1.2 Scheduled MAC Protocols

The scheduled MAC protocol is based on the Time Division Multiple Access (TDMA). In each slot, only one node is allowed to transmit. Nodes are organized into clusters. A center of each cluster have the cluster head. It is responsible for all communication inside the cluster as well as for inter-cluster communication. It also takes care of channel time division and time synchronization of nodes. The Frequency (FDMA) or Code (CDMA) division can be used in order to avoid interference at intercluster communication.



There are no collisions in the schedule-based protocols, as only one node at a time is allowed to transmit. There is also no overhearing or idle-listening. When a node's time slot expires, it goes back to the sleep mode. The disadvantage of these protocols is lack of peer-topeer connection.

Consequently, nodes can only communicate with a cluster head. These protocols are also poorly adaptable and scalable. When a node joins or leaves a cluster, the cluster head needs to redefine the whole framework timetable and synchronize all nodes inside the cluster. There is also huge pressure on the cluster head which has to be a unit exercising typical node performance. Because of clock drifts in the cluster of nodes, the time synchronization must also be precisely kept. Two examples of the scheduled protocols are LEACH (Low Energy Adaptive Clustering Hierarchy) and Bluetooth.

LEACH

Two versions of the LEACH protocol exist: distributed (LEACH-D) and centralized (LEACH-C) LEACH. In both versions, nodes are organized in clusters with TDMA within each cluster. In LEACH-D, the role of the cluster head is randomly rotated among all nodes in the network. The protocol is organized in rounds which consist of startup and transmission phases. In the startup phase, the nodes organize themselves into clusters, where the cluster head is picked up randomly. During the transmission phase, the cluster head collects data for the nodes within the cluster and applies data fusion before sending them to the base station.

In LEACH-C, the base station decides which node will be the cluster head. The role of the cluster head is selected by the node location and its remaining energy level.

Bluetooth

The Bluetooth standard has been developed for personal area networks (PAN) where nodes are laptop computers, PDAs, cell phones, etc. The nodes are organized into clusters, called piconets. Each piconet consists of one master node and up to seven slave nodes.

DMA is used within a cluster and frequency-hopping CDMA for intercluster communication. The master node's role is usually given to the node which starts the piconet. This one is responsible for time synchronization and traffic control inside piconet. Larger networks are constructed as scatternet. In such a case, a border node is used to bridge two piconets together (Figure 1).

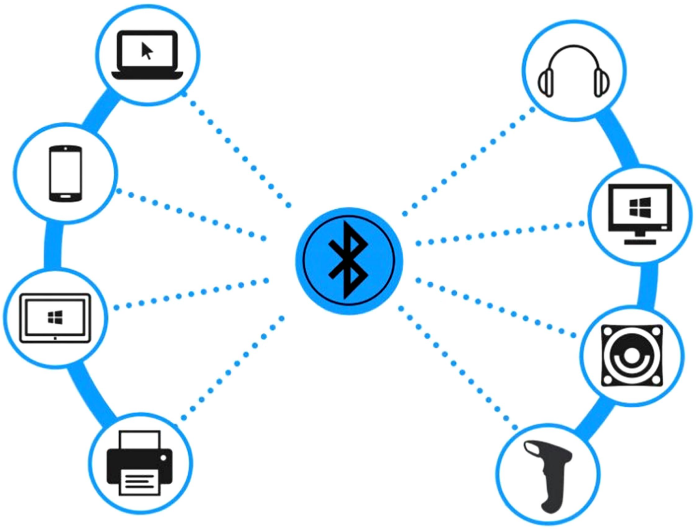
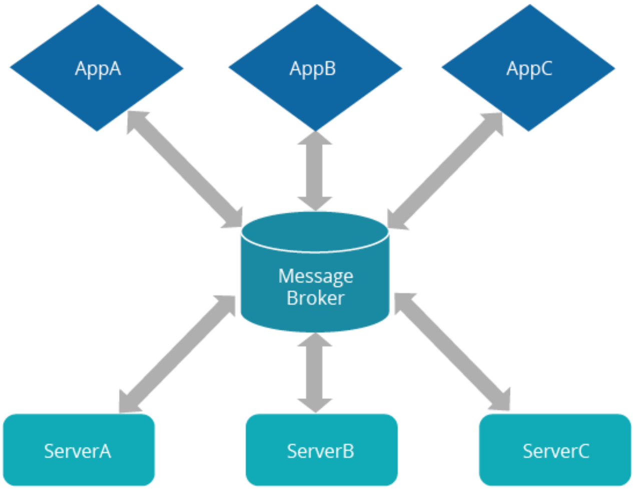


Figure 1: Bluetooth networks organizations.

4.2 EVENT-DRIVEN PROTOCOLS

Unlike the scheduled protocols, event-driven protocols do not pre-allocate the channel for each node, regardless of whether they have data to send or not. Instead, they allocate a channel only to those nodes which need to send data. A major advantage over the schedule-based protocols it is that these protocols are more adaptable to network topology changes. They are also susceptible to changes in the node density and changes in the traffic load. They support peer-to-peer communication, so there is no need for communication clusters. They also don't require time synchronization as is the case with the TDMA protocols.

The disadvantage of these protocols is in idle listening and overhearing. A node needs to listen to a medium if it is available before transmitting data. This leads to energy waste. Energy is also wasted due to frequent collisions during the transmission.



4.2.1 Aloha

Aloha was one of the first attempts to design the MAC protocol for regular networks. Its main idea is that the transmitter sending packets whenever it wants without the need for coordination between nodes. LOHA is a system for coordinating and arbitrating access to a shared communication Networks channel. It was developed in the 1970s by Norman Abramson and his colleagues at the University of Hawaii. The original system used for ground based radio broadcasting, but the system has been implemented in satellite communication systems.

A shared communication system like ALOHA requires a method of handling collisions that occur when two or more systems attempt to transmit on the channel at the same time. In the ALOHA system, a node transmits whenever data is available to send. If another node transmits at the same time, a collision occurs, and the frames that were transmitted are lost. However, a node can listen to broadcasts on the medium, even its own, and determine whether the frames were transmitted.

Aloha means "Hello". Aloha is a multiple access protocol at the datalink layer and proposes how multiple terminals access the medium without interference or collision. In 1972 Roberts developed a protocol that would increase the capacity of aloha two fold. The Slotted Aloha protocol involves dividing the time interval into discrete slots and each slot interval corresponds to the time period of one frame. This method requires synchronization between the sending nodes to prevent collisions.

There are two different types of ALOHA:

- Pure ALOHA
- Slottecl ALOHA

Pure ALOHA

- In pure ALOHA, the stations transmit frames whenever they have data to send.
- When two or more stations transmit simultaneously, there is collision and the frames are destroyed.
- In pure ALOHA, whenever any station transmits a frame, it expects the acknowledgement from the receiver.
- If acknowledgement is not received within specified time, the station assumes that the frame (or acknowledgement) has been destroyed.
- If the frame is destroyed because of collision the station waits for a random amount of time and sends it again. This waiting time must be random otherwise same frames will collide again and again.
- Therefore pure ALOHA dictates that when time-out period passes, each station must wait for a random amount of time before resending its frame. This randomness will help avoid more collisions.
- In fig there are four stations that .contended with one another for access to shared channel. All these stations are transmitting frames. Some of these frames collide because multiple frames are in contention for the shared channel. Only two frames, frame 1.1 and frame 2.2 survive. All other frames are destroyed.
- Whenever two frames try to occupy the channel at the same time, there will be a collision and both will be damaged. If first bit of a new frame overlaps with just the last bit of a frame almost finished, both frames will be totally destroyed and both will have to be retransmitted.

Slotted ALOHA

- Slotted ALOHA was invented to improve the efficiency of pure ALOHA as chances of collision in pure ALOHA are very high.
- In slotted ALOHA, the time of the shared channel is divided into discrete intervals called slots.

- The stations can send a frame only at the beginning of the slot and only one frame is sent in each slot.
- In slotted ALOHA, if any station is not able to place the frame onto the channel at the beginning of the slot *i.e.* it misses the time slot then the station has to wait until the beginning of the next time slot.
- In slotted ALOHA, there is still a possibility of collision if two stations try to send at the beginning of the same time slot.
- Slotted ALOHA still has an edge over pure ALOHA as chances of collision are reduced to one-half.

Protocol Flow Chart for ALOHA:

Figure shows the protocol flow chart for ALOHA.

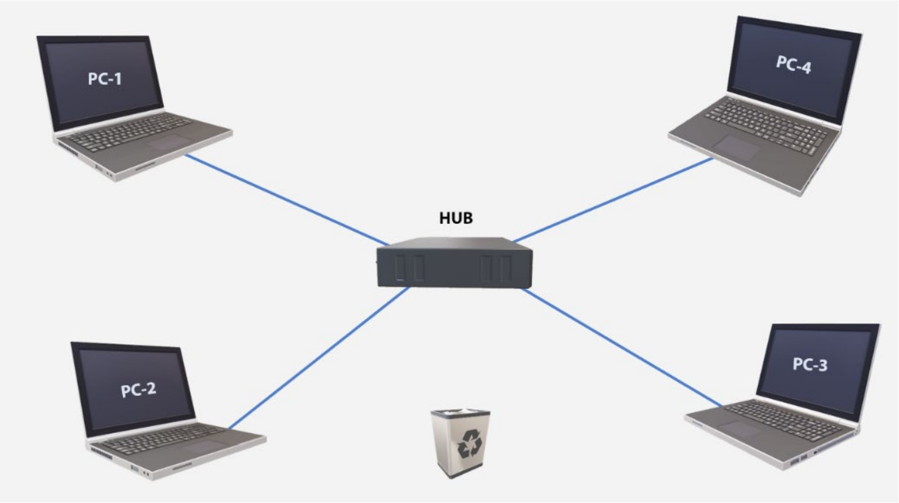
Explanation:

- A station which has a frame ready will send it.
- Then it waits for some time.
- If it receives the acknowledgement then the transmission is successful.
- Otherwise the station uses a back off strategy, and sends the packet again.
- After many times if there is no acknowledgement then the station aborts the idea of transmission.

4.2.2 Carrier Sensed Multiple Access (CSMA)

CSMA is a network access method used on shared network topologies such as Ethernet to control access to the network. Devices attached to the network cable listen (carrier sense) before transmitting. If the channel is in use, devices wait before transmitting. MA (Multiple Access) indicates that many devices can connect to and share the same network. All devices have equal access to use the network when it is clear.

In other words, a station that wants to communicate "listen" first on the media communication and awaits a "silence" of a preset time (called the Distributed Inter Frame Space or DIFS). After this compulsory period, the station starts a countdown for a random period considered. The maximum duration of this countdown is called the collision window (Window Collision, CW). If no equipment speaks before the end of the countdown, the station simply deliver its package. However, if it is overtaken by another station, it stops immediately its countdown and waits for the next silence. She then continued his account countdown where it left off. This is summarized in Figure. The waiting time random has the advantage of allowing a statistically equitable distribution of speaking time between the various network equipment, while making little unlikely (but not impossible) that both devices speak exactly the same time. The countdown system prevents a station waiting too long before issuing its package. It's a bit what place in a meeting room when no master session (and all the World's polite) expected a silence, then a few moments before speaking, to allow time for someone else to speak. The time is and randomly assigned, that is to say, more or less equally.



CSMA protocol was developed to overcome the problem found in ALOHA i.e. to minimize the chances of collision, so as to improve the performance. CSMA protocol is based on the principle of 'carrier sense'. The station senses the carrier or channel before transmitting a frame. It means the station checks the state of channel, whether it is idle or busy.

Even though devices attempt to sense whether the network is in use, there is a good chance that two stations will attempt to access it at the same time. On large networks, the transmission time between one end of the cable and another is enough that one station may access the cable even though another has already just accessed it.

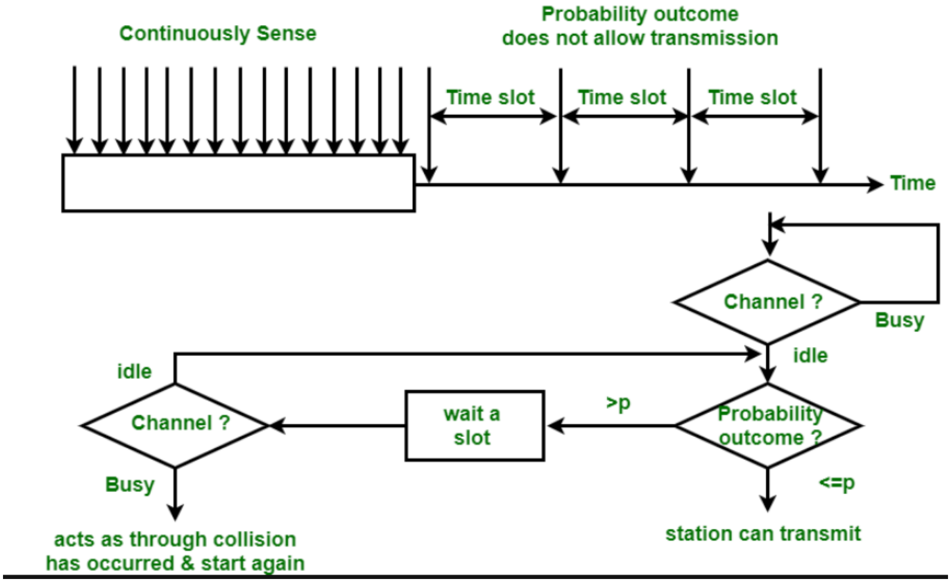
The chances of collision still exist because of propagation delay. The frame transmitted by one station takes some time to reach other stations. In the meantime, other stations may sense the channel to be idle and transmit their frames. This results in the collision.

There Are Three Different Type of CSMA Protocols

- I-persistent CSMA
- Non- Persistent CSMA
- P-persistent CSMA

I-persistent CSMA

- In this method, station that wants to transmit data continuously senses the channel to check whether the channel is idle or busy.
- If the channel is busy, the station waits until it becomes idle.
- When the station detects an idle-channel, it immediately transmits the frame with probability 1. Hence it is called I-persistent CSMA.
- This method has the highest chance of collision because two or more stations may find channel to be idle at the same time and transmit their frames.
- When the collision occurs, the stations wait a random amount of time and start all over again.



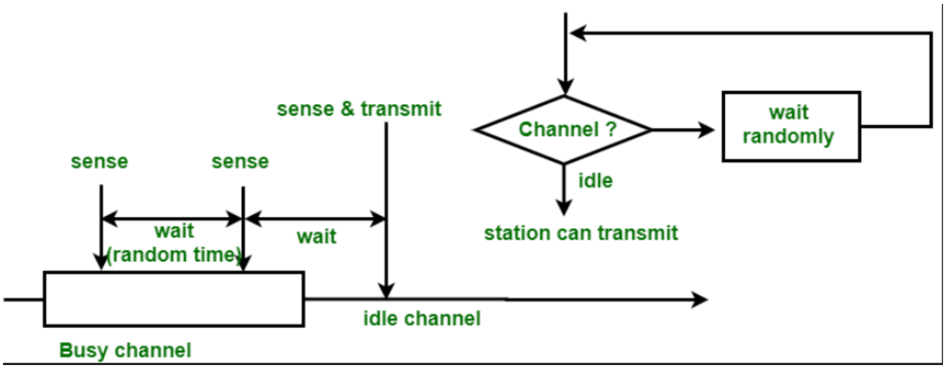
Drawback of I-persistent

- The propagation delay time greatly affects this protocol. Let us suppose, just after the station I begins its transmission, station 2 also became ready to send its data and senses the channel. If the station I signal has not yet reached station 2, station 2 will sense the channel to be idle and will begin its transmission. This will result in collision.

Even if propagation delay time is zero, collision will still occur. If two stations became .ready in the middle of third station's transmission, both stations will wait until the transmission of first station ends and then both will begin their transmission exactly simultaneously. This will also result in collision.

Non-persistent CSMA

- In this scheme, if a station wants to transmit a frame and it finds that the channel is busy (some other station is transmitting) then it will wait for fixed interval oftime.
- After this time, it again checks the status of the channel and if the channel is.free it will transmit.
- A station that has a frame to send senses the channel.
- If the channel is idle, it sends immediately.
- If the channel is busy, it waits a random amount of time and then senses the channel again.
- In non-persistent CSMA the station does not continuously sense the channel for the purpose of capturing it when it detects the end of previous transmission.



Advantage of non-persistent

- It reduces the chance of collision because the stations wait a random amount of time. It is unlikely that two or more stations will wait for same amount of time and will retransmit at the same time.

Disadvantage of non-persistent

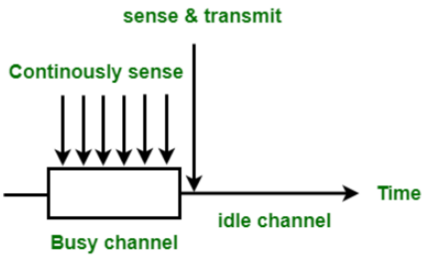
- It reduces the efficiency of network because the channel remains idle when there may be stations with frames to send. This is due to the fact that the stations wait a random amount of time after the collision.

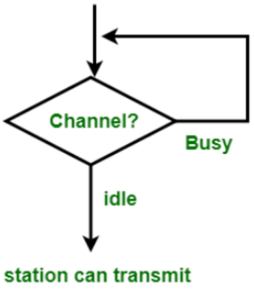
p-persistent CSMA

- This method is used when channel has time slots such that the time slot duration is equal to or greater than the maximum propagation delay time.
- Whenever a station becomes ready to send, it senses the channel.
- If channel is busy, station waits until next slot.
- If channel is idle, it transmits with a probability p.
- With the probability q=l-p, the station then waits for the beginning of the next time slot.
- If the next slot is also idle, it either transmits or waits again with probabilities p and q.
- This process is repeated till either frame has been transmitted or another station has begun transmitting.
- In case of the transmission by another station, the station acts as though a collision has occurred and it waits a random amount of time and starts again.

Advantage of p-persistent

- It reduces the chance of collision and improves the efficiency of the network.





4.2.3 Carrier Sense Multiple Access with Collision Detection (CSMA/CD)

To reduce the impact of collisions on the network performance, Ethernet uses an algorithm called CSMA with Collision Detection (CSMA / CD): CSMA/CD is a protocol in which the station senses the carrier or channel before transmitting frame just as in persistent and non-persistent CSMA. If the channel is busy, the station waits. It listens at the same time on communication media to ensure that there is no collision with a packet sent by another station. In a collision, the issuer immediately cancel the sending of the package. This allows to limit the duration of collisions: we do not waste time to send a packet complete if it detects a collision. After a collision, the transmitter waits again silence and again, he continued his hold for a random number; but this time the random number is nearly double the previous one: it is this called back-off (that is to say, the "decline") exponential. In fact, the window collision is simply doubled (unless it has already reached a maximum). From a packet is transmitted successfully, the window will return to its original size.

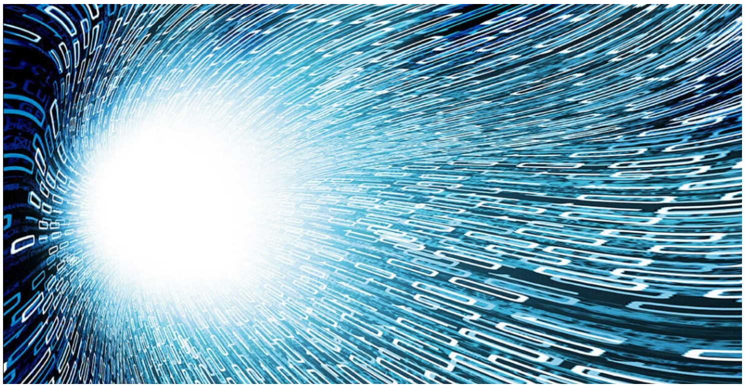
Again, this is what we do naturally in a meeting room if many people speak exactly the same time, they are realizing account immediately (as they listen at the same time they speak), and they interrupt without completing their sentence. After a while, one of them speaks again. If a new collision occurs, the two are interrupted again and tend to wait a little longer before speaking again.

Frame format of CSMA/CD

The frame format specified by IEEE 802.3 standard contains following fields.

- Preamble: It is seven bytes (56 bits) that provides bit synchronization. It consists of alternating Os and 1s. The purpose is to provide alert and timing pulse.
- Start Frame Delimiter (SFD): It is one byte field with unique pattern: 10 10 1011. It marks the beginning of frame.
- Destination Address (DA): It is six byte field that contains physical address of packet's destination.
- Source Address (SA): It is also a six byte field and contains the physical address of source or last device to forward the packet (most recent router to receiver).

- Length: This two byte field specifies the length or number of bytes in data field.
- Data: It can be of 46 to 1500 bytes, depending upon the type of frame and the length of the information field.
- Frame Check Sequence (FCS): This for byte field contains CRC for error detection.



4.3 CLASSIFICATION OF MAC PROTOCOLS

Ad hoc network MAC protocols can be classified into three basic types:

- Contention-based protocols
- Contention-based protocols with reservation mechanisms
- Contention-based protocols with scheduling mechanisms

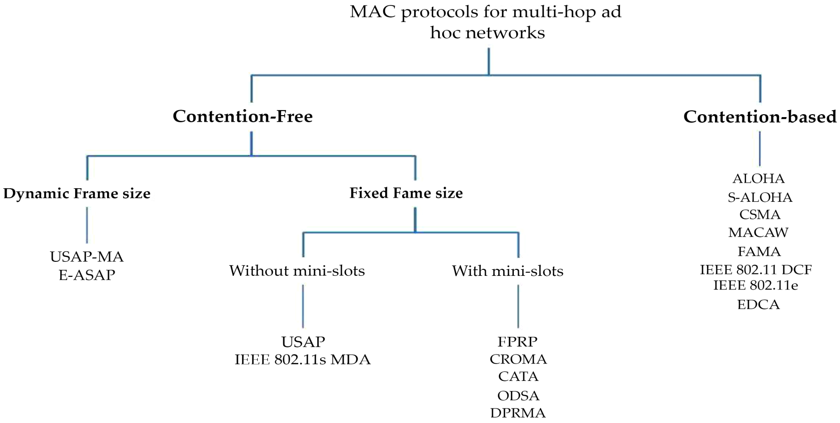
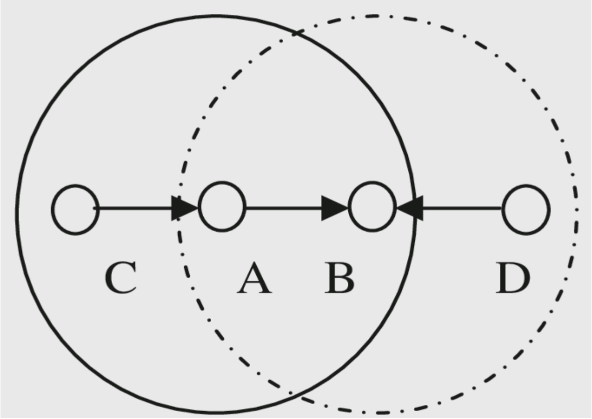


Figure 2: Classification of the MAC protocols for ad hoc networks

4.3.1 Contention-Based Protocols

- Sender-initiated protocols: Packet transmissions are initiated by the sender node.
- Single-channel sender-initiated protocols: A node that wins the contention to the channel can make use of the entire bandwidth.
- Multichannel sender-initiated protocols: The available bandwidth is divided into multiple channels.
- Receiver-initiated protocols: The receiver node initiates the contention resolution protocol.



4.3.2 Contention-based protocols with reservation mechanisms

These protocols provide bandwidth reservation ahead; therefore, they can provide QoS support. These can be further subdivided into:

- Synchronous protocols: there is time synchronization among all nodes in the network, the nodes in the neighborhood are informed of the reservations
- Asynchronous protocols: no global synchronization is needed. Relative time is used for the reservations.

Even though these protocols are contention-based, the contention takes place only during the bandwidth reservation phase.

Distributed Packet Reservation Multiple Access (D-PRMA) Protocol

D-PRMA is based on TDMA. The time division of the channel is done into frames, then further into slots, then further into minislots. Each minislot contains two control fields, RTS/BI – Request To Send / Busy Indication and CTS/BI – Request To Send / Busy Indication.

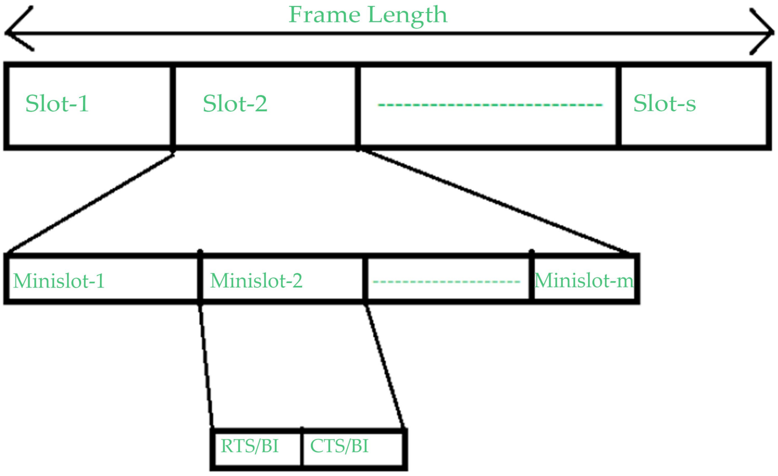


Figure 3: Time Division in the D-PRMA protocol.

The mechanism of competition for slots is such that a certain period at the beginning of every slot is reserved for carrier-sensing. The nodes compete for the first minislot in each slot. The winning one transmits a RTS packet through the RTS/BI part of the first minislot. The receiver responds by sending a CTS packet through the CTS/BI field. Thus, the node is granted all the subsequent minislots. In addition to that, this very same slot in the subsequent frames is reserved for the same node, until it ends its transmission. Within a time slot, communication between the source and destination nodes is done either by Time Division Duplexing (TDD), or by Frequency Division Duplexing (FDD).

There are two rules for the reservation, which prioritize voice traffic:

- Contention for the first minislot is done with probability 1 for voice traffic, and a smaller probability for other traffic.
- The reservation of a minislot brings reservation of the subsequent slots only if the winning node is a voice one.

Collision Avoidance Time Allocation (CATA) Protocol

In this protocol, time is divided into frames, each frame into slots, and each slot into 5 minislots. The first four minislots are control ones, CMS, only the fifth is used for data transmission, DMS, and it is longer than the other ones.

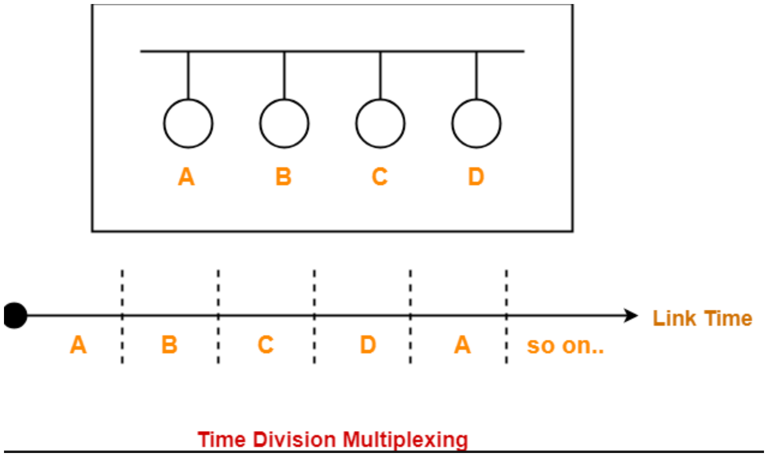


Figure 4: Time Division in the CATA protocol.

CATA supports broadcast, unicast, and multicast transmissions at the same time. CATA has two basic principles:

- The receiver of a flow must inform other potential source nodes about the reservation of the slot, and also inform them about interferences in the slot.
- Negative acknowledgements are used at the beginning of each slot for distributing slot reservation information to senders of broadcast or multicast sessions.

The CMS1 and CMS2 are used to inform neighbors of the receiving and the sending nodes accordingly about the reservation. The CMS3 and CMS4 are used for channel reservation.

CATA provides support for collision-free broadcast and multicast traffic.

Hop Reservation Multiple Access (HRMA) Protocol

HRMA is a time slot-reservation protocol where each slot is assigned a separate frequency channel. A handshake mechanism is used for reservation to enable node pairs to reserve a frequency hop, thus providing collision-free communication and avoiding the hidden terminal problem.

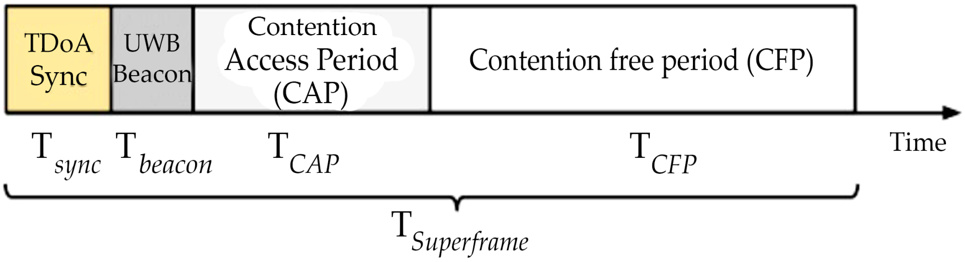


Figure 5: Time Division in the HRMA protocol.

One frequency channel is a dedicated synchronizing channel where nodes exchange information. The remaining frequency channels are paired, one channel in each pair is used for reservation and data packets, and the other one is used for acknowledgements. As mentioned above, each time slot has a frequency channel. The time slot is divided into four periods, each period is reserved for sending a particular kind of packet or its acknowledgement, depending on which frequency channel of the pair this time slot belongs to.

After the handshaking is over, the two nodes communicate by sending data and ACKs on the very same frequency channels. When a new node wants to join the network, it listens to the dedicated frequency and gathers information. When a node wants to send data, it listens to the Hop Reservation (HR) period. If there is a packet there, it tries again after a random amount of time, otherwise it sends a RTS packet, and waits for the CTS acknowledgement packet in the CTS period of the corresponding frequency channel.

MACA with Piggy-backed Reservation (MACA/PR) Protocol

There are three main components of this protocol:

- A MAC protocol;
- A reservation protocol;
- A QoS routing protocol.

MACA/PR differentiates between real-time packets and best-effort packets; it provides bandwidth to real-time traffic. Time is divided into slots that are asynchronous in nature and have different lengths. Each node records the transmit and receive reservations of its neighbors in a reservation table. A node that wants to transmit a non-real-time packet finds a free slot in the table. Then it waits for the same slot the next time around. If it is still free, it sends a RTS packet in the slot, expects a CTS packet, then sends the data and receives the acknowledgement still in the same slot. The RTS and CTS packets contain in them the amount of time that the data transmission is going to take place. In this way, the neighbors of the source and destination nodes can update their tables.

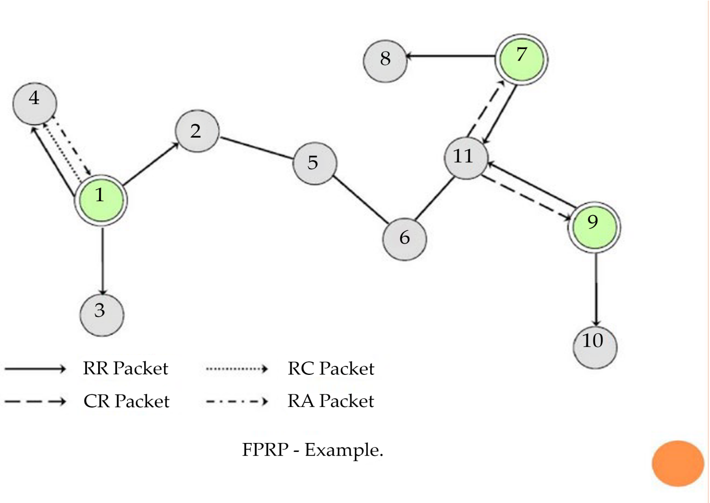


Figure 6: Packet exchange in MACA/PR.

For real-time traffic, the first part is identical until the first data packet is sent. Each data packet contains information about the reservation of the next data packet. This information is piggy-backed to it. Each acknowledging packet also contains this information. Thus, the neighbors of both communicating nodes can update the information. When the sender receives the acknowledgement, it makes sure that the reservation was successful. If several acknowledgements do not come, the sender assumes that the reservation has been lost and restarts the whole procedure. The acknowledgement refreshes the reservation; unrefreshed ones are simply dropped from the reservation table. The nodes exchange the information in their reservation tables; this eliminates the hidden terminal problem. This mechanism works as a TDM for real-time traffic, while best-effort packets are transmitted in the empty slots. When a new node joins the network, at first it learns about it by receiving the reservation tables from the neighbors. Then it behaves just like the others.

Advantage: global synchronization not required.

Drawback: the RTS-CTS-DATA-ACK exchange takes place in the same slot in different cycles; therefore, random empty slots are not utilized.

4.3.3 Contention-Based Protocols with Scheduling Mechanisms

There can be packet scheduling at the nodes, or node scheduling for access to the channel. Node scheduling should not treat the nodes unfairly. Some of these protocols consider battery power in their node scheduling.

- Node scheduling is done in a manner so that all nodes are treated fairly and no node is starved of bandwidth.
- Scheduling-based schemes are also used for enforcing priorities among flows whose packets are queued at nodes.
- Some scheduling schemes also consider battery characteristics.

These protocols handle packet scheduling at the nodes and scheduling of the nodes. Rather than aiming to provide node or flow level fairness, the target here is an ordering mechanism to achieve either QoS differentiation or fairness. In other words, the objective is to design a distributed MAC protocol in which, packets are serviced to the maximum extent possible in the order defined by a reference scheduler.

Distributed Priority Scheduling (DPS)

DPS uses the RTS-CTS-DATA-ACK packet exchange mechanism. The RTS packet contains the priority label for the DATA packet that is to be transmitted. The corresponding CTS contains it also. Neighbor nodes receiving the RTS packet update their scheduling tables with the node and its priority. When the DATA packet is sent, it contains piggybacked information about the priority of the next packet from this node and its priority, so the other nodes can record this information. Finally, when the acknowledgement comes, the nodes delete the entry for the data packet that is being acknowledged. This mechanism enables the nodes to evaluate their priority in relation to the priority of the other nodes.

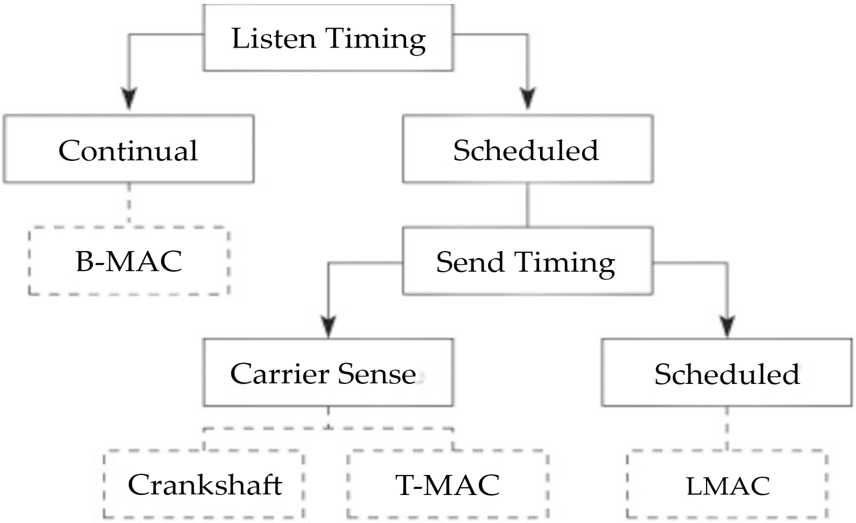
Distributed Wireless Ordering Protocol (DWOP)

The purpose of this protocol is to achieve a distributed FIFO schedule among multiple nodes in an ad hoc network. When a node transmits a packet, it adds the information about the arrival time of queued packets. All nodes overhear this information and record it in their local scheduling table. This information helps a node establish its relative priority in relation to the partial list of the nodes within its range, associated with their arrival times. According to DWOP, a node should contend for channel access only when it has the lowest arrival time of all the nodes within its range.

Entries in the table are deleted when the node hears the ACK packet. In case these ACK packets are not heard, we end up having false entries in the tables. This can be detected if a node notices other packets with lower priority being transmitted; it can solve the problem by deleting the oldest entry. In case not all the nodes are within radio range of each-other, incomplete table information will lead to collisions, and will prevent a pure FIFO scheduling from happening.

4.4 MAC PROTOCOLS USED IN WSN

There are many MAC protocols adapted for different WSN applications. They differ in channel utilization, complexity and efficiency regarding energy saving.



4.4.1 Sensor MAC

In the Sensor MAC (SMAC) protocols, nodes form virtual clusters with one common sleep schedule, so all the clusters wake up and start communicating at the same time (Figure 7). The channel is divided into an active and sleeping period. Potential energy saving is determined by the ratio between the active and passive periods during the active period. The node starting a synchronization sequence is called synchronizer. It emits an SYNC packet which synchronizes all nodes inside the virtual cluster. Collision avoidance is achieved by the carrier sense and by the data exchange schemes RTS/CTS/ DATA/ACK.

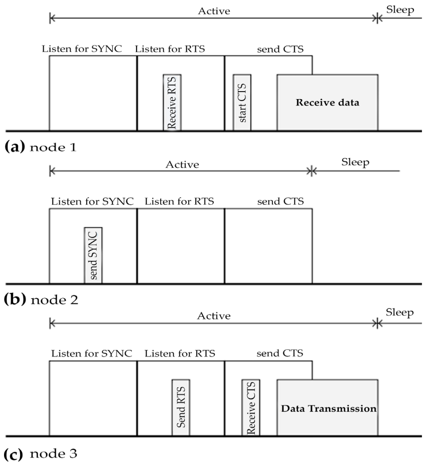


Figure 7: SMAC protocol.

This protocol has one major problem. It is addressed to border nodes which are located at the cross-section of two virtual clusters (Figure 8). In order to connect virtual clusters in one network, these nodes have to transmit all traffic from one cluster to another towards the sink node, therefore, they need to follow both sleeping schedules. Consequently, these nodes can quickly deplete their batteries. This problem can be solved by frequently changing synchronizer allocation inside virtual clusters which causes borders to move between clusters.

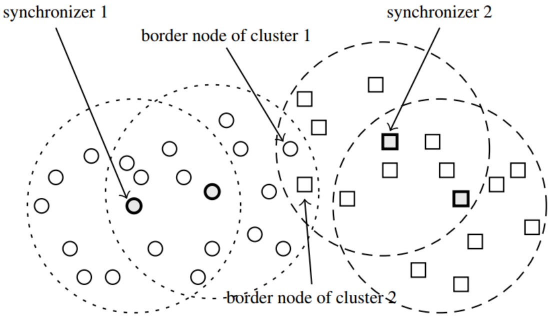


Figure 8: Problem of bordering nodes in SMAC.

Energy efficiency in SMAC is proportional to the ratio between active and sleeping periods. This ratio is constant regardless of traffic intensity. When traffic is low, most of the active-period nodes listen to an idle channel. When traffic is heavy, only some of nodes can use active period so they buffer data which they cannot send. This problem increases the packet latency.

4.4.2 TMAC

The TMAC protocol is an extension of the SMAC protocol for the time-division based approach. Weakness of the SMAC protocol can be solved by introducing an adaptive active period. All communication during the active period is done in one burst. When all communications are over, nodes still listen to the medium t_a seconds for any communication demand left. After that they go into an early sleeping-mode (Figure 9). When traffic is heavy, the active period finishes after all the nodes have sent their packets.

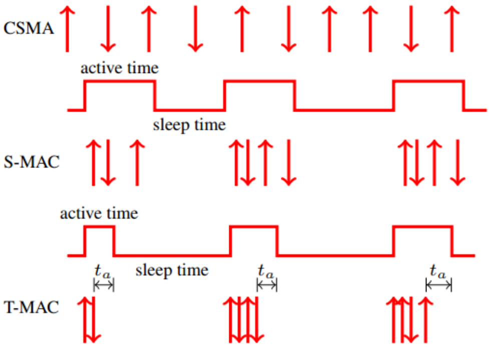


Figure 9: TMAC protocol.

A major advantage of the TMAC over the SMAC protocol is in the adaptive frame time. In the SMAC protocol, as the traffic load changes the duty cycle needs to be changed in order to operate efficiently. The TMAC protocol adapts to changes in network traffic by itself. TMAC also supports overhearing avoidance, full-buffer priority and Future Ready To Send (FRTS) packets.

4.4.3 WiseMAC

The WiseMAC protocol represents an extension of Aloha with Preamble Sampling. A big disadvantage of preamble sampling is the long preamble that has to be received by all receivers, even if they are not addressed. In WiseMAC, the sender does not send a packet immediately, but shortly before the receiver is expected to wake up. Receivers wake up at constant intervals, probe the channel and if it is idle, they go back to sleep. If the receiver detects a preamble, it stays awake and receives packet. Each node must have the wake-up time table of its neighbors. Before transmission, it waits until the neighbor wakes up (Figure 10). The protocol works as follows. The sender node doesn't start its preamble immediately, but waits until the neighbor node is expected to wake-up. The sender can spend this time in the sleep-mode to preserve energy. Preamble starts a short time before the anticipated wake-up time of the receiver in order to compensate for potential clock drifts between nodes. When the receiver node wakes up, it can hear the preamble and waits to start receiving data. A successful data reception is confirmed by the ACK frame in which the receiver piggy-backs its time schedule.

The WiseMAC protocol does not have problems with idle listening and doesn't suffer from energy waste caused by long preamble, as the case with Aloha with preamble sampling. A disadvantage of this protocol is the need for clock synchronization and scheduling tables which need to be constantly updated.

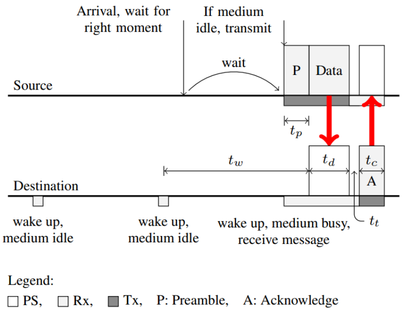


Figure 10: Example of communication using the WiseMAC protocol. PS: prepare state, where a node is able to quick power on to Rx or Tx state.

SUMMARY

- Wireless Sensor Networks (WSN) consist of a large number of battery-powered sensors capable of communicating wireless. They are distributed within an area of interest in order to track, measure and monitors various events. They are often deployed in an ad-hoc fashion, without careful planning.
- The Media Access Control (MAC) data communication Networks protocol sub-layer, also known as the Medium Access Control, is a sub-layer of the data link layer specified in the seven-layer OSI model.
- In WSN, nodes usually have to share a common channel. Therefore, the MAC sublayer task is to provide fair access to channels by avoiding possible collisions. The main goal in MAC protocol design for WSN is energy efficiency in order to prolong the lifetimes of sensors.
- The scarcity of bandwidth resources in these networks calls for its efficient usage. To quantify this, we could say that bandwidth efficiency is the ratio of the bandwidth utilized for data transmission to the total available bandwidth. In these terms, the target will be to maximize this value.
- In radio transmission, a node can listen to all traffic within its range. Therefore, when there is communication going on no other node should transmit, otherwise there would be interferences.
- Most radio-receivers are designed in such a way that only halfduplex communication can take place. When a node is transmitting, the power level of the outgoing signal is higher than any received signal; therefore, the node receives its own transmission. Here, we can also add hardware switching time – time needed to shift from one mode to the other.

REFERENCES

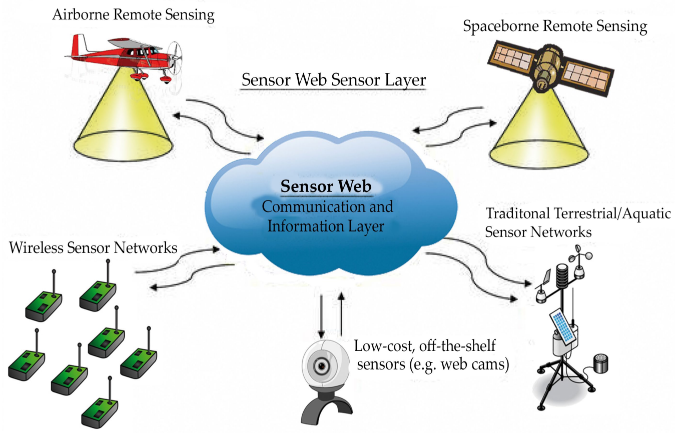
1. N. Abramson, “The ALOHA System-Another Alternative for Computer Communications,” Proc. Fall Joint Computer Conf., NJ, 1970, Vol. 37, pp. 281-285.
2. IETF MANET Working Group, http://www.ietf.org/html.charters/manet-charter.html.
3. S. Chakrabarti and A Mishra, “QoS Issues in Ad Hoc Wireless Networks,” IEEE Commun. Mag., Vol. 39(2), Feb. 2001, pp. 142-48.
4. Z. J. Haas and S. Tabrizi, “On Some Challenges and Design Choices in Ad-Hoc Communications,” Proc. IEEE MILCOM98, Vol. 1, 1998.
5. E. M. Royer and C. K. Toh, “A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks,” IEEE Personal Commun., Vol. 6(2), April 1999, pp. 46-55.
6. C.-K. Toh, Ad Hoc Mobile Wireless Networks: Protocols and Systems, Prentice Hall PTR, NJ, 2002.
7. I. F. Akyildiz, J. McNair, L.C. Martorell, R. Puigjaner and Y. Yesha, “Medium access control protocols for multimedia traffic in wireless networks,” IEEE Network, Vol. 13, July/August 1999, pp. 39-47.
8. Charles E. Perkins, Ad Hoc Networking, Addison Wesley, 2001.
9. L. Kleinrock and F. A. Tobagi, “Packet Switching in Radio Channels: Part I - Carrier Sense Multiple Access Modes and Their Throughput-Delay Characteristics,” IEEE Trans. Commun., Vol. 23, pp. 1400-1416, Dec. 1975.
10. N. Poojary, S. V. Krishnamurthy and S. Dao, “Medium Access Control in a Network of Ad Hoc Mobile Nodes with Heterogeneous Power Capabilities,” Proc. IEEE ICC, Vol. 3, 2001, pp. 872-877.
11. L. Kleinrock and F. A. Tobagi, “Packet Switching in Radio Channels: Part II - The Hidden Terminal Problem in Carrier Sense Multiple Access and Busy Tone Solution,” IEEE Trans. Commun., Vol. 23, pp. 1417-1433, Dec. 1975.
12. R. G. Gallager, ``A perspective on multi access channels,’’ IEEE Trans. Inf. Th., Vol. 31(2), pp. 124-142, 1985.
13. P. Karn, “MACA - A New Channel Access Method for Packet Radio,” ARRL/CRRL Amateur Radio 9th Computer Networking Conf., Sept. 22, 1990.
14. V. Bhargavan, A. Demers, S. Shenker and L. Zhang, “MACAW: A Media Access Protocol for Wireless LANs,” Proc. ACM SIGCOMM, 1994, pp. 212-225.

CHAPTER 5

SECURITY IN WIRELESS SENSOR NETWORKS

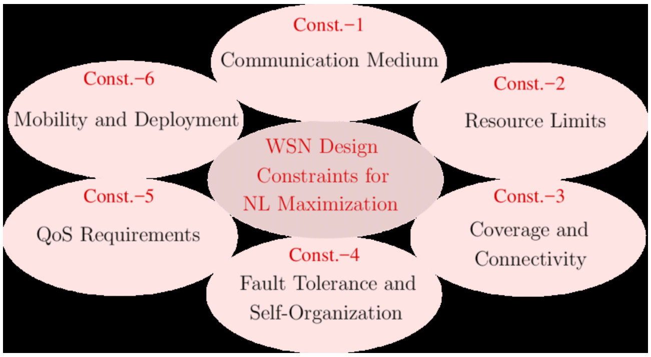
INTRODUCTION

Wireless sensor networks are quickly gaining popularity due to the fact that they are potentially low cost solutions to a variety of real-world challenges. Their low cost provides a means to deploy large sensor arrays in a variety of conditions capable of performing both military and civilian tasks. But sensor networks also introduce severe resource constraints due to their lack of data storage and power. Both of these represent major obstacles to the implementation of traditional computer security techniques in a wireless sensor network. The unreliable communication channel and unattended operation make the security defenses even harder. Indeed, as pointed out in, wireless sensors often have the processing characteristics of machines that are decades old (or longer), and the industrial trend is to reduce the cost of wireless sensors while maintaining similar computing power. With that in mind, many researchers have begun to address the challenges of maximizing the processing capabilities and energy reserves of wireless sensor nodes while also securing them against attackers.



5.1 CONSTRAINTS IN WIRELESS SENSOR NETWORKS

A WSN consists of a large number of sensor nodes that are inherently resource-constrained devices. These nodes have limited processing capability, very low storage capacity, and constrained communication bandwidth. These constraints are due to limited energy and physical size of the sensor nodes. Due to these constraints, it is difficult to directly employ the conventional security mechanisms in WSNs. In order to optimize the conventional security algorithms for WSNs, it is necessary to be aware about the constraints of sensor nodes.

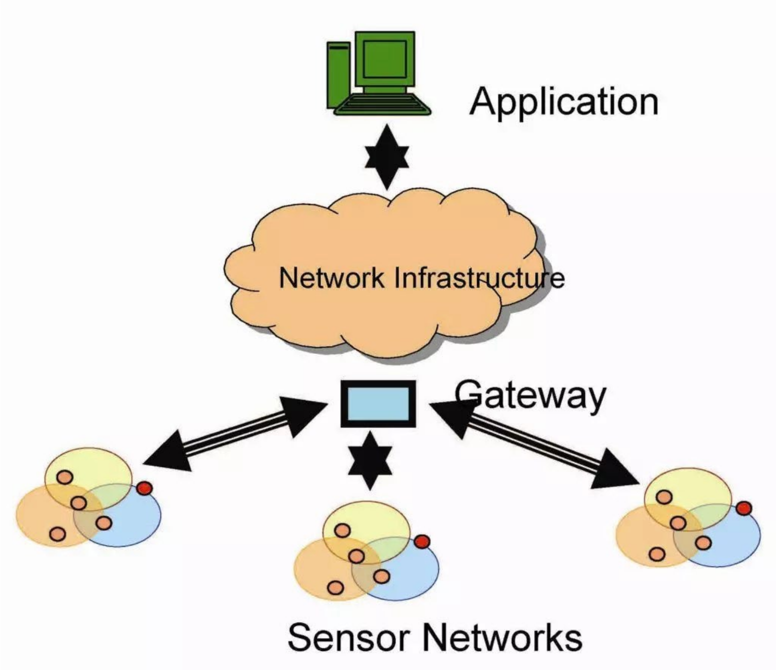


Some of the major constraints of a WSN are listed below.

Energy constraints: Energy is the biggest constraint for a WSN. In general, energy consumption in sensor nodes can be categorized in three parts: (i) energy for the sensor transducer, (ii) energy for communication among sensor nodes, and (iii) energy for microprocessor computation. The study in found that each bit transmitted in WSNs consumes about as much power as executing 800 to 1000 instructions. Thus, communication is more costly than computation in WSNs. Any message expansion caused by security mechanisms comes at a significant cost. Further, higher security levels in WSNs usually correspond to more energy consumption for cryptographic functions. Thus, WSNs could be divided into different security levels depending on energy cost.

5.2 SECURITY REQUIREMENTS IN WIRELESS SENSOR NETWORKS

Following are the Security Requirements in Wireless Sensor Networks.

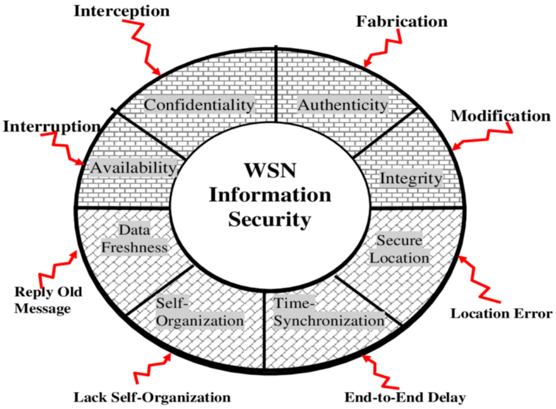


5.2.1 Confidentiality

Confidentiality requirement is needed to ensure that sensitive information is well protected and not revealed to unauthorized third parties.

The confidentiality objective is required in sensors environment to protect information traveling between the sensor nodes of the network or between the sensors and the base station from disclosure, since an adversary having the appropriate equipment may eavesdrop on the communication. By eavesdropping, the adversary could overhear critical information such as sensing data and routing information. Based on the sensitivity of the data stolen, an adversary may cause severe damage since he can use the sensing data for many illegal purposes i.e. sabotage, blackmail. For example, competitors may use the data to produce a better product i.e. safety monitoring sensor application. Furthermore, by stealing routing information the adversary could introduce his own malicious nodes into the network in an attempt to overhear the entire communication.

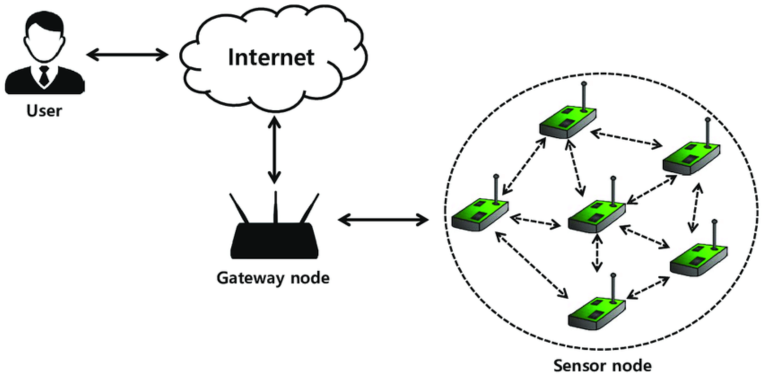
If we consider eavesdropping to be a network level threat, then a local level threat could be a compromised node that an adversary has in his possession. Compromised nodes are a big threat to confidentiality objective since the adversary could steal critical data stored on nodes such as cryptographic keys that are used to encrypt the communication.



5.2.2 Authentication

As in conventional systems, authentication techniques verify the identity of the participants in a communication, distinguishing in this way legitimate users from intruders.

In the case of sensor networks, it is essential for each sensor node and base station to have the ability to verify that the data received was really send by a trusted sender and not by an adversary that tricked legitimate nodes into accepting false data. If such a case happens and false data are supplied into the network, then the behavior of the network could not be predicted and most of times will not outcome as expected.

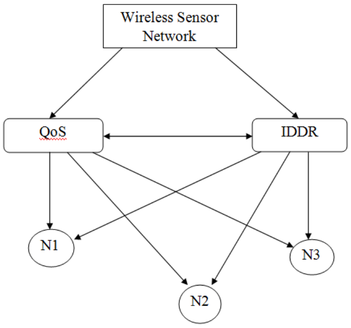


Authentication objective is essential to be achieved when clustering of nodes is performed. clustering involves grouping nodes based on some attribute such as their location, sensing data etc. and that each cluster usually has a cluster head that is the node that joins its cluster with the rest of the sensor network (meaning that the communication among different clusters is performed through the cluster heads). In these cases, where clustering is required, there are two authentication situations which should be investigated; first it is critical to ensure that the nodes contained in each cluster will exchange data only with the authorized nodes contained and which are trusted by the specified cluster (based on some authentication protocol). Otherwise, if nodes within a cluster receive data from nodes that are not trusted within the current community of nodes and further process it, then the expected data from that cluster will be based on false data and may cause damage. The second authentication situation involves the communication between the cluster heads of each cluster; communication must be established only with cluster heads that can prove their identity. No malicious node should be able to masquerade as a cluster head and communicate with a legitimate cluster head, sending it false data or either compromising exchanged data.

5.2.3 Integrity

Moving on to the integrity objective, there is the danger that information could be altered when exchanged over insecure networks. Lack of integrity could result in many problems since the consequences of using inaccurate information could be disastrous, for example for the healthcare sector where lives are endangered.

Integrity controls must be implemented to ensure that information will not be altered in any unexpected way. Many sensor applications such as pollution and healthcare monitoring rely on the integrity of the information to function with accurate outcomes; it is unacceptable to measure the magnitude of the pollution caused by chemicals waste and find out later on that the information provided was improperly altered by the factory that was located near by the monitored lake. Therefore, there is urgent need to make sure that information is traveling from one end to the other without being intercepted and modified in the process.

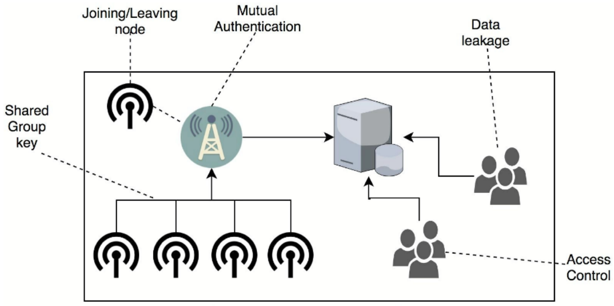


5.2.4 Freshness

One of the many attacks launched against sensor networks is the message replay attack where an adversary may capture messages exchanged between nodes and replay them later to cause confusion to the network. Data freshness objective ensures that messages are fresh, meaning that they obey in a message ordering and have not been reused. To achieve freshness, network protocols must be designed in a way to identify duplicate packets and discard them preventing potential mix-up.

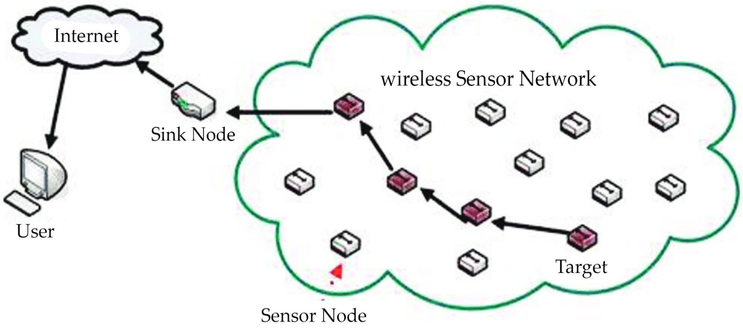
5.2.5 Secure Management

Management is required in every system that is constituted from multi components and handles sensitive information. In the case of sensor networks, we need secure management on base station level; since sensor nodes communication ends up at the base station, issues like key distribution to sensor nodes in order to establish encryption and routing information need secure management. Furthermore, clustering requires secure management as well, since each group of nodes may include a large number of nodes that need to be authenticated with each other and exchange data in a secure manner. In addition, clustering in each sensor network can change dynamically and rapidly. Therefore, secure protocols for group management are required for adding and removing members, and authenticating data from groups of nodes.



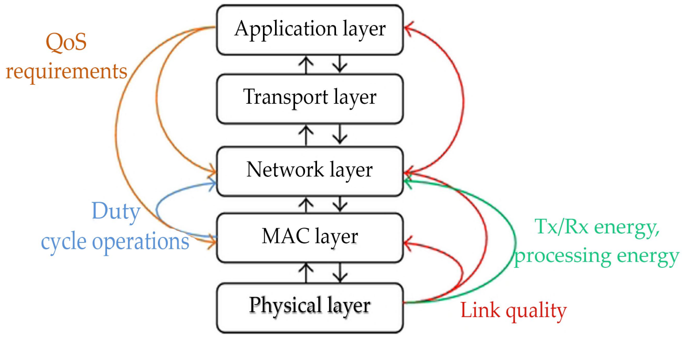
5.2.6 Availability

Availability ensures that services and information can be accessed at the time that they are required. In sensor networks there are many risks that could result in loss of availability such as sensor node capturing and denial of service attacks. Lack of availability may affect the operation of many critical real time applications like those in the healthcare sector that require a 24/7 operation that could even result in the loss of life. Therefore, it is critical to ensure resilience to attacks targeting the availability of the system and find ways to fill in the gap created by the capturing or disablement of a specific node by assigning its duties to some other nodes in the network.



5.2.7 Quality of Service

Quality of Service objective is a big headache to security. And when we are speaking about sensor networks with all the limitations they have, quality of service becomes even more constrained. Security mechanisms must be lightweight so that the overhead caused for example by encryption must be minimized and not affect the performance of the network. Performance and quality in sensor networks involve the timely delivery of data to prevent for example propagation of pollution and the accuracy with which the data reported match what is actually occurring in their environment.



5.3 SECURITY VULNERABILITIES IN WIRELESS SENSOR NETWORKS

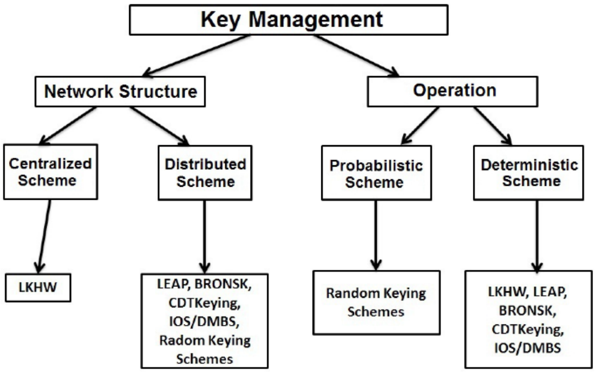
Despite the fact that the WSN offers a lot, the security challenges must be discerned and tackled accordingly. Failure to do this timely and sufficiently may render it not quite useful to say the least Just like any kind of network, WSN security seeks to achieve the following:

- Confidentiality: concealing message from unauthorized 'ears'
- Integrity: ensuring that message is not altered over the network
- Authenticity: ensuring the other party is who it claims to be
- Availability: ability to use the network resource

It should be noted that the way security is handled in WSN requires a lot more than what obtains in other kinds of network because WSN has its own peculiarities. Argues that existing security mechanisms are inadequate, and new ideas are needed because of the following reasons:

- Energy Limitation
- Deployment in an environment more open to physical attack
- Close interaction with physical environment and with people

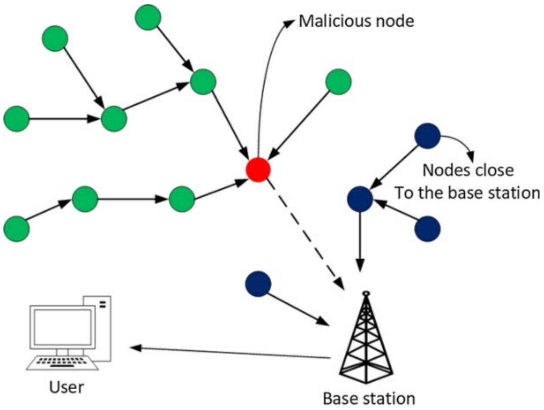
Therefore, because of its peculiar nature, the WSN must be secured with more than the traditional computer network security techniques. The attack scenarios or security vulnerabilities and their mitigation are discussed next.



5.3.1 Denial of service attacks

Denial of Service (DoS) attack is an attempt to make a network resource unavailable for its legitimate users. The Sensor node may get rogue broadcast of unrelenting high energy messages. This broadcast interferes with the radio frequency of the WSN thereby causing what is called jamming. Given this situation, the WSN will be negatively affected in terms of giving services to the legitimate users of the WSN. DoS could also occur at the data link layer where the medium access control (MAC) protocol of IEEE 802.11 gets violated. For an instance, a sensor node could be made to continuously send a request-to-send signal. Collision ensues, thereby forcing retransmission of colliding packets. Depending on the level of collision the attacker can succeed in making the sensor's power supply depleted

A spread spectrum can be used to tackle jamming of signals. Spread spectrum is the technique of using more bandwidth than the original message without losing the signal. This will prevent jamming. Collisions on the other hand can be stopped by using error correcting codes (ECC). Pathan however argues that ECC incurs more processing and communication overheads



5.3.2 Data Aggregation Attack

Depending on the WSN architecture, data may be aggregated in order to reduce the amount of data transmitted to the base station. For an instance, the average (instead of individually sensed) temperature of a certain geographical region could be taken and sent to the base station. According to the data aggregation node (also called cluster head) may be attacked through:

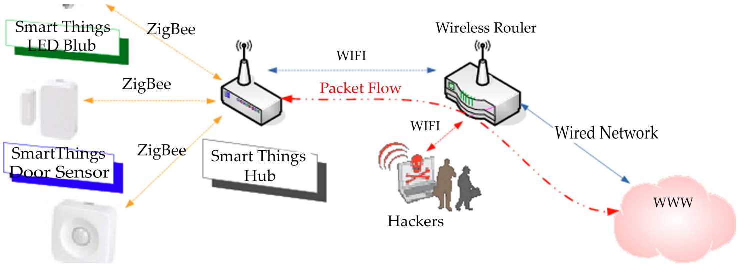
- Compromising a node physically to affect aggregated results
- Attacking aggregator nodes using different attacks (e.g. DoS)
- Sending false information to affect the aggregation results.

Tackling Data Aggregation Attacks will require Data encryption to be used. Voting technique can also be used. In this scheme the aggregator consults its witness before sending to the BS. The witness, upon approval, sends their MAC. This is costly to implement. In a Secure-Enhanced Data Aggregation based on Elliptic Curve Cryptography (SEDA-ECC) is proposed for WSNs. Here, the aggregation tree is divided into three subtrees. Also, three aggregated results are generated by performing Privacy Homomorphic-based aggregations in the three subtrees, respectively, to enable the base station (BS) verify the subtree aggregated results by comparing the aggregated count value.

5.3.3 Traffic Analysis Attack

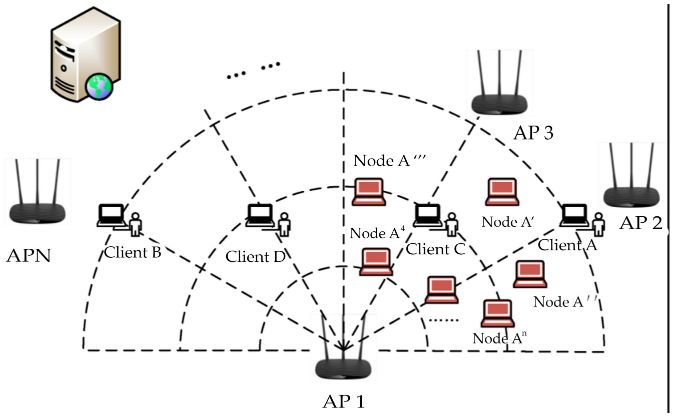
This kind of attack occurs when the attacker is able to gather information about the network topology. The important nodes (e.g. gateway) and base stations are identified by studying the traffic pattern. This can be rate monitoring or time correlated. The rate monitoring attacker tries to move towards the nodes that have a higher rate of packet sending. The assumption is that nodes close to the base station tend to forward more packets than those farther away from the base station. In time correlation attack, the path to base station is deduced by observing the correlation between neighbor nodes sending time to the base station.

Tackling Data Traffic Analysis Attack will require Sensor identities and public keys encryption to be used. Anonymity mechanisms can be used to check traffic analysis. One of such mechanisms is decentralizing sensitive data by using spanning tree such that no single node holds a complete view of the original data. Random forwarding of packets to non-parent nodes can check rate monitoring attack while fractal propagation can tackle time correlated attack. In fractal propagation, a node generates a fake packet when its neighbor is sending packet to the BS. The fake packet is sent randomly to another neighbor thus confusing the attacker of who is the BS.



5.3.4 Sybil Attack

Sybil attack happens when a device in WSN presents itself to the network with multiple identities that are all false. Through this spoof, the device can impersonate legitimate devices on the network. This situation is capable of deceiving devices on the network into accepting the impersonating device as a neighbor and as such, they forward their traffic to the trickster device as shown in Figure 1. This may corrupt the routing table.



Radio resource testing (RST) is a technique that can be used to tackle Sybil attacks. It has a node assigning each of its neighbours a different channel on which to communicate. The node then randomly chooses a channel and listens. If the node detects a transmission on the channel it is assumed that the node transmitting on the channel is a physical node. Random Key Predistribution (RKP) is another method that can mitigate Sybil attacks. Here nodes are assigned a random set of keys to enable them communicate with their neighbour. Because of this, if a node randomly generates identities, it will not possess enough keys to take on multiple identities and thus will be unable to exchange messages on the network due to the fact that the invalid identity will be unable to encrypt or decrypt messages. In the Random Password Algorithm (RPA) is proposed. Here a routing table stores each node's id, the time and a password. The node's information is then compared with the table. Where there is a match the node is considered to be a normal node otherwise, a Sybil node. A further attempt to tackle Sybil attacks was proposed by. They proposed a Grid Based Transitory Master Key (GBTMK) scheme where the base station of the WSN is not engaged in key establishment and each node maintains a list of its authenticated neighbours that help to prevent the Sybil attack.

5.3.5 Eavesdropping

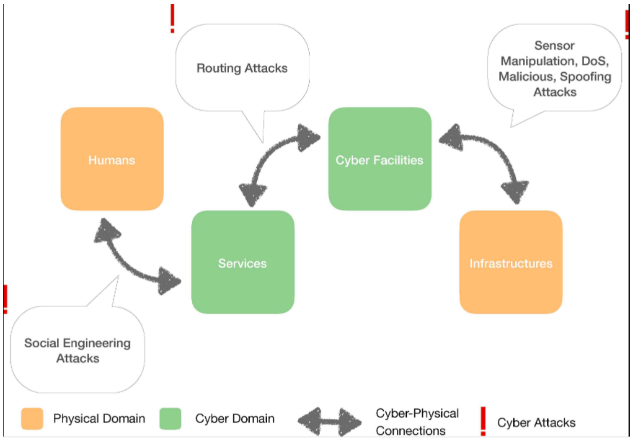
This occurs when an attacker snoops on the transmitted signal and secretly overhears what was supposed to be a private conversation over a confidential channel, in an unauthorized way, thereby compromising the confidentiality of the network. In the process of eavesdropping, some information could be gathered which the attacker could use to launch other forms of attack on a WSN. Such information includes user credentials, MAC address and cryptographic information.

Encryption can be used to check Eavesdropping proposed that using directional antennas to radiate radio signals on desired directions can potentially reduce the possibility of the eavesdropping attacks.

5.3.6 Routing Attacks

A number of attacks fall under routing attacks. The following are some of them

i. Blackhole attack: A node, usually malicious, drops packets received from its neighbor thereby making packets not to get to its destination as illustrated in Figure 2.


Routing Attacks
Sensor Manipulation, DoS, Malicious, Spoofing Attacks
Humans
Cyber Facilities
Services
Infrastructures
Social Engineering Attacks
Physical Domain
Cyber Domain
Cyber-Physical Connections
Cyber Attacks

ii. Selective forwarding attack: Here a malicious node selectively drops packets that match certain criteria and forwards the rest as shown in Figure 3.

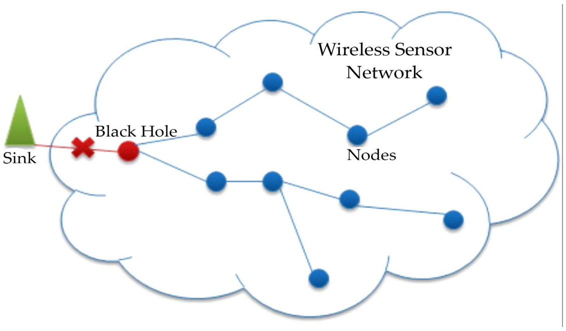


Figure 3: Selective Forwarding attack.

iii. Wormhole attack: In this case, the attacker deceives devices in the network by creating paths which appear to be the best. This approach can be used to unleash other attacks such as black hole as shown in Figure 4.

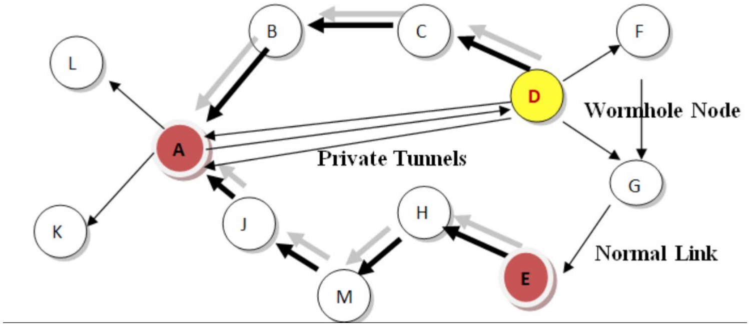


Figure 4: Wormhole Attack.

iv Sinkhole attack: Figure 5 illustrate this form of attack. As much traffic as possible is drawn to the attacking node and in most cases, the base station is cut off from receiving data from nodes.

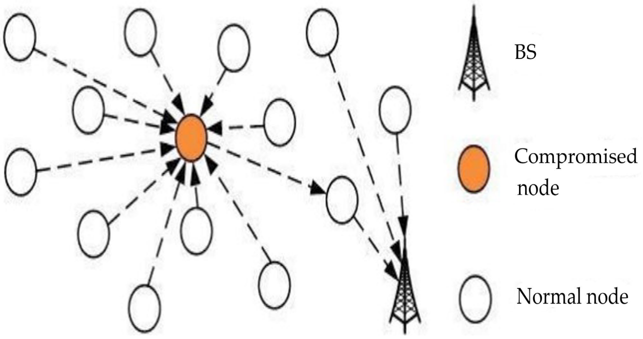


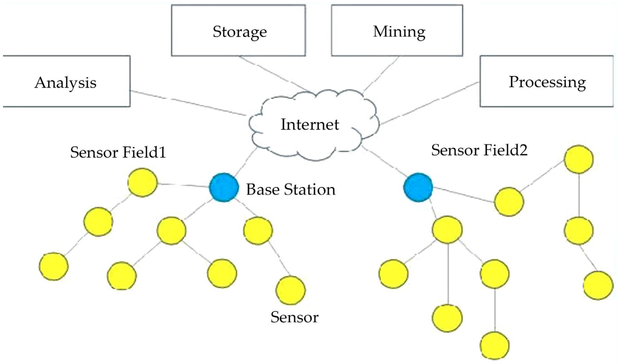
Figure 5: Sinkhole Attack.

Routing attacks are generally handled through key management and secure routing schemes. Although costly to implement on WSN due to the nature of sensors, however argues that the introduction of a trust model can create a balance to reduce this cost.

5.4 SECURITY MECHANISMS FOR WIRELESS SENSOR NETWORKS

Recent advances in electronic and computer technologies have paved the way for the proliferation of wireless sensor networks. Sensor networks usually consist of a large number of ultra-small autonomous devices. Each device, called a sensor node, is battery powered and equipped with integrated sensors, data processing capabilities, and short-range radio communications. In typical application scenarios, sensor nodes are spread randomly over the deployment region under scrutiny and collect sensor data. Wireless sensor networks are being deployed for a wide variety of applications, including military sensing and tracking, environment monitoring, patient monitoring and tracking, smart environments, etc. When sensor networks are deployed in a hostile environment, security becomes extremely important, as they are prone to different types of malicious attacks. For example, an adversary can easily listen to the traffic, impersonate one of the network nodes, or intentionally provide misleading information to other nodes. To provide safe data, communication should adopt security mechanisms.

Wireless sensor network distinguishes itself from other traditional wireless networks by relying on extremely constrained resources like energy, bandwidth and capabilities of processing and storing data. Traditional security techniques used in traditional networks cannot be applied directly, and new ideas are need.



5.4.1 Security Threats and Analysis

Threats

Wireless networks, in general, are more vulnerable to security attacks than wired networks, due to the broadcast nature of the transmission medium. Furthermore, wireless sensor networks have an additional vulnerability because nodes are often placed in a hostile or dangerous environment where they are not physically protected. For data sent through the network, the main security threats are as follows:

- Insertion of malicious code is the most dangerous attack that can occur. Malicious code injected in the network could spread to all nodes, potentially destroying the whole network, or even worse, taking over the network on behalf of an adversary. A seized sensor network can either send false observations about the environment to a legitimate user or send observations about the monitored area to a malicious user.
- Interception of the messages containing the physical locations of sensor nodes allows an attacker to locate the nodes and destroy them. The significance of hiding the location information from an attacker lies in the fact that the sensor nodes have small dimensions and their location cannot be trivially traced. Thus, it is important to hide the locations of the nodes. In the case of static nodes, the location information does not age and must be protected through the lifetime of the network.
- Besides the locations of sensor nodes, an adversary can observe the application specific content of messages including message IDs, timestamps and other fields. Confidentiality of those fields in the application is less important than confidentiality of location information, because the application specific data does not contain sensitive information, and the lifetime of such data is significantly shorter.
- An adversary can inject false messages that give incorrect information about the environment to the user. Such messages also consume the scarce energy resources of the nodes. This type of attack is called sleep deprivation torture in.

Analysis

The major security concerns in wireless sensor networks and their corresponding requirements.

Confidentiality: Unauthorized parties should not be able to infer the content of messages. Due to the shared wireless medium, the adversary can eavesdrop on the messages exchanged between sensor nodes. To prevent the release of message content to eavesdroppers, efficient cryptographies can be used for message encryption before transmissions.

Integrity: The receiver should be able to detect any modifications to a received message during its transmission. This prevents, for example, man-in-themiddle attacks where an adversary overhears, alters, and re-broadcasts messages. By including message authentication codes (MAC), a cryptographically strong un-forgeable hash, with the packet, the packet integrity can be protected. Using a secret key for code generation, unauthenticated nodes will not be able to alter the content of legitimate messages in the network.

Authentication: Message authentication is important for many applications in sensor networks. Within the building sensor network, authentication is necessary for many administrative tasks (e.g. network reprogramming or controlling sensor node duty cycle). At the same time, an adversary can easily inject messages, so the receiver needs to make sure that the data used in any decision-making process originates from the correct source. Informally, data authentication allows a receiver to verify that the data really was sent by the claimed sender. In the two-party communication case, data authentication can be achieved through a purely symmetric mechanism: The sender and the receiver share a secret key to compute a message authentication code (MAC) of all communicated data. When a message with a correct MAC arrives, the receiver knows that it must have been sent by the sender.

Access Control: Unauthorized nodes should not be able to participate in the network by either acting as a router or injecting new traffic. By including message authentication code (MAC) with the packet, unauthenticated nodes will not be able to send legitimate messages into the network.

Semantic security: Semantic security ensures that an eavesdropping adversary cannot obtain information about the plaintext, even if it sees multiple encryptions of the same message. The lack of semantic security makes traffic analysis easy. One common method of achieving this in symmetric block cipher is to use an Initial Value in the encryption function; this value may be a random value sent with the message or kept implicitly by both parties as a counter or the clock value.

Message replay protection: Even if messages are cryptographically protected so that their contents cannot be inferred or forged, an attacker would be able to capture valid messages and replay them later. Thus, independence on what mechanism is selected to secure the messages, that mechanism must be protected against replay attacks. Replay protection guarantees the system is immune to the stale or falsely located information. Generally, replay attacks can be defeated at the price of network synchronization and additional communication overhead.

Freshness: Given that all sensor networks stream some forms of time varying measurements, it is not enough to guarantee confidentiality and authentication; we also must ensure each message is fresh. Informally, data freshness implies that the data is recent, and it ensures that no adversary replayed old messages. Two types of freshness are identified: weak freshness, which provides partial message ordering, but carries no delay information, and strong freshness, which provides a total order on a request response pair, and allows for delay estimation. Weak freshness is required by sensor measurements, while strong freshness is useful for time synchronization within the network.

5.4.2 Security Mechanisms

The security of wireless sensor networks has attracted a lot of attention in the recent years. Many researchers have proposed some security mechanisms. We primarily introduce several ones.

Localized Encryption and Authentication Protocol (LEAP)

LEAP provides multiple keying mechanisms that can be used for providing confidentiality and authentication in sensor networks. It supports the establishment of four types of keys for each sensor node – an individual key shared with the base station, a pairwise key shared with another sensor node, a cluster key shared with multiple neighboring nodes, and a group key that is shared by all the nodes in the network. Now each of these keys is discussed and established in the LEAP protocol.

Type of Key

Individual Key: Every node has a unique key that it shares pairwise with the base station. This key is used for secure communication between a node and the base station. For example, a node may send an alert to the base station if it observes any abnormal or unexpected behavior by a neighboring node. Similarly, the base station can use this key to encrypt any sensitive information, e.g. keying material or special instruction, it sends to an individual node.

Cluster Key: A cluster key is a key shared by a node and all its neighbors, and it is mainly used for securing locally broadcast messages, e.g. routing control information, or securing sensor messages which can benefit from passive participation. Researchers have shown that in-network processing techniques, including data aggregation and passive participation are very important for saving energy consumption in sensor networks. For example, a node which overhears a neighboring sensor node transmitting the same reading as its own current reading can elect not to transmit the same. In responding to aggregation operations such as MAX, a node can also suppress its own reading if its reading is not larger than an overheard one. For passive participation to be feasible, neighboring nodes should be able to decrypt and authenticate some classes of messages, e.g. sensor readings, transmitted by their neighbors. This means that such messages should be encrypted or authenticated by a locally shared key. Therefore, in LEAP each node possesses a unique cluster key that it uses for securing its messages, while its immediate neighbors use the same key for decryption or authentication of its messages.

Pairwise Shared Key: Every node shares a pairwise key with each of its immediate neighbors. In LEAP, pairwise keys are used for securing communications that require privacy or source authentication. For example, a node can use its pairwise keys to secure the distribution of its cluster key to its neighbors, or to secure the transmissions of its sensor readings to an aggregation node. Note that the use of pairwise keys precludes passive participation.

Group Key: This is a globally shared key that is used by the base station for encrypting messages that are broadcast to the whole group. For example, the base station issues missions, sends queries and interests. Note that from the confidentiality point of view there is no advantage to separately encrypting a broadcast message using the individual key of each node. However, since the group key is shared among all the nodes in the network, an efficient re-keying mechanism is necessary for updating this key after a compromised node is revoked.

Key Establishment

Individual Keys: Every node has an individual key that is only shared with the base station. This key is generated and pre-loaded into each node prior to its deployment. The individual key k_u^m for a node u (each node has a unique ID) is generated as follows:$k_u^m = f_{k_s^m}(u)$ Here f is a pseudo-random function and k_s^m is a master key known only to the controller. In this scheme the controller might only keep its master key to save the storage for keeping all the individual keys. When it needs to communicate with an individual node u, it computes k_u^m on the fly. Due to the computational efficiency of pseudo random functions, the computational overhead is negligible.

Cluster Keys: The cluster key establishment phase follows the pairwise key establishment phase, and the process is very straightforward. Consider the case that node u wants to establish a cluster key with all its immediate neighbors v_1, v_2... vm. Node u first generates a random keyk_u^c , then encrypts this key with the pairwise key of each neighbor, and then transmits the encrypted key to each neighbor v_i.

Node v_i decrypts the key k_u^c and stores it in a table. When one of the neighbors is revoked, node u generates a new cluster key and transmits to all the remaining neighbors in the same way.

Pairwise Shared Key: A pairwise shared key belonging to a node refers to a key shared only between the node and one of its direct neighbors (i.e. one-hop neighbors). For nodes whose neighborhood relationships are predetermined (e.g. via physical installation), pairwise key establishment is simply done by preloading the sensor nodes with the corresponding keys. The protocol establishes pairwise keys for sensor nodes unaware of their neighbors until their deployment (e.g. via aerial scattering). The approach exploits the special property of sensor networks consisting of stationary nodes that the set of neighbors of a node is relatively static, and that a sensor node that is being added to the network will discover most of its neighbors at the time of its initial deployment. Second, it is that a sensor node deployed in a security critical environment must be designed to sustain possible break-in attacks at least for a short interval (say several seconds) when captured by the adversary; otherwise, the adversary could easily compromise all the sensor nodes in a sensor network and then take over the network.

Group Key: A group key is a key shared by all the nodes in the network, and it is necessary when the controller is distributing a secure message, e.g. a query on some event of interest or a confidential instruction, to all the nodes in the network. One way for the base station to distribute a message M securely to all the nodes is using hop-by-hop translation. Specifically, the base station encrypts M with its cluster key and then broadcasts the message. Each neighbor receiving the message decrypts it to obtain M, reencrypts M with its own cluster key, and then rebroadcasts the message. The process is repeated until all the nodes receive M. However, this approach has a major drawback, that is, each intermediate node needs to encrypt and decrypt the message, thus consuming a non-trivial amount of energy on computation. Therefore, using a group key for encrypting a broadcast message is preferable from the performance point of view. A simple way to bootstrap a group key for a sensor network is to pre-load every node with the group key. An important issue that arises immediately is the need to securely update this key when a compromised node is detected. In other words, the group key must be changed and distributed to all the remaining nodes in a secure, reliable and timely fashion. The naive approach in which the base station encrypts the updated group key using the individual key of each node and then sends the encrypted key to each node separately is not scalable because its communication and computational costs increase linearly with the size of the network. The protocol proposes an efficient key updating scheme based on cluster keys: authentic node revocation and secure key distribution.

Random Key Predistribution Schemes

The main phases for random key predistribution schemes are as follows:

Key predistribution phase: A centralized key server generates a large key pool offline. The procedure for offline key distribution is as follows:

- Assign a unique node identifier or key ring identifier to each sensor.
- Select m different keys for each sensor from the key pool to form a key ring.
- Load the key ring into the memory of the sensor.

Sensor deployment phase: The sensors are randomly picked and uniformly distributed in a large area. Typically, the number of neighbors of a sensor (n) is much smaller than the total number of deployed sensors (N).

Key discovery phase: During the key discovery phase, each sensor broadcasts its key identifiers in clear-text or uses private share-key discovery scheme to discover the keys shared with its neighbors. By comparing the possessed keys, a sensor can build the list of reachable nodes with which share keys and then broadcast its list. Using the lists received from neighbors, a sensor can build a key graph based on the key-share relations among neighbors.

Pairwise key establishment phase: If a sensor shares key(s) with a given neighbor, the shared key(s) can be used as their pairwise key(s). If a sensor does not share key(s) with a given neighbor, the sensor uses the key graph built during key discovery phase to find a key path (see Definition 2) to set up the pairwise key. The set of all neighbors of sensor i is represented by W_i. The definition of key graph is given as follows:

Definition 1 (key graph). A key graph maintained by node i is defined as $G_i = (V_i , E_i)$ where, the vertices set $Vi = \{j \mid j\ W_i \in \vee_j = i\}$, the edges set $Ei = \{e_{jk} \mid j, k\ W_i \in \wedge_j\ R\ k\ \}$, R is a relation defined between any pair of nodes j and k if they share required number of key(s) after the key discovery phase

Definition 2 (key path). A key path between node A and B is defined as a sequence of nodes A, N_1, N_2,. . ., N_i, B, such that, each pair of nodes (A, N_1), (N_1, N_2), . . ., (N_{i-1}, N_i), (N_i ,B) has required number of shared key(s) after the key discovery phase. The length of the key path is the number of pairs of nodes in it.

Purely Randoom Key Predistribution (P-RKP)

There are two characteristics of current P-RKP schemes. First, the m keys preinstalled in a sensor can also be installed in other sensors. That is, a key can be shared by more than one pair of sensors. Second, in most of current schemes, there is no relation between the set of preloaded keys and the sensor ID. A recent solution proposed by Pietro attempts to define this relation. However, the scheme is not scalable in that the size of the network is restricted by a function of number of preinstalled keys.

Structured Key Pool Random Key Predistribution (SK-RKP) Scheme

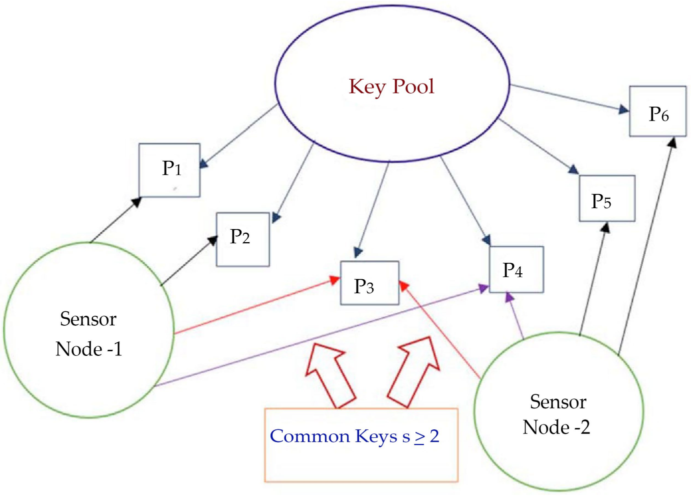
Unlike in P-RKP schemes, in SK-RKP scheme, each sensor is preloaded with a unique set of keys in its memory. The key discovery is not simply finding a shared key with the neighboring sensor, but using a set of polynomial variables (constructed by the keys possessed by the sensor) to derive the shared key. In addition, the key ID can serve as the sensor ID which is linked to the set of preinstalled keys. This link can prevent the attackers from misusing the sensors' IDs. In the following paragraphs, a brief description of structured key pool scheme is given. The SK-RKP scheme uses the key predistribution scheme proposed by Blom. This scheme allows any pair of nodes in a network to find a pairwise key in a secure way as long as no more than λ nodes are compromised. The scheme is built on two matrices: a publicly known matrix G of size $(\lambda + 1) \times N$; a secret matrix D of size $(\lambda + 1) \times (\lambda + 1)$ created by key distribution center. The matrix A of size $N \times (\lambda + 1)$ is then created as $A = (D \cdot G)^{T}$. Each row of A is the keys distributed to a group member and the row number can serve as a sensor's ID. Since $K = A \cdot G$ is a symmetric matrix, nodes i and j can generate a shared key (K_{ij} or K_{ji}) from their predistributed secrets, where K_{ij} is the element in K located in the ith row and jth column.

A key pool is constructed by many key spaces, represented by $A^{(t)}$, where t = 1, . . . , ω. Each sensor randomly selects τ key spaces out of ω key spaces, where $\tau < \omega$. If sensor k selects key space $A^{(t)}$, the kth row of $A^{(t)}$ and kth column of G are preinstalled in the sensor (note that the G matrix is unique). The SK-RKP scheme has following properties

- Once two nodes i and j have keys presinstalled from the same key space A(t), they can derive a shared key $K_{ij}^{(t)} = K_{ji}^{(t)}$
- If x rows of a key space $A^{(t)}$ are predistributed to x sensors and $x \leq \lambda$, any subset of the x sensors cannot collude to derive the secrets in other sensors.
- The ID of a sensor is represented by the row number of the key matrix A. No other sensor can impersonate this sensor, since the row of A is uniquely distributed to this sensor.

Security Levels Based on Different Data

The mechanism for communication security in wireless sensor networks is that data items must be protected to a degree consistent with their value. There are three types of data sent through the network: mobile code, locations of sensor nodes and application specific data. Following this categorization, the three security levels described here are based on private key cryptography utilizing group keys. Since all three types of data contain more or less confidential information, the content of all messages in the network is encrypted. The mechanism is assumed that all sensor nodes in the network are allowed to access the content of any message.



The deployment of security mechanisms in a sensor network creates additional overhead. Not only does latency increases due to the execution of the security related procedures, but also the consumed energy directly decreases the lifetime of the network. To minimize the security related costs, following the taxonomy of the types of data in the network, three security levels are defined:

- Security level I is reserved for mobile code, the most sensitive information sent through the network.
- Security level II is dedicated to the location information conveyed in messages.
- Security level III is applied to the application specific information.

The strength of the encryption for each of security levels corresponds to the sensitivity of the encrypted information.

Therefore, the encryption applied at level I is stronger than the encryption applied at level II, while the encryption on level II is stronger than the one applied at level III. Different security levels are implemented either by using various algorithms or by using the same algorithm with adjustable parameters that change its strength and corresponding computational overhead. Using one algorithm with adjustable parameters has the advantage of occupying less memory space. RC6 is selected. It is suitable for modification of its security strength because it has an adjustable parameter (number of rounds) that directly affects its strength. The overhead for the RC6 encryption algorithm increases with the strength of the encryption measured by the number of rounds.

Security Level I: The messages that contain mobile code are less frequent than the messages that the application instances on different nodes exchange. It allows us to use a strong encryption in spite of the resulting overhead. For information protected at this security level, nodes use the current master key. The set of master keys, the corresponding pseudorandom number generator, and a seed are credentials that a potential user must have in order to access the network. Once when the user obtains those credentials, she can insert any code into the network. If a malicious user breaks the encryption on this level using a "brute force" attack, she can insert harmful code into the network.

Security Level II: For data that contains locations of sensor nodes, a novel security mechanism is provided which isolates parts of the network, so that breach of security in one part of the network does not affect the rest of the network.

According to the applications expected to run in sensor networks, the locations of sensor nodes are likely to be included in the majority of messages. Thus, the overhead that corresponds to the encryption of the location information significantly influences the overall security overhead in the network. This must be taken into account when the strength of the encryption at this level is determined. Since the protection level is lower for the location information than for mobile code, the probability that the key for the level II can be broken is higher. Having the key, an adversary could potentially locate all nodes in the network. To constrain the damage to only one part of the network, the following security mechanism is proposed. Sensor nodes use location-based keys for level II encryption. The location-based keys enable separation between the regions where the location of nodes are compromised and the areas where nodes continue to operate safely.

The area covered by a sensor network is divided into cells. Nodes within one cell share a common location-based key, which is a function of a fixed location in the cell and the current master key. Between the cells, there is a bordering region whose width is equal to the transmission range. Nodes belonging to those regions have the keys for all adjacent cells. This ensures that two nodes within a transmission range from each other have a common key. The dimensions of the cells must be big enough so that the localized nature of the algorithms in the network ensures that the traffic among the cells is relatively low, compared to overall traffic. The areas can be of an arbitrary shape with the only requirement that the whole sensor terrain is covered. A division of the area in uniformly sized cells is the most appropriate solution, because it allows a fast and easy way for a node to determine its cell membership. The network is divided into hexagonal cells, since it ensures that the gateway nodes have at most three keys.

Security Level III: The application specific data use a weaker encryption than the one used for the two aforementioned types of data. The weaker encryption requires lower computational overhead for application specific data. Additionally, the high frequency of messages with application specific data prevents using stronger and resource consuming encryption. Therefore, an encryption algorithm that demands less computational resources with a corresponding decrease in the strength of security is adopted.

The key used for the encryption of the level III information is derived from the current master key. The MD5 hash function accepts the master key and generates a key for level III. Since the master key is periodically changed, the corresponding key at this level follows those changes.

In the discussion above the major assumptions of the all the proposed security schemes is that the sensor nodes are perfectly time synchronized and have exact knowledge of their location. It is not unrealistic that the nodes can be synchronized up to μs.

SUMMARY

- Wireless sensor networks are quickly gaining popularity due to the fact that they are potentially low cost solutions to a variety of real-world challenges. Their low cost provides a means to deploy large sensor arrays in a variety of conditions capable of performing both military and civilian tasks.
- A WSN consists of a large number of sensor nodes that are inherently resource-constrained devices. These nodes have limited processing capability, very low storage capacity, and constrained communication bandwidth. These constraints are due to limited energy and physical size of the sensor nodes.
- Authentication objective is essential to be achieved when clustering of nodes is performed. clustering involves grouping nodes based on some attribute such as their location, sensing data etc. and that each cluster usually has a cluster head that is the node that joins its cluster with the rest of the sensor network (meaning that the communication among different clusters is performed through the cluster heads).
- Management is required in every system that is constituted from multi components and handles sensitive information. In the case of sensor networks, we need secure management on base station level; since sensor nodes communication ends up at the base station, issues like key distribution to sensor nodes in order to establish encryption and routing information need secure management.
- Availability ensures that services and information can be accessed at the time that they are required. In sensor networks there are many risks that could result in loss of availability such as sensor node capturing and denial of service attacks.
- Radio resource testing (RST) is a technique that can be used to tackle Sybil attacks. It has a node assigning each of its neighbours a different channel on which to communicate. The node then randomly chooses a channel and listens. If the node detects a transmission on the channel it is assumed that the node transmitting on the channel is a physical node.

REFERENCES

1. Akyildiz, F., W. Su, Y. Sankarasubramaniam, and E. Cayirci. August 2002. "A Survey on Sensor Networks." IEEE Communications Magazine 40 (80): 102 – 114.

2. Deng, J., R. Han, and S. Mishra. November 2002. "INSENS: Intrusion-Tolerant Routing in Wireless Sensor Networks", Technical Report CU-CS-939-02, Department of Computer Science, University of Colorado at Boulder, November 2002.

3. Karp. B. and H. T. Kung. "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks." In Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MobiCom'00), 243-254, August 2000, Boston, Massachusetts, USA.

4. Parno, B., A. Perrig, and V. Gligor. May 2005. "Distributed Detection of Node Replication Attacks in Sensor Networks." In Proceedings of the IEEE Symposium on Security and Privacy (S&P'05), 49-63, Oakland, California, USA.

5. Gruteser, M., G. Schelle, A. Jain, R. Han, and D. Grunwald. May 2003. "Privacy-Aware Location Sensor Networks." In Proceedings of the 9th USENIX Workshop on Hot Topics in Operating Systems (HotOS IX), Vol 9, 28, Lihue, Hawaii, USA.

6. Malan, D. J., M. Welsh, and M.D. Smith. October 2004. "A Public-Key Infrastructure for Key Distribution in TinyOS based on Elliptic Curve Cryptography." In Proceedings of the 1st IEEE International Conference on Sensor and Ad Hoc Communications and Networks, Santa Clara, California, October, 2004.

7. Rivest, R. L., A. Shamir, and L. Adleman. 1983. "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems." Communications of the ACM, 26 (1): 96-99.

8. Brown, M., D. Cheung, D. Hankerson, J.L. Hernandez, M. Kirkup, and A. Menezes. August 2000. "PGP in Constrained Wireless Devices." In Proceedings of the 9th USENIX Security Symposium (SSYM'00), Vol 9, 19.

9. Gura, N., A. Patel, A. Wander, H. Eberle, and S. Shantz. August 2004. "Comparing Elliptic Curve Cryptography and RSA on 8-bit CPUs." In Proceedings of the 6th International Workshop on Cryptographic Hardware and Embedded Systems (CHES '04), 119-132, Cambridge, Massachusetts, USA., Springer LNCS Vol. 3156.

10. Elliptic Curve Cryptography, SECG Std. SEC1, 2000. http://www.secg.org/collateral/sec1.pdf. Accessed on July 11, 2012.

11. Kaliski, B. May 2003. TWIRL and RSA Key Size, RSA Laboratories, Technical Note.

12. Recommended Elliptic Curve Domain Parameters, SECG Std. SEC 2, 2000. http://www.secg.org/collateral/sec2_final.pdf. Accessed on July 11, 2012.

13. Hankerson, D., A.Menezes, and S. Vanstone. 2004. Guide to Elliptic Curve Cryptography, New York, Springer-Verlag.

14. Freier, A., P. Karlton, and P. Kocher. The SSL Protocol, version 3.0.. http://www.mozilla.org/projects/security/pki/nss/ssl/draft302.text. Accessed on July 11, 2012.

15. Watro, R., D. Kong, S. Cuti, C. Gardiner, C. Lynn, and P. Kruus. 2004. "TinyPK: Securing Sensor Networks with Public Key Technology." In Proceedings of the 2nd ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN'04), 59-64, New York: ACM Press.

16. Liu A., and P. Ning. April 2008. "TinyECC: A Configurable Library for Elliptic Curve Cryptography in Wireless Sensor Networks." In Proceedings of the 7th International Conference on Information Processing in Sensor Networks (IPSN'08), SPOTS Track, 245-256, St Louis, Missouri, USA. http://discovery.csc.ncsu.edu/software/TinyECC/. Accessed on July 11, 2012.

17. Du, W., J. Deng, Y.S. Han, S. Chen, and P. K. Varshney. 2004. "A Key Management Scheme for Wireless Sensor Networks Using Deployment Knowledge." In Proceedings of IEEE INFOCOM, 586-597, Hong Kong, China.

18. Hwang, D. D., B. Lai, and I. Verbauwhede. July 2004. "Energy-memory-security tradeoffs in distributed sensor networks, in Proceedings of the 3rd International Conference on Ad-hoc Networks and Wireless (ADHOC-NOW), 70-81. Springer LNCS, Vol. 3158.

19. Deng, J., R. Han, and S. Mishra. July 2005. "Security, Privacy, and Fault-Tolerance in Wireless Sensor Networks." In: Wireless Sensor Networks: A Systems Perspective, N. Bulusu and S. Jha (Eds.), Artech House.

20. Sen, J. 2010. "Reputation- and Trust-Based Systems for Wireless Self-Organizing Networks.",In: Security of Self-Organizing Networks: AMNET, WSN, WMN, VANET, edited by Al-Sakib Khan Pathan, Aurbach Publications, Book Chapter No. 5, 91- 124, CRC Press, Taylor & Francis Group, USA.

CHAPTER 6

TRANSPORT CONTROL PROTOCOLS FOR WIRELESS SENSOR NETWORKS

INTRODUCTION

Wireless sensor networks (WSNs) provide a powerful means to collect information on a wide variety of natural phenomena. WSNs typically consist of a cluster of densely deployed nodes communicating with a sink node which, in turn, communicates with the outside world. WSNs are constrained by low power, dense deployment, and limited processing power and memory. WSNs are composed of small, cheap, self-contained, and disposable sensor nodes. The unique constraints imposed by WSNs present unique challenges in the design of such networks.

The need for a transport layer to handle congestion and packet loss recovery in WSNs has been debated; the idea of a cheap, easily deployable network runs contrary to the costly, lengthy process of implementing a unique and specialized transport layer for a WSN. WSNs have advanced to the level of specialization where congestion control and reliability can be incorporated at each individual node.

Reliable data transmission in WNSs is difficult due to the following characteristics of WSNs:

- limited processing capabilities and transmission range of sensor nodes;
- close proximity to ground causes signal attenuation or channel fading which leads to asymmetric links;
- close proximity to ground and variable terrain also leads to shadowing which can effectively isolate nodes from the network;

- conservation of energy requires unused nodes and wake only when needed;
- dense deployment of sensor nodes creates significant channel contention and congestion.

The above characteristics can cause loss of data in WSNs. Fortunately, WSNs also provide unique features that can be leveraged to help mitigate losses and design energy-efficient transport layer protocols by network designers. For example,

- When the nature of the data allows, it can be aggregated at intermediate nodes.
- Network density, multiple paths to any given destination, and data aggregation in combination with a good choice of network layer can lessen some of the losses due to channel fading and shadowing.
- Some amount of loss can be made acceptable by employing data aggregation at the sensor nodes.
- Data aggregation may result in smaller packet size and consequently lower packet loss.
- Granularity of sensing an event can be controlled.
- Some events may require a very rough granularity.

Traditional transport layer protocols, such as TCP, are not suitable for severely resource constrained WSNs having characteristics which are different from traditional wired networks. The *objective* of this chapter is to illustrate the need for a standard transport layer in WSNs, outline future challenges involved in designing a transport layer protocol that fits the unique constraints imposed by WSNs, and present current implementations of transport layers for WSNs. This chapter gives out a survey on transport control protocol for wireless sensor networks (WSNs). First, it lists the disadvantages of traditional transport control protocols (TCP and UDP) for the environment of WSNs. Second, several design issues of transport control protocols for WSNs are presented. Third, some existing transport control protocols for WSNs are classified and compared. Finally, several problems needing further studying are outlined.

6.1 TCP

TCP stands for Transmission Control Protocol. It is a transport layer protocol that facilitates the transmission of packets from source to destination. It is a connection-oriented protocol that means it establishes the connection prior to the communication that occurs between the computing devices in a network. This protocol is used with an IP protocol, so together, they are referred to as a TCP/IP.

The main functionality of the TCP is to take the data from the application layer. Then it divides the data into a several packets, provides numbering to these packets, and finally transmits these packets to the destination. The TCP, on the other side, will reassemble the packets and transmits them to the application layer. As we know that TCP is a connection-oriented protocol, so the connection will remain established until the communication is not completed between the sender and the receiver.

6.1.1 Features of TCP Protocol

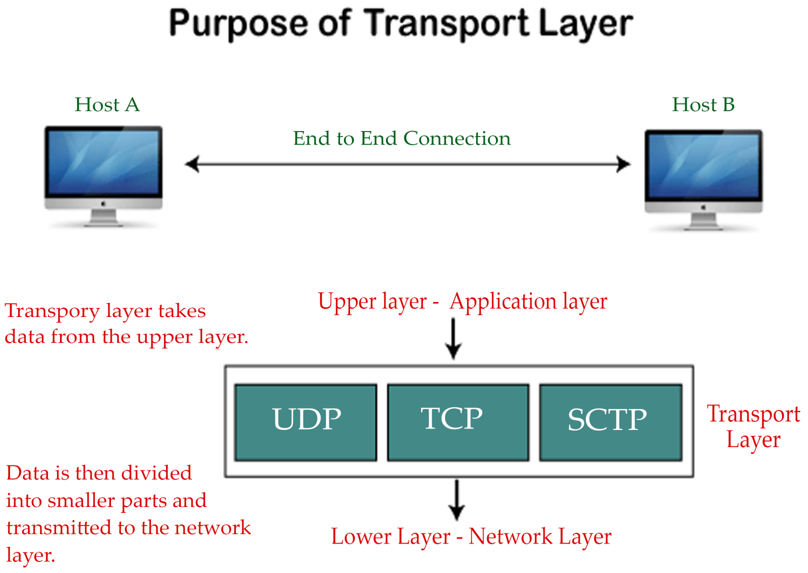
The following are the features of a TCP protocol:

- **Transport Layer Protocol:** TCP is a transport layer protocol as it is used in transmitting the data from the sender to the receiver.
- **Reliable:** TCP is a reliable protocol as it follows the flow and error control mechanism. It also supports the acknowledgment mechanism, which checks the state and sound arrival of the data. In the acknowledgment mechanism, the receiver sends either positive or negative acknowledgment to the sender so that the sender can get to know whether the data packet has been received or needs to resend.
- **Order of the data is maintained:** This protocol ensures that the data reaches the intended receiver in the same order in which it is sent. It orders and numbers each segment so that the TCP layer on the destination side can reassemble them based on their ordering.
- **Connection-oriented:** It is a connection-oriented service that means the data exchange occurs only after the connection establishment. When the data transfer is completed, then the connection will get terminated.
- **Full duplex:** It is a full-duplex means that the data can transfer in both directions at the same time.
- **Stream-oriented:** TCP is a stream-oriented protocol as it allows the sender to send the data in the form of a stream of bytes

and also allows the receiver to accept the data in the form of a stream of bytes. TCP creates an environment in which both the sender and receiver are connected by an imaginary tube known as a virtual circuit. This virtual circuit carries the stream of bytes across the internet.

6.1.2 Need of Transport Control Protocol

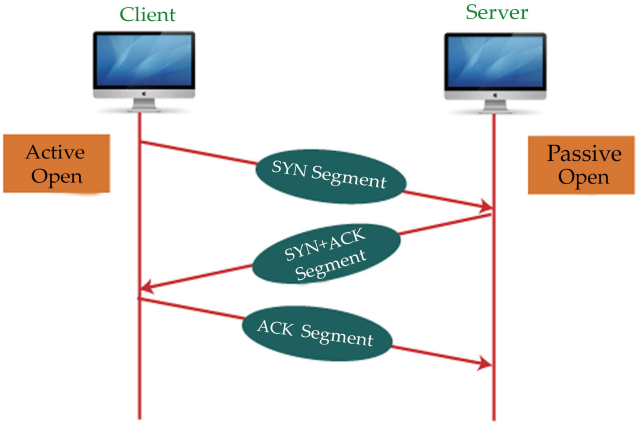
In the layered architecture of a network model, the whole task is divided into smaller tasks. Each task is assigned to a particular layer that processes the task. In the TCP/IP model, five layers are application layer, transport layer, network layer, data link layer, and physical layer. The transport layer has a critical role in providing end-to-end communication to the directly application processes. It creates 65,000 ports so that the multiple applications can be accessed at the same time. It takes the data from the upper layer, and it divides the data into smaller packets and then transmits them to the network layer.


Purpose of Transport Layer
Host A
Host B
End to End Connection
Transpory layer takes data from the upper layer.
Upper layer - Application layer
UDP
TCP
SCTP
Transport Layer
Data is then divided into smaller parts and transmitted to the network layer.
Lower Layer - Network Layer

6.1.3 Working of TCP

In TCP, the connection is established by using three-way handshaking. The client sends the segment with its sequence number. The server, in return, sends its segment with its own sequence number as well as the acknowledgement sequence, which is one more than the client sequence number. When the client receives the acknowledgment of its segment, then it sends the acknowledgment to the server. In this way, the connection is established between the client and the server.

Working of the TCP protocol



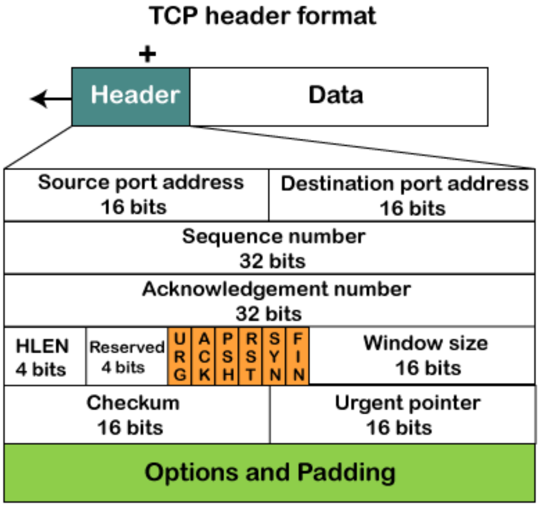
6.1.4 Advantages of TCP

- It provides a connection-oriented reliable service, which means that it guarantees the delivery of data packets. If the data packet is lost across the network, then the TCP will resend the lost packets.
- It provides a flow control mechanism using a sliding window protocol.
- It provides error detection by using checksum and error control by using Go Back or ARP protocol.
- It eliminates the congestion by using a network congestion avoidance algorithm that includes various schemes such as additive increase/multiplicative decrease (AIMD), slow start, and congestion window.

6.1.5 Disadvantage of TCP

It increases a large amount of overhead as each segment gets its own TCP header, so fragmentation by the router increases the overhead.

6.1.6 TCP Header Format



- **Source port:** It defines the port of the application, which is sending the data. So, this field contains the source port address, which is 16 bits.
- **Destination port:** It defines the port of the application on the receiving side. So, this field contains the destination port address, which is 16 bits.
- **Sequence number:** This field contains the sequence number of data bytes in a particular session.
- **Acknowledgment number:** When the ACK flag is set, then this contains the next sequence number of the data byte and works as an acknowledgment for the previous data received. For example, if the receiver receives the segment number 'x', then it responds 'x+1' as an acknowledgment number.
- **HLEN:** It specifies the length of the header indicated by the 4-byte words in the header. The size of the header lies between 20 and 60 bytes. Therefore, the value of this field would lie between 5 and 15.
- **Reserved:** It is a 4-bit field reserved for future use, and by default, all are set to zero.
- **Flags**

There are six control bits or flags:

- **URG:** It represents an urgent pointer. If it is set, then the data is processed urgently.
- **ACK:** If the ACK is set to 0, then it means that the data packet does not contain an acknowledgment.
- **PSH:** If this field is set, then it requests the receiving device to push the data to the receiving application without buffering it.
- **RST:** If it is set, then it requests to restart a connection.
- **SYN:** It is used to establish a connection between the hosts.
- **FIN:** It is used to release a connection, and no further data exchange will happen.

- **Window size:** It is a 16-bit field. It contains the size of data that the receiver can accept. This field is used for the flow control between the sender and receiver and also determines the amount of buffer allocated by the receiver for a segment. The value of this field is determined by the receiver.
- **Checksum:** It is a 16-bit field. This field is optional in UDP, but in the case of TCP/IP, this field is mandatory.
- **Urgent pointer:** It is a pointer that points to the urgent data byte if the URG flag is set to 1. It defines a value that will be added to the sequence number to get the sequence number of the last urgent byte.
- **Options:** It provides additional options. The optional field is represented in 32-bits. If this field contains the data less than 32-bit, then padding is required to obtain the remaining bits.

6.2 RELIABILITY IN WIRELESS SENSOR NETWORKS

Traffic from many applications in WSNs is considered loss tolerant. Loss tolerance in WSNs is due to the dense deployment of sensor nodes and data aggregation properties, giving rise to directional reliability. The design of WSN transport layer protocols should exploit directional reliability to lower the number of transmissions, especially for sensors that are close together and are expected to generate highly correlated data, and decrease the computational overhead by lowering the amount of data to be aggregated.

Some transport layer protocols only offer unidirectional reliable message delivery, where the idea of directional reliability is especially important. In the rest of this section, we discuss the following three types of reliability in a WSN:

- Point-to-point – Communication between sink and a remote host,
- Point-to-multipoint – Communication between sink and sensor nodes,
- Multipoint-to-point – Communication between sink and multiple wireless sensors.

6.2.1 Point-to-Point Reliability

The transport connection between the sink and a remote host uses a traditional TCP/IP transport layer. Sinks may either be robust nodes on a network with continual power and much more computational power than sensor nodes, or they may be a more robust version of a sensor node. In the latter case, a lightweight TCP/IP protocol, may be beneficial to these types of sink/proxy nodes.

6.2.2 Point-to-Multipoint Reliability

Messages originating at the sink may be queries and control messages, such as those related to congestion control and reprogramming the sensor nodes. These messages generally need to be delivered to sensor nodes with a higher degree of reliability than those originating at source sensor nodes. Loss of these messages could be detrimental to the life of the sensor network.

6.2.3 Multipoint-to-Point Reliability

Sensor nodes may process information received from other sensor nodes about an observed phenomenon. This process is called data aggregation and allows nodes to reduce the amount of information that must be forwarded. Data aggregation can reduce the impact of data loss by providing an averaged or smoothed value. Consequently, we may not be able to sense the phenomenon with fine granularity, but the impact of loss is reduced by sensing phenomena at a coarse level.

Even though sensor networks are fault tolerant we still have to guarantee the quality of the received data, i.e. the gathered data should be representative of the region queried, or event sensed. Collecting data tainted by packet loss can be more dangerous than not collecting any data at all. For example, if the sink queries the WSN and receives no response, we can assume we have experienced loss after some interval, but if we receive misleading or skewed data we have no way to verify that the data should be discarded at the sink. Figure 1 illustrates this idea. In Fig. 1 (a), the message never reaches the sink, we do not have the data, but we do not have corrupt data. After some interval, the sink may realize that no data has been received and resend the request.

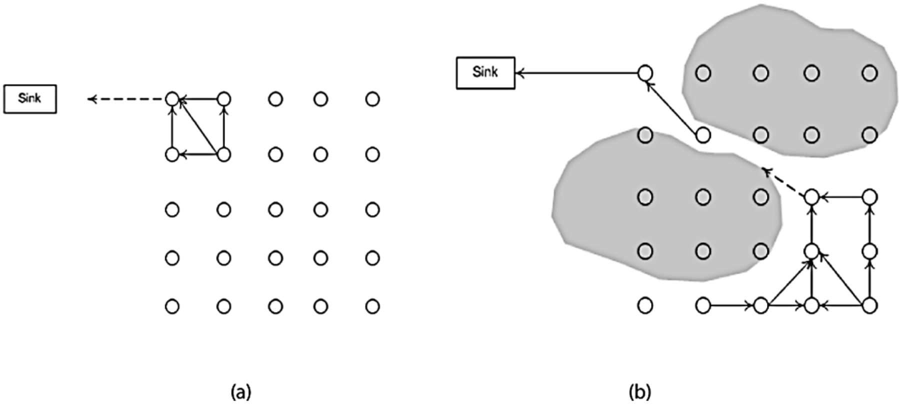


Figure 1. Sensor network loss combined with data aggregation could cause data to be skewed in certain situations.

Figure 1 (b) illustrates, a worse case scenario for loss with data aggregation. The gray areas indicate nodes that are unreachable. The aggregated response of many sensor nodes could be dropped, and data is forwarded from a sensor further from the event source. If the node is sufficiently removed from the event center, the data may not accurately reflect the event. In these cases it would be desirable to have a measure of the “goodness” of the data sent to the sink.

In this case, the “goodness” of the data becomes a new measure of the reliability of the data. The accuracy or granularity that is acceptable for the event varies between applications. ESRT is a proposed transport layer protocol for WSNs that allows control over the level of granularity with which the event is detected.

6.3 TRANSPORT PROTOCOLS FOR SENSOR NETWORKS

In addition to energy-efficient transport layer protocols in resource constrained WSNs, the protocol should also support

- reliable message delivery,
- congestion control, and
- energy efficiency.

The need for a transport layer protocol in WSNs has been debated. Some have suggested that (a) loss detection and recovery can be handled below the transport layer and mitigated using data aggregation, and (b) congestion is not an issue because sensor nodes spend most of the time sleeping resulting in sparse traffic in the network.

In contrast to the above arguments against the need for a transport layer protocol, Yarvis et al. and Dunkels et al. have shown that the generally dense deployment of sensor nodes give rise to congestion in a WSN. Data from sensor nodes to sink (multipoint-to-point) may suffer from channel contention; in the absence of congestion control, the ability of the sensor nodes to deliver data to the sink decreases.

Wan et al. and Stan et al. demonstrated scenarios where data must be delivered reliably in WSNs. In such cases, it is not sufficient to rely only on loss detection and reliability techniques at layers below the transport layer, since layers beneath the transport layer do not provide guaranteed end-to-end reliability.

The need for reliable message delivery and congestion control suggest that WSNs should have a transport layer, just as 802.3 and 802.11 networks need a transport layer. However, WSNs add a new constraint—energy efficiency. To prolong the lifetime of a WSN, an ideal transport layer needs to support reliable message delivery and provide congestion control in the most energy efficient manner possible. In the rest of this section, we discuss a number of transport layer protocols, including those which have been suggested for WSN.

6.3.1 TCP/IP

TCP/IP has been used successfully in wired 802.3 and wireless 802.11 networks and has been discussed as a possible transport layer for WSN. Certain attributes, such as IP addressing for individual nodes, unnecessary header overhead for data segments, no support for data centric routing, a heavyweight protocol stack, and an end-to-end reliability scheme that attributes segment losses network congestion, of TCP/IP; however, they make it unsuitable for use in WSNs without modification. Even if TCP/IP is not entirely suitable for WSNs, it is informative to compare TCP/IP to transport protocols designed specifically for WSNs. Such a comparison helps to illustrate that WSNs operate in a different paradigm, and thus need specially designed transport layers to meet their unique needs.

TCP/IP may not be suitable for standard sensor nodes in a WSN, but may still be used at the sink to communicate with other remote endpoints. Sensor nodes with high robustness, such as Crossbow, may use TCP/IP as a virtual sink or proxy between the WSN and the remote host to reduce the number of retransmissions of a data segment by less powerful sensor nodes.

Loss Detection/Recovery

TCP/IP, by default, uses an ACK-based end-to-end reliability mechanism; however, an end-to-end reliability mechanism is not appropriate for sensor networks, given their high loss rates due to signal attenuation and path loss arising from low power radios and channel contention from dense sensor deployment. The probability of receiving an errored packet increases exponentially with the increase in the number of hops on a WSN. To reduce this problem, Dunkels et al. suggest Distributed TCP Caching (DTC) which allows intermediate nodes to cache data segments; on detection of loss, the lost packets can be distributed to nodes using local retransmissions.

DTC requires intermediate nodes to cache intermediate segments. In a worst case scenario, when none of the surrounding nodes have the required segment cached, DTC degrades to end-to-end recovery (see Fig. 2). To help mitigate this problem, a sensor node caches the highest segment number it has seen. Although this improves the chances of a local neighbor having the required segment, it does not eliminate the possibility of DTC degrading to end-to-end recovery.

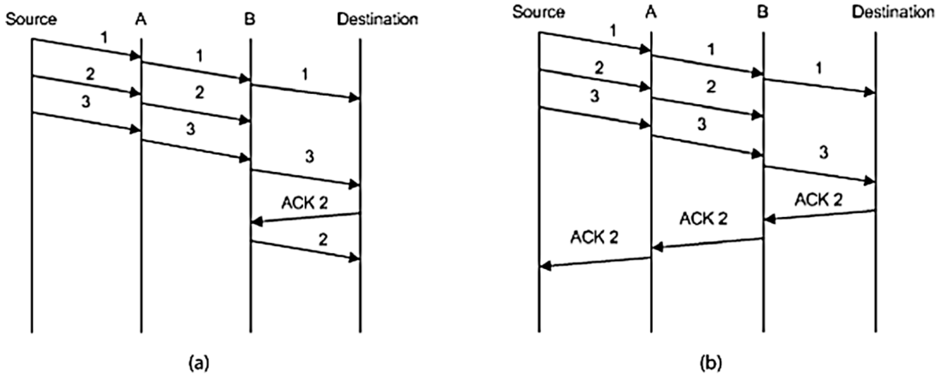


Figure 2. DTC caching performs aggressive hop-by-hop recovery when loss is detected; however, if the lost packet has been removed from cache, the NACK must be forwarded on potentially to the destination.

Congestion Control

No modification of the congestion control mechanism has been suggested by Dunkels et al. However, DTC should localize the reduction in transmission rates when segments can be recovered form neighboring sensor nodes.

Although the overhead needed to run TCP/IP seems prohibitive for a WSN, it may still be desirable to use TCP/IP for certain types of sensor nodes, specifically those which are less resource-restrained.

6.3.2 Pump Slowly, Fetch Quickly (PSFQ)

Pump Slowly Fetch Quickly (PSFQ) is a transport layer protocol, designed specifically to meet the unique resource challenges presented by WSNs with a focus on point-to-multipoint reliability. Data is pumped slowly from a root node into the network. Sensor nodes that experience loss can recover data segments by fetching them quickly from their immediate neighbors on a hop-by-hop basis. To reduce signal overhead, nodes signal the loss of segments using negative acknowledgement, rather than acknowledging each received packet.

Table 1. Problems and proposed solutions to using TCP/IP on WSN

Problem	Description	Solution
IP addressing architecture	Sensor networks are dense networks with as many as 10 nodes per cubic meter. This combined with the limited memory available to sensor nodes makes traditional IP addressing impractical.	Use spatial IP addressing.
Header overhead	Communication is one of the most costly activities in a WSN. The transmission of large headers of TCP/IP requires lot of energy.	Use header compression.
No support for data centric routing	Routing in IP networks is based on the host and network address. Routing in sensor networks needs to be data centric.	Use an application overlay network.
Sensor Nodes are severely resource limited.	The TCP/IP stack is considered to be too heavyweight for sensors with limited capabilities. Sensor nodes with limited memory may not be able to support a TCP/IP implementation.	Dunkels et al. have shown that a TCP/IP stack can be implemented for 8-bit processors with only a few hundred bytes of memory.
TCP performance and energy inefficiency	End-to-end acknowledgement and retransmission scheme in TCP translate to unnecessary expense in networks with multiple hops and limited energy.	Implement an energy-efficient distributed mechanism for acknowledgements and retransmissions.

PSFQ is based on the assumption that a WSN will generate light traffic most of the time; thus, it is designed to avoid loss due to instability of the wireless medium, rather than loss due to network congestion. As such, it does not offer any active congestion control scheme.

PSFQ is designed for tasks that require reliable delivery of all message segments. Its focus is on the transport of binary images, such as new sensor control programs used for sensor re-tasking in the field. Since PSFQ expects low network traffic and does not provide any active congestion control scheme it may not be efficient for reliable transport of multipoint-to-point sensor events.

Loss Detection/Recovery

Reliability in PSFQ is achieved with a negative acknowledgement (NACK)-based quick fetch mechanism. Loss is detected using gap detection. Each injected message has a sequence number in the message header. If a receiving node determines a gap in sequence number, it begins aggressively broadcasting NACK messages to try to recover the lost message before the injection interval T_{min} is exceeded, and the next packet is sent.

In case a downstream node needs to quickly recover a lost packet, a NACK-based scheme requires upstream nodes to buffer messages that have been sent downstream, to conserve energy, NACK requests are bundled, as illustrated in Fig. 3. A sending node near the receiving node caches message segments it forwards; this recovery scheme is called "local recovery" PSFQ's assumption that all intermediate nodes store all the segments they forward may not be feasible on a real WSN due to a limited cache size on sensor nodes. At the very least the amount of segments stored would have to be heavily optimized for the small amount of storage space available on sensor nodes.

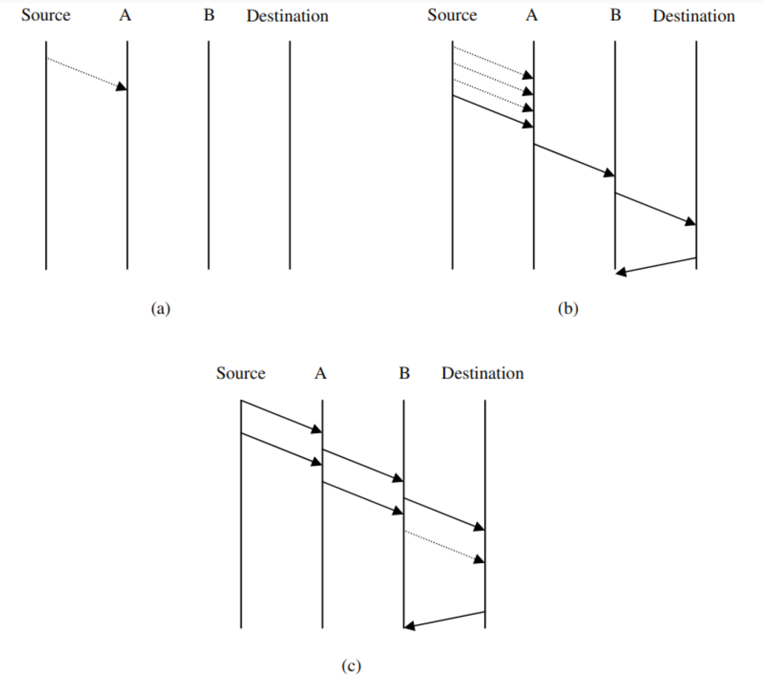


Figure 3. Loss detection/recovery in PSFQ. (a) A message consisting of a single data segment is sent from the Source and never received at node A. Since no data is ever received at node A, nothing can be recovered. (b) All data segments up to the last data segement are lost. The Destination receives the last data segment and is able to NACK for retransmission of all the lost data segments at once. (c) The last data segment is lost. The Destination creates a proactive fetch after some interval to retrieve the lost data segment.

A negative acknowledgement gap detection scheme leaves holes at the beginning and end of messages potentially undetected. Detecting dropped segments at the beginning of messages can only be done if one message segment is received downstream. If a message consists of only a single segment, and that segment is somehow dropped on the way downstream, it will not be detected. Likewise, a node cannot detect the loss of the last data segment in a transmission, since it will not be able to tell if the data segment has been lost or has not reached it yet.

To address the shortcomings of gap detection, PSFQ uses a "proactive fetch" scheme that allows it to set a timer that starts from the receipt of the last message until the next message is received. This continues while the total size of the received data segments is less than the file size specified in the header field of the inject message. If no message is received from any upstream neighbor before the timer times out, then a downstream sensor node will manually generate and broadcast a NACK event to actively try to recover the segements that were presumably lost. To save energy, proactive fetches, like the normal fetch mechanism, aggregate missing message segments into one NACK message.

PSFQ will buffer messages received if a gap is detected until the lost data segments have been recovered. As a side effect this means that data is delivered in order.

Congestion Control

PSFQ assumes light traffic in most cases in a WSN; not much is done to detect and control congestion. Instead, PSFQ attempts to avoid introducing congestion into the network through the use of a time-to-live (TTL) field in the segment header. Also, if a message with a sequence number lower than the last forwarded message is received, the message is silently discarded. Silently discarding messages helps to decrease the likelihood of flooding between the sensor nodes.

6.3.3 Reliable Multi-Segment Transport (RMST)

RMST is a reliable transport layer for WSNs. RMST is meant to operate on top of the gradient mechanism used in directed diffusion. RMST adds two important features to directed diffusion,

- fragmentation and reassembly of segments, and
- reliable message delivery.

One of the most intriguing features of RMST is that it is an extension of directed diffusion that can be applied to a sensor node and configured without having to recompile. Essentially RMST is a plugin transport layer mechanism for an already widely accepted and studied WSN network layer.

RMST can be configured to allow hop-by-hop recovery (using local broadcast NACK) or end-to-end recovery (end-to-end NACK) at run time, and can be combined with a MAC-level Automatic Repeat Query (ARQ). The configuration between hop-by-hop (cached) recovery and end-to-end (non-cached) recovery can be configured at the sensor nodes at runtime.

Loss Detection/Recovery Mechanisms

RMST employs a Negative Acknowledgement (NACK) gap detection to detect and recover lost messages similar to the scheme used by PSFQ. However, RMST makes no guarantee of in-order message delivery, rendering loss detection is particularly difficult since it is difficult for sensor nodes to determine whether gaps are caused by out-of-order delivery or lost messages. To help assuage this problem RMST creates a “hole map” for detected gaps and assigns a “watchdog” timer to generate an automatic NACK for any segment that has not been received in the timer interval.

Multiple fragment numbers can be combined into a single NACK, as in PSFQ, to cut down on the network traffic generated during message recovery, as shown in Fig. 4. Since RMST uses the same gap acknowledgement scheme as PSFQ, it inherits the same shortcomings when detecting loss of truncated messages. As seen with PSFQ’s recovery scheme, at least one data segment must be received downstream for RMST to detect message loss.

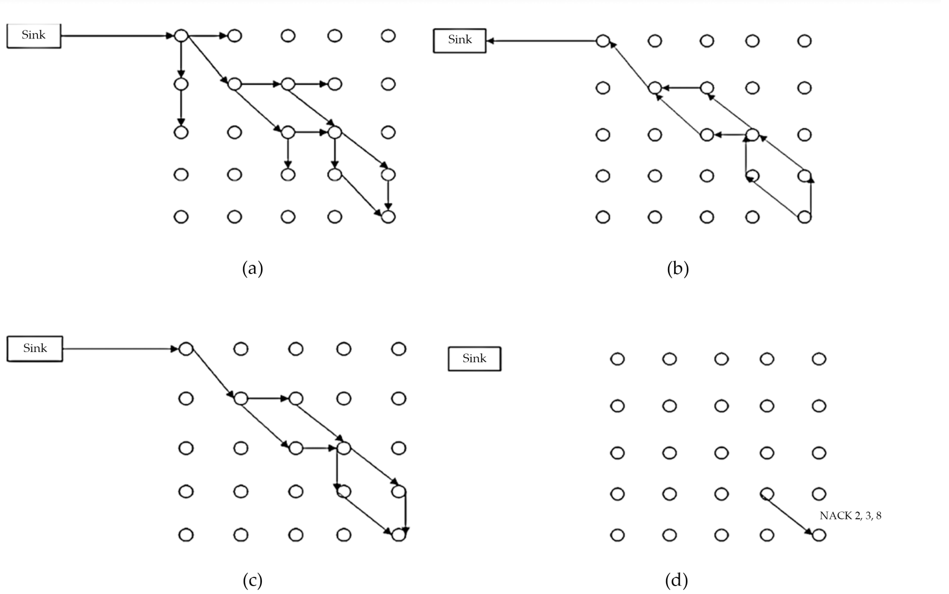


Figure 4. An example of the RMST protocol.

Congestion Control Mechanisms

RMST does not specify any congestion control or detection mechanism. It is concerned solely with reliable data transfer between the sensor nodes and the sink. Any congestion control mechanisms are a byproduct of the use of directed diffusion which offers minimal congestion control. For example, sensor nodes having gradients that show interest in the same information, but have different reporting intervals, may "downconvert" to the lower of the two reporting intervals.

6.3.4 ESRT (Event to Sink Reliable Transport)

ESRT introduces the idea of reliable event detection from the sensor nodes to the sink. ESRT leverages the loss tolerant characteristic of WSNs, the goal being to pass a course description of the event rather than providing fine details. Since ESRT will only reliably pass a course description of the event, it is unacceptable for applications that require delivery of all message segments. Unlike PSFQ and RMST, ESRT would be a good choice for tasks such as sensor retasking or transporting binary objects in general.

ESRT uses a different paradigm to measure reliability in wireless sensor networks. The assumption is not made that only messages in the point-to-multipoint direction, i.e. from the sink to the sensor nodes, is the only type of message that needs to be reliably delivered. Instead a measure of goodness is created using a defined event detection threshold and that threshold is used to define reliability in the multipoint-to-point direction.

The five essential features of ESRT are summarized in Table 2.

Table 2. Essential features of ESRT

Feature	Description
Self-configuration	Events must be detected reliably even in adverse network conditions. WSNs may also be randomly deployed. ESRT addresses this by controlling and adjusting the optimal operating interval.
Energy awareness	Sensor nodes have a finite lifetime. ESRT places most of the responsibility for ensuring reliability on the sink, since it is usually more robust. To extend the lifetime of the sensor nodes the sink may decrease the reporting frequency of sensor nodes.
Congestion control	ESRT will decrease the reporting rate of sensor nodes to alleviate congestion on WSNs while still using the event detection threshold to ensure that events are reliably detected.
Collective identification	Since sinks are more often interested in events than individual nodes, ESRT does not require individual node IDs. Instead event IDs are used to correlate data flows with events.
Biased implementation	To conserve energy algorithms used to ensure reliable event detection are mainly run on the sink. Since the sinks nodes are generally more robust nodes in a WSN, this feature conserves energy and preserves the lifetime of the sensor nodes.

Loss Detection/Recovery Mechanisms

ESRT's loss detection and recovery mechanism is tied inextricably to its congestion control mechanism. It does not prevent all losses, nor does it guarantee delivery of all message segments from all source nodes. Instead ESRT tries to find the correct frequency, f, to send messages.

Sankarasubramaniam et al. introduce definitions for observed event reliability, r_i, and desired event reliability, R. Observed event reliability, r_i, is defined as the number of data segments received over some interval i at the sink, and desired event reliability, R, is defined as the number of packets required for reliable event detection, i.e. R is the threshold for reliable event detection. Data segments are given event IDs, and thus r_i can be computed in real time by incrementing a counter at the sink for all correlated segments.

Sankarasubramaniam et al. control reliable event detection and network congestion by relating r_i and R to f. The problem of reliable event detection then becomes adjusting f to maintain r_i in an optimal interval around R. To help illustrate this Sankarasubramaniam et al. define five operating intervals.

Vuran et al. go on to further explore the idea of maximizing energy efficiency on WSNs by minimizing the transmission of highly correlated data flows. Eliminating the need to send data from all sensor nodes allows for some redundancy for the sensor nodes in WSNs and can prolong the lifetime of the network.

Congestion Control Mechanisms

ESRT recognizes the need for avoiding and controlling congestion in WSNs. To this end, ESRT defines the following five intervals illustrated in Table 3.

Table 3. ESRT defined operation intervals

Operation Interval	Abbreviation	Characteristics
No congestion, Low reliability	(NC, LR)	$f < f_{\max}$ and $\eta < 1 - \in$
No congestion, High reliability	(NC, HR)	$f \leq f_{\max}$ and $\eta > 1 + \in$
Congestion, High reliability	(C, HR)	$f > f_{\max}$ and $\eta > 1$
Congestion, Low reliability	(C, LR)	$f > f_{\max}$ and $\eta \leq 1$
Optimal Operating Region	OOR	$f < f_{\max}$ and $1 - \in \leq \eta \leq 1 + \in$

ESRT provides a new twist on providing reliability in WSNs. It introduces the idea that reliable data on a sensor network can mean not only delivering an entire binary object reliably, but for tasks where some loss is acceptable we should still provide a measure of reliability that provides the gathering entity with a measure of the "goodness" of the data.

6.4 TRADITIONAL TRANSPORT CONTROL PROTOCOLS

Transmission Control Protocol (TCP) is the transport layer protocol that serves as an interface between client and server. The TCP/IP protocol is used to transfer the data packets between transport layer and network layer. Transport protocol is mainly designed for fixed end systems and fixed, wired networks. In simple terms, the traditional TCP is defined as a wired network while classical TCP uses wireless approach. Mainly TCP is designed for fixed networks and fixed, wired networks.

The main research activities in TCP are as listed below.

6.4.1 Congestion Control

During data transmission from sender to receiver, sometimes the data packet may be lost. It is not because of hardware or software problem. Whenever the packet loss is confirmed, the probable reason might be the temporary overload at some point in the transmission path. This temporary overload is otherwise called as Congestion.

Congestion is caused often even when the network is designed perfectly. The transmission speed of receiver may not be equal to the transmission speed of the sender. if the capacity of the sender is more than the capacity of output link, then the packet buffer of a router is filled and the router cannot forward the packets fast enough. The only thing the router can do in this situation is to drop some packets.

The receiver sense the packet loss but does not send message regarding packet loss to the sender. Instead, the receiver starts to send acknowledgement for all the received packets and the sender soon identifies the missing acknowledgement. The sender now notices that a packet is lost and slows down the transmission process. By this, the congestion is reduced. This feature of TCP is one of the reason for its demand even today.

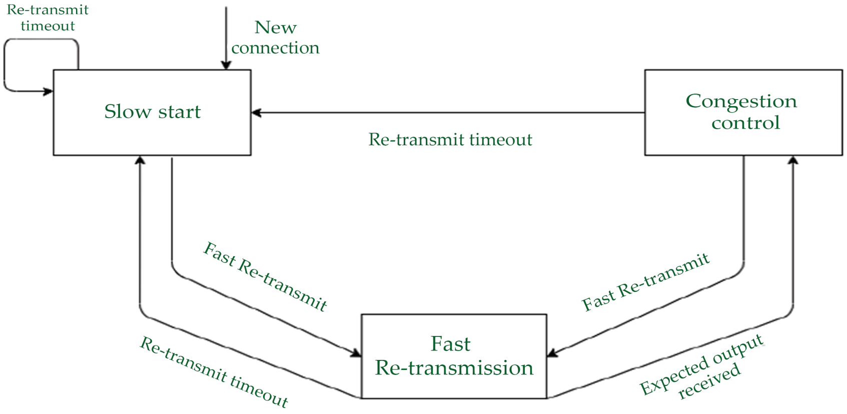
6.4.2 Slow Start

The behavior TCP shows after the detection of congestion is called as slow start. The sender always calculates a congestion window for a receiver. At first the sender sends a packet and waits for the acknowledgement. Once the acknowledgement is back it doubles the packet size and sends two packets. After receiving two acknowledgements, one for each packet, the sender again doubles the packet size and this process continues. This is called Exponential growth.

It is dangerous to double the congestion window each time because the steps might become too large. The exponential growth stops at congestion threshold. As it reaches congestion threshold, the increase in transmission rate becomes linear (i.e., the increase is only by 1). Linear increase continues until the sender notices gap between the acknowledgments. In this case, the sender sets the size of congestion window to half of its congestion threshold and the process continues.

6.4.3 Fast Re-transmission

In TCP, two things lead to a reduction of the congestion threshold. One of those is sender receiving continuous acknowledgements for the single packet. By this it can convey either of two things. One such thing is that the receiver received all the packets up to the acknowledged one and the other thing is the gap is due to packet loss. Now the sender immediately re-transmit the missing packet before the given time expires. This is called as Fast re-transmission.



Traditional TCP

6.4.4 Example

Assume that few packets of data are being transferred from sender to receiver, and the speed of sender is 2 Mbps and the speed of receiver is 1 Mbps respectively. Now the packets that are being transferred from sender sender to receiver makes a traffic jam inside the network. Due to this the network may drop some of the packets. When these packets are lost, the receiver sends the acknowledgement to the sender and the sender identifies the missing acknowledgement. This process is called as congestion control.

Now the slowstart mechanism takes up the plan. The sender slows down the packet transfer and then the traffic is slightly reduces. After sometime it puts a request to fast re-transmission through which the missing packets can be sent again as fast as possible. After all these mechanisms, the process of next packet begins.

6.5 USER DATAGRAM PROTOCOL (UDP)

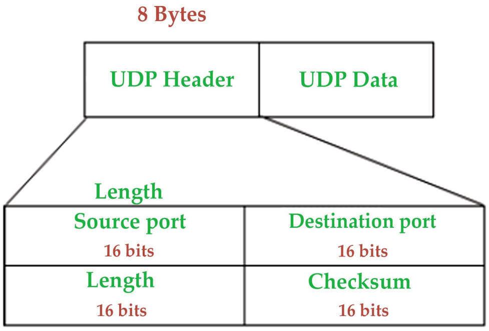
User Datagram Protocol (UDP) is a Transport Layer protocol. UDP is a part of the Internet Protocol suite, referred to as UDP/IP suite. Unlike TCP, it is an unreliable and connectionless protocol. So, there is no need to establish a connection prior to data transfer.

Though Transmission Control Protocol (TCP) is the dominant transport layer protocol used with most of the Internet services; provides assured delivery, reliability, and much more but all these services cost us additional overhead and latency. Here, UDP comes into the picture. For real-time services like computer gaming, voice or video communication, live conferences; we need UDP. Since high performance is needed, UDP permits packets to be dropped instead of processing delayed packets. There is no error checking in UDP, so it also saves bandwidth.

User Datagram Protocol (UDP) is more efficient in terms of both latency and bandwidth.

6.5.1 UDP Header

UDP header is an **8-bytes** fixed and simple header, while for TCP it may vary from 20 bytes to 60 bytes. The first 8 Bytes contains all necessary header information and the remaining part consist of data. UDP port number fields are each 16 bits long, therefore the range for port numbers is defined from 0 to 65535; port number 0 is reserved. Port numbers help to distinguish different user requests or processes.



- **Source Port:** Source Port is a 2 Byte long field used to identify the port number of the source.
- **Destination Port:** It is a 2 Byte long field, used to identify the port of the destined packet.
- **Length:** Length is the length of UDP including the header and the data. It is a 16-bits field.
- **Checksum:** Checksum is 2 Bytes long field. It is the 16-bit one's complement of the one's complement sum of the UDP header, the pseudo-header of information from the IP header, and the data, padded with zero octets at the end (if necessary) to make a multiple of two octets.

Notes – Unlike TCP, the Checksum calculation is not mandatory in UDP. No Error control or flow control is provided by UDP. Hence UDP depends on IP and ICMP for error reporting.

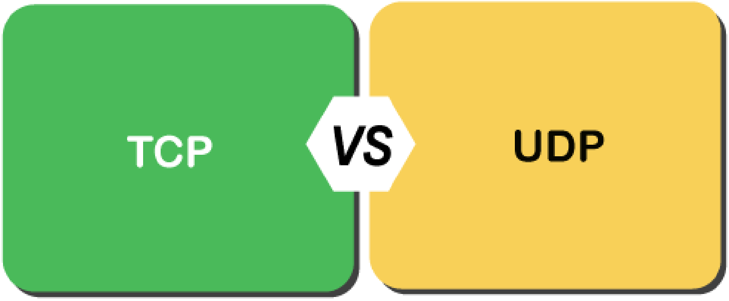
6.5.2 Applications of UDP

- Used for simple request-response communication when the size of data is less and hence there is lesser concern about flow and error control.
- It is a suitable protocol for multicasting as UDP supports packet switching.
- UDP is used for some routing update protocols like RIP (Routing Information Protocol).
- Normally used for real-time applications which cannot tolerate uneven delays between sections of a received message.
- Following implementations uses UDP as a transport layer protocol:
  - NTP (Network Time Protocol)
  - DNS (Domain Name Service)
  - BOOTP, DHCP.
  - NNP (Network News Protocol)
  - Quote of the day protocol
  - TFTP, RTSP, RIP.
- The application layer can do some of the tasks through UDP-
  - Trace Route
  - Record Route

  - Timestamp
- UDP takes a datagram from Network Layer, attaches its header, and sends it to the user. So, it works fast.
- Actually, UDP is a null protocol if you remove the checksum field.

1. Reduce the requirement of computer resources.
2. When using the Multicast or Broadcast to transfer.
3. The transmission of Real-time packets, mainly in multimedia applications.

6.5.3 Differences between the TCP and UDP



- **Type of protocol:** Both the protocols, i.e., TCP and UDP, are the transport layer protocol. TCP is a connection-oriented protocol, whereas UDP is a connectionless protocol. It means that TCP requires connection prior to the communication, but the UDP does not require any connection.
- **Reliability:** TCP is a reliable protocol as it provides assurance for the delivery of the data. It follows the acknowledgment mechanism. In this mechanism, the sender receives the acknowledgment from the receiver and checks whether the acknowledgment is positive or negative. If the ACK is positive means, the data has been received successfully. If ACK is negative, then TCP will resend the data. It also follows the flow and error control mechanism. UDP is an unreliable protocol as it does not ensure the delivery of the data.
- **Flow Control:** TCP follows the flow control mechanism that ensures a large number of packets are not sent to the receiver at the same time, while UDP does not follow the flow control mechanism.
- **Ordering:** TCP uses ordering and sequencing techniques to ensure that the data packets are received in the same order in which they

are sent. On the other hand, UDP does not follow any ordering and sequencing technique; i.e., data can be sent in any sequence.

- **Speed:** Since TCP establishes a connection between a sender and receiver, performs error checking, and also guarantees the delivery of data packets while UDP neither creates a connection nor it guarantees the delivery of data packets, so UDP is faster than TCP.
- **Flow of data:** In TCP, data can flow in both directions means that it provides the full-duplex service. On the other hand, UDP is mainly suitable for the unidirectional flow of data.

	TCP	**UDP**
Full form	It stands for Transmission Control Protocol.	It stands for User Datagram Protocol.
Type of connection	It is a connection-oriented protocol, which means that the connection needs to be established before the data is transmitted over the network.	It is a connectionless protocol, which means that it sends the data without checking whether the system is ready to receive or not.
Reliable	TCP is a reliable protocol as it provides assurance for the delivery of data packets.	UDP is an unreliable protocol as it does not take the guarantee for the delivery of packets.
Speed	TCP is slower than UDP as it performs error checking, flow control, and provides assurance for the delivery of	UDP is faster than TCP as it does not guarantee the delivery of data packets.
Header size	The size of TCP is 20 bytes.	The size of the UDP is 8 bytes.
Acknowledgment	TCP uses the three-way-handshake concept. In this concept, if the sender receives the ACK, then the sender will send the data. TCP also has the ability to resend the lost data.	UDP does not wait for any acknowledgment; it just sends the data.
Flow control mechanism	It follows the flow control mechanism in which too many packets cannot be sent to the receiver at the same time.	This protocol follows no such mechanism.
Error checking	TCP performs error checking by using a checksum. When the data is corrected, then the data is retransmitted to the receiver.	It does not perform any error checking, and also does not resend the lost data packets.
Applications	This protocol is mainly used where a secure and reliable communication process is required, like military services, web browsing, and e-mail.	This protocol is used where fast communication is required and does not care about the reliability like VoIP, game streaming, video and music streaming, etc.

6.6 ISSUES OF TRANSPORT CONTROL PROTOCOLS FOR WIRELESS

Wireless sensor networks have been experiencing more and more attentions in academia and industry in recent years, especially under the possibility of much more cheap sensors with certain computation and communication capability. WSNs can be used for many applications such as habitat monitoring, in-door monitoring, target tracking, and security surveillance, etc. However there is a path before commercially deploying sensors, because WSNs have some problems to be overcome, for example, energy-conservation, congestion control, reliability data dissemination, security, and management of a WSN itself. These problems often involve in one or several layers top-down from application layer to physical layer, and can be studies separately in each corresponding layer, or collaboratively cross each layer. For example, congestion control may involve in only transport layer, but energy-conservation may be related to physical layer, data link layer, network layer, and high layers. Some researchers recently turn their attentions to transport control protocols, which are important for reliable data dissemination and energy-conservation for WSNs.

Generally speaking, transport control protocols, especially for connection-oriented transport protocols, may include two main functions: congestion control and loss recovery. As for congestion control, it is firstly required how to detect whether or not congestion happens, and when and where it happens. Congestion can be detected through monitoring node buffer occupancy and link (or wireless channel) load. In traditional Internet, the methods to weaken congestion are packet dropping at congestion point such as AQM (active queue management), rate decreasing in source node such as AIMD (Additive Increase Multiplicative Decrease) in TCP (Transport Control Protocol), and routing techniques. For WSNs, it should be carefully considered how to detect congestion and how to overcome it, because sensors are often with limited resources. These protocols must consider its simplicity and scalability to save energy and possibility to prolong the life-time of whole networks. For example, in order to weaken congestion, we can use end-to-end mechanism like TCP or hop-by-hop backpressure like that in ATM (Asynchronous Transfer Mode) networks or Frame Relay networks. The end-to-end approaches are very simple and robust, but it will bring with more on-going packets in networks. However, hop-by-hop approaches can quickly weaken congestion and bring with less on-going packets in networks, while it needs to change the behavior of each node on the way from source to destination. Since less on-going packets can result in saved energy, there is a trade-off between end-to-end and hop-by-hop mechanism, which should be carefully considered when designing practical congestion control algorithms for WSNs.

Packet loss is usual under wireless sensor networks due to bad quality of wireless channel, sensor failure, and/or congestion. WSNs must guarantee certain reliability in packet-level or application-level through loss recovery in order to abstract correct information. Some critical applications need reliable transmission of each packet and thus packet-level reliability is needed. Other applications need only a proportionally reliable transmission of total packets and thus application reliability is needed. Anyway, we first need to detect packet loss in order to correctly recover missing packets. The traditional methods used in packet-switched networks can be used to detect packet loss for wireless sensor networks. For example, each packet can piggyback sequence number, and a receiver can detect loss though arbitrating the continuity of received sequence number during a time interval (if not considering of packet disorder resulted from multi-path). After detecting packet loss, ACK and/or NACK (and their variant) can be used to recover missing packets based on an end-to-end or hop-by-hop approach. Like that in congestion control, there is still a trade-off between end-to-end and hop-by-hop approach, which should be thought over. When designing transport control protocols for wireless sensor networks, we must consider energy-conservation at the same time.

Intuitively, if there are few on-going packets and few re-transmissions, energy can be saved. Effective congestion control can result in few on-going packets and effective loss recovery approach can result in few re-transmissions. So congestion control and reliability guarantee can additionally save energy in a wireless sensor network. In a summary, the problem of transport control protocols for sensor networks is how to effectively control congestion and how to guarantee reliability while conserving energy as more as possible simultaneously.

6.6.1 Disadvantages of TCP and UDP

TCP and UDP (User Datagram Protocol) are two well-known transport control protocols widely deployed in Internet. But both of them are not the good choice for WSNs. First let us see the characteristics of TCP protocols as follows:

- TCP is a connection-oriented protocol. Before data transmission, there is a three-way handshake interactive process. If and only if after the TCP connection has been established, TCP sender can begin to transmit data. In WSNs, the sensed data for event-based applications is just several bytes or so (a value of an interest). The three-way handshake process will be a big overhead for the small volume data. Also since wireless link is error-prone under WSNs, the time to setup TCP connection might be much longer than that under Internet. Then the data will be probably outdated after TCP connection has been established.

- In TCP, it is assumed that all segment losses are resulted from congestion and will trigger window-based flow control and congestion control. This style will incur that TCP will unwisely reduce transmission rate under WSNs when there is no congestion, but packet losses from bit-error. The behavior will lead to low throughput especially under multiple wireless hops. Therefore it is hard for sensor nodes, especially the ones far away from sink, to obtain enough throughout to support such WSNs applications that require continual data transmissions.
- TCP uses end-to-end approach to control congestion. This approach generally has longer response time when congestion occurs, and in-turn will result in lots of segment dropping. The segment dropping means useless energy consumption and not energy-efficient. Also the long response time will make it hard to fully fill wireless channel after congestion.
- TCP uses end-to-end ACK and retransmission to guarantee reliability. This approach will cause much lower throughput and longer transmission time if RTT (Round-Trip Time) is larger as that in large-scale WSNs, since the sender will stop to wait for the ACK after each data transmission.
- Under WSNs, sensor nodes may have different hops and different RTT from sink. TCP in such environment may cause unfairness. The sensor nodes near to sink may get more opportunities to transmit data and may deplete their energy first, and the whole wireless sensor network will be disjointed with a high probability.

Although UDP is a connectionless transport control protocol, it is still not suitable for WSNs considering the following reasons:

- There is no any flow control and congestion control mechanism in UDP. If UDP is used for WSNs, it will cause lots of datagram dropping when congestion happens. In this point at least, UDP is not energy-efficient for WSNs.

- UDP contains no ACK mechanism, no any reliability mechanism. The datagram loss can be only recovered by lower MAC algorithms or upper layers including application layer.

Beside the disadvantages listed above, there is no any interaction between TCP (or UDP) and lower layer protocols such as routing and MAC (Medial Access Control) algorithm. But under wireless sensor networks, the lower layers can provide rich and helpful information to transport control layer and make it possible to optimize system performance. In a summary, neither TCP nor UDP are well suitable for wireless sensor networks.

6.7 THE DESIGN ISSUES OF TRANSPORT CONTROL PROTOCOLS FOR WSNS

Generally speaking, the transport control protocol for WSNs should consider the following factors. First, it should provide congestion control mechanism and guarantee reliability, especially the latter. The most data streams are flowed from sensor nodes to sink in WSNs, so congestion might occur around sink. Also there are some high-bandwidth data streams produced by multi-media sensors. Therefore it is necessary to design effective congestion detection, congestion avoidance, and congestion control mechanisms for WSNs. Although MAC protocol can recover packets loss from bit-error, it has no way to handle packets loss from buffer overflow. Then the transport protocol for WSNs should have mechanism for packets loss recovery such as ACK and Selective ACK used in TCP protocol so as to guarantee reliability. At the same time, the reliability under WSNs may have different meaning from traditional networks where it generally guarantees the correct transmission of every packet. For some application, WSNs only needs to correctly receive packets from a certain area not every sensor nodes in this area, or some ratio of successful transmission from a sensor node. These new reliability can be utilized to design more efficient transport control protocols. It would be better to use hop-by-hop mechanism for congestion control and loss recovery since it can reduce packet dropping and conserve energy. The hop-by-hop mechanism can lower the buffer requirement in intermediate nodes simultaneously. It is helpful for sensor nodes with limited memory.

Second, transport control protocols for wireless sensor networks should simplify initial connecting process or use connectionless protocol so as to speedup start and guarantee throughput and lower transmission delay. Most of applications in WSNs are reactive which passively monitor and wait for event occurring before reporting to sink. These applications may have only several packets for each reporting, and the simple and short initial setup process is more effective and efficient.

Third, the transport control protocols for WSNs should avoid as few packets dropping as possible since packet dropping means energy wastage. In order to avoid packet dropping, the transport protocol can use active congestion control at the cost of a bit lower link utility. The active congestion control (ACC) can trigger congestion avoidance before congestion occurs. An example of ACC is to make sender (or intermediate nodes) reduce sending (or forwarding) rate when the buffer size of their downstream neighbors overruns a threshold.

Fourth, the transport control protocols should guarantee fairness for different sensor nodes in order that each sensor nodes can achieve fair throughput. Otherwise the biased sensor nodes cannot report the events in their area and system may misunderstand there is no any event in the area. Fifth, it would be better if the transport control protocol can enable cross-layer optimization. For example, if routing algorithm can tell route failure to transport protocol, the transport protocol will know the packet loss is not from congestion but from route failure and the sender will frozen its status and keep its current sending rate to guarantee high throughput and low delay.

6.8 THE EXISTING TRANSPORT CONTROL PROTOCOLS FOR WSNS

There are several transport control protocols (see Table 4) for wireless sensor networks. They aim at congestion control and/or reliability guarantee in upstream (from sensor nodes to sink) or downstream (from sink to sensor nodes), and can be classified into four types: upstream congestion control, downstream congestion control, upstream reliability guarantee, and downstream reliability guarantee.

Table 4. Several transport control protocols for WSNs

Attributes		**CODA**	**ESRT**	**RMST**	**PSFQ**	**GARUDA**	**SenTCP**
Direction		Upstream	Upstream	Upstream	Down-stream	Down-stream	Upstream
Congestion	Support	Yes	Passive	No	No	No	Yes
	Congestion detection	Buffer size& Channel condition	Buffer size	-	-	-	Buffer size& Pkts arrival rare
	Open-loop or Closed-loop	Both	Close	-	-	-	
Reliability	Support	No	Yes	Yes	Yes	Yes	No
	Packet or Application Reliability	-	Application	Packet	Packet	Packet	-
	Loss detection		No	Yes	Yes	Yes	
	End-to-End or Hop-by-Hop	-	E2E	HbH	HbH	HbH	
	Cache	-	No	Yes or No	Yes	Yes	
	In-sequence or Out-of-sequence NACK	-	N/A	In-seq	Out-of-seq	Out-of-seq	
	ACK or NACK	-	ACK	NACK	NACK	NACK	
IEnergy-conservation		Good	Fair	No result	No result	Yes	Good

(E2E: End-to-end; HbH: Hop-by-hop; Upstream: from sensor to sink; Down-stream: from sink to sensor)

6.8.1 CODA

CODA (COngestion Detection and Avoidance) belongs to upstream congestion control. It contains three components: congestion detection, open-loop hop-by-hop backpressure, and closed-loop end-to-end multi-source regulation. CODA attempts to detect congestion by monitoring current buffer occupancy and wireless channel load. If buffer occupancy or wireless channel load exceeds a threshold-based value, it means that congestion happens. Then node detecting congestion will notify its upstream neighbor nodes to decrease rate, with the manner of open-loop hop-by-hop backpressure. The upstream neighbor nodes will trigger to decrease output rate like AIMD and to replay backpressure continuously, after they receive backpressure signal. Finally CODA can regulate multi-source rate through closed-loop end-to-end approach, which works as follows:

- When a sensor rate overruns theoretical throughput, it will set "regulation" bit in even packet.
- If the event packet received by sink has "regulation" bit, sink should send ACK control message to sensors and to inform them to decrease their rate.
- If congestion is cleared, sink will actively send ACK control message to sensors and to inform them to increase their rate.

CODA uses AIMD-like mode in TCP protocols to regulate sensors rate. CODA brings with such disadvantages:

- Unidirectional control from sensors to sink;
- Consider no reliability but congestion control;
- Result in decreased reliability (although conserving energy) especially under such scenarios with sparse source and high data rate;
- The delay or response time of closed-loop multi-source regulation will be increased under heavy congestion since the ACK issued from sink would loss with high probability at this time.

6.8.2 ESRT

ESRT (Event-to-Sink Reliable Transport) aims at providing reliability from sensors to sink while congestion control simultaneously. It belongs to upstream reliability guarantee. Firstly it needs to periodically compute the factual reliability r according to successfully received packets in a time interval. Secondly and the most importantly, ESRT deduces the required sensor report frequency f from r: f=G(r). Thirdly and finally, ESRT informs f to all sensors through an assumed channel with high power and sensors can report even and transmit packets with frequency f. ESRT is an End-to-End approach to guarantee a desired reliability through regulating sensor report frequency. It provides reliability for applications not for each single packet. The addition benefit resulted from ESRT is energy-conservation since it can control sensor report frequency. ESRT brings with such disadvantages:

- ESRT regulates report frequency of all sensors using the same value. But it may be more reasonable if using different value since each sensor may have different contributions to congestion.
- ESRT assumes and uses a channel (one-hop) with high power that will influence the on-going data transmission.
- ESRT mainly considers reliability and energy-conservation.

6.8.3 RMST

RMST (Reliable Multi-Segment Transport) also belongs to upstream reliability guarantee. It is designed to run above Directed Diffusion (to use its discovered path from sensors to sink) in order to provide guaranteed reliability from sensors to sink (delivery and fragmentation/reassembly) for applications. RMST is a selective NACK-based protocol. RMST basically operates as follows. Firstly, RMST uses timer-driver mechanism to detect data loss and send NACK on the way from detecting node to sources (Cache or non-Cache mode). Secondly, NACK receivers are responsible for looking for the missing packet, or forward NACK on the path toward sink if it fails to find the missing packet or in non-cache mode. Several advantages of RMST are including:

- No congestion control.
- No effective energy conservation mechanism.
- No application-level reliability.

6.8.4 PSFQ

PSFQ (Pump Slowly Fetch Quickly) aims to distribute data from sink to sensors by pacing data at a relatively slow-speed, but allowing nodes that experience data loss to fetch (recover) any missing segments from immediate neighbors very aggressively (local recovery, “fetch quickly”).

It belongs to downstream reliability guarantee. The motivation of PSFQ is to achieve loose delay bounds while minimizing the loss recovery cost by localized recovery of data among immediate neighbors. It contains three components: Pump operation, Fetch operation, and Report operation. Firstly, sink slowly broadcasts a packet (with such fields-file ID, file length, sequence number, TTL, and report bit) to its neighbors every T until all the data fragments has been sent out. Secondly, a sensor can go into fetch mode once a sequence number gap in a file fragment is detected and issue NACK in reverse path to recover missing fragment. The NACK don't need to be relayed unless the number of times the same NACK is heard exceeds a predefined threshold while the missing segments requested by the NACK message are no longer retained in a sensor's cache. Thirdly, sink can make sensors to feedback data delivery status information to it through a simple and scalable hop-by-hop report mechanism. PSFQ has several disadvantages:

- PSFQ can't detect the loss of single packet since it used only NACK not ACK.
- PSFQ uses statically and slowly pump that result in large delay.
- Hop-by-hop recovery with cache will need more buffer.

6.8.5 GARUDA

GARUDA belongs to downstream reliability guarantee. It has three primary components. Firstly, GARUDA uses WFP (Wait-for First -Packet) pulse transmission to guarantee success of single/first packet delivery, in order to choose and construct Core sensors. Secondly, GARUDA performs Core election using such methods-only sensors with HopCount of the form 3*i where i is a positive integer, are allowed to elect themselves as Core sensors. Thirdly, GARUDA begins two phase loss recovery-Loss recovery for Core sensors and Loss recovery for non-Core sensors using out-of-sequence NACK. Several disadvantages are including:

- Support reliability only on the downstream direction from sink to sensors.
- It provides no congestion control.

6.8.6 ATP

ATP is a new transport protocol for ad-hoc networks. It is a receiver-based and network-assisted end-to-end feedback control algorithm. It uses selective ACKs (SACKs) for packets loss recovery. In ATP, intermediate network nodes compute the sum of exponentially averaged packet queuing delay and transmission delay, called D. The idea is that the required end-to-end rate should be the reverse of D. The D is computed over all the packets traversing the node and used to update the value piggybacked in each outgoing packet if the new value of D is bigger than the old value. After this hop-by-hop computation and piggyback, the receiver can get the largest value of D that each packet experience on the way. Then the receiver can calculate the required end-to-end rate, the reverse of D, for the sender and feedback it to the sender. Then the sender can intelligently adjust its sending rate according to received D from the receiver. In order to guarantee reliability, ATP uses selective ACKs (SACKs) as an end-to-end mechanism for loss detection. But the SACK block in ATP is 20, much larger that that in TCP (only 3). ATP decouples congestion control and reliability and achieves better fairness and higher throughput than TCP. But it doesn't consider energy issues and its end-to-end approach might be not the optimal for sensor networks.

6.8.7 SenTCP

SenTCP is an open-loop hop-by-hop congestion control protocol with two special features: 1) It jointly uses average local packet service time and average local packet inter-arrival time in order to estimate current local congestion degree in each intermediate sensor node. The use of packet arrival time and service time not only precisely calculates congestion degree, but effectively helps to differentiate the reason of packet loss occurrence in wireless environments, since arrival time (or service time) may become small (or large) if congestion occurs. 2) It uses hop-by-hop congestion control. In SenTCP, each intermediate sensor node will issues feedback signal backward and hop-by-hop. The feedback signal, which carries local congestion degree and the buffer occupancy ratio, is used for the neighboring sensor nodes to adjust their sending rate in the transport layer. The use of hop-by-hop feedback control can remove congestion quickly and reduce packet dropping, which in turn conserves energy. SenTCP realizes higher throughput and good energy-efficiency since it obviously reduces packet dropping; however, SenTCP copes with only congestion and guarantees no reliability.

6.9 THE FUTURE PROBLEMS OF THE EXISTING TRANSPORT CONTROL PROTOCOLS FOR WIRELESS SENSOR NETWORKS

The major functions of transport control protocols for wireless sensors networks are congestion control, reliability guarantee, and energy conservation that can be passively realized by congestion control and reliability guarantee. What the existing protocols studied is only either congestion or reliability guarantee in uni-direction (upstream or downstream), and none of them settles congestion control and reliability simultaneously in both directions. Moreover, some protocols such as only focusing on congestion control have decreased reliability. However some applications in wireless sensor networks require both functions in both directions, for example, re-tasking and critical time-sensitive monitoring and surveillance.

The second problem of the existing transport control protocols for wireless sensor networks is that they control congestion either through end-to-end or through hop-by-hop (Although there are end-to-end and hop-by-hop mechanism for congestion control in CODA, CODA only simply uses them at the same time, and has no any adaptive method to integrate the two mechanisms for optimization). But an adaptive congestion control that integrates end-to-end and hop-by-hop may be more helpful for wireless sensor networks with diverse applications on it, and useful for energy-conservation and simplification of sensor operation.

The third problem is that the protocols guaranteeing reliability provide either packet-level reliability or application-level reliability, not both of them. If a sensor network supports two applications (one of them needs only packet-level reliability, and another needs only application-level reliability), the existing transport control protocols will face difficulty and will be not the optimal choices. Therefore, an adaptive recovery mechanism is required to support packet-level and application-level reliability, and to be helpful for energy-conservation.

The fourth problem is that the existing transport control protocols have hardly implemented any cross-layer optimization. However lower-layers such as network layer and MAC layer can provide useful information up to transport layer. A new effective and cross-layer optimized transport control protocol can be available through such cross-layer optimization.

SUMMARY

- Wireless sensor networks (WSNs) provide a powerful means to collect information on a wide variety of natural phenomena. WSNs typically consist of a cluster of densely deployed nodes communicating with a sink node which, in turn, communicates with the outside world.
- TCP stands for Transmission Control Protocol. It is a transport layer protocol that facilitates the transmission of packets from source to destination. It is a connection-oriented protocol that means it establishes the connection prior to the communication that occurs between the computing devices in a network. This protocol is used with an IP protocol, so together, they are referred to as a TCP/IP.
- TCP is a transport layer protocol as it is used in transmitting the data from the sender to the receiver.
- Traffic from many applications in WSNs is considered loss tolerant. Loss tolerance in WSNs is due to the dense deployment of sensor nodes and data aggregation properties, giving rise to directional reliability.
- Transmission Control Protocol (TCP) is the transport layer protocol that serves as an interface between client and server. The TCP/IP protocol is used to transfer the data packets between transport layer and network layer. Transport protocol is mainly designed for fixed end systems and fixed, wired networks.
- User Datagram Protocol (UDP) is a Transport Layer protocol. UDP is a part of the Internet Protocol suite, referred to as UDP/IP suite. Unlike TCP, it is an unreliable and connectionless protocol. So, there is no need to establish a connection prior to data transfer.
- Generally speaking, transport control protocols, especially for connection-oriented transport protocols, may include two main functions: congestion control and loss recovery. As for congestion control, it is firstly required how to detect whether or not congestion happens, and when and where it happens. Congestion can be detected through monitoring node buffer occupancy and link (or wireless channel) load.
- CODA (COngestion Detection and Avoidance) belongs to upstream congestion control. It contains three components: congestion detection, open-loop hop-by-hop backpressure, and closed-loop end-to-end multi-source regulation.

- ESRT (Event-to-Sink Reliable Transport) aims at providing reliability from sensors to sink while congestion control simultaneously. It belongs to upstream reliability guarantee.
- RMST (Reliable Multi-Segment Transport) also belongs to upstream reliability guarantee. It is designed to run above Directed Diffusion (to use its discovered path from sensors to sink) in order to provide guaranteed reliability from sensors to sink (delivery and fragmentation/reassembly) for applications.
- PSFQ (Pump Slowly Fetch Quickly) aims to distribute data from sink to sensors by pacing data at a relatively slow-speed, but allowing nodes that experience data loss to fetch (recover) any missing segments from immediate neighbors very aggressively (local recovery, "fetch quickly"). It belongs to downstream reliability guarantee.
- ATP is a new transport protocol for ad-hoc networks. It is a receiver-based and network-assisted end-to-end feedback control algorithm. It uses selective ACKs (SACKs) for packets loss recovery.

REFERENCES

1. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless sensor networks: A survey,” Computer Networks, vol. 38, 2002, pp. 393-422.
2. C.-Y. Wan, S. B. Eisenman, and A. T. Campbell, “CODA: Congestion detection and avoidance in sensor networks,” in Proceedings of ACM Sensys’03, November 5-7, 2003, Los Angeles, USA.
3. Y. Sankarasubramaniam, O. B. Akan, and I. F. Akyidiz, “ESRT: Event-to-sink reliable transport in wireless sensor networks,” in Proceedings of ACM Mobihoc’03, June 1-3, 2003, Annapolis, USA.
4. F. Stann and J. Heidemann, “RMST: Reliable data transport in senor networks,” in Proceedings of IEEE SNPA’03, May 11, 2003, Anchorage, USA.
5. C.-Y. Wan, A. T. Campbell, “PSFQ: A reliable transport protocol for wireless sensor networks,” in Proceedings of ACM WSNA’02, September 28, 2002, Atlanta, USA.
6. S.-J. Park, R. Vedantham, R. Sivakumar, and I. F. Akyildiz, “A scalable approach for reliable downstream data delivery in wireless sensor networks,” in Proceedings of ACM MobiHoc’04, May 24-26, 2004, Roppongi, Japan.
7. K. Sundaresan, V. Anantharaman, H.-Y. Hseeh, and R. Sivakumar, “ATP: A reliable transport protocol for ad-hoc networks,” in Proceedings of ACM Mobihoc’03, June 1-3, 2003, Annapolis, Maryland, USA.
8. C. Wang, K. Sohraby, and B. Li, “SenTCP: A hop-by-hop congestion control protocol for wireless sensor networks,” in Proceedings of IEEE INFOCOM 2005 (Poster Paper), Miami, Florida, USA, Mar. 2005.

CHAPTER 7

MIDDLEWARE FOR WIRELESS SENSOR NETWORK

INTRODUCTION

Wireless Sensor Networks (WSNs) have found more and more applications in a variety of pervasive computing environments. However, how to support the development, maintenance, deployment and execution of applications over WSNs remains to be a nontrivial and challenging task, mainly because of the gap between the high level requirements from pervasive computing applications and the underlying operation of WSNs. Middleware for WSN can help bridge the gap and remove impediments.

In recent years, a new wave of networks labeled Wireless Sensor Networks (WSNs) has attracted a lot of attentions from researchers in both academic and industrial communities. WSNs can be used to form the underlying sensing and network infrastructure for pervasive computing environments. A WSN consists a collection of sensor nodes and a sink node connected through wireless channels, and can be used to build distributed systems for data collection and processing, covering the functions of on-field signal sensing and processing, in-network data aggregation, and self-organized wireless communication. WSNs have found many applications in different areas, including environmental surveillance, intelligent building, health monitoring, intelligent transportations, etc.

Middleware refers to software and tools that can help hide the complexity and heterogeneity of the underlying hardware and network platforms, ease the management of system resources, and increase the predictability of application executions. WSN middleware is a kind of middleware providing the desired services for sensing based pervasive computing applications that make use of a wireless sensor network and the related embedded operating system or firmware of the sensor nodes.

The motivation behind the research on WSN middleware derives from the gap between the high-level requirements from pervasive computing applications and the complexity of the operations in the underlying WSNs. The application requirements include high flexibility, re-usability, and reliability. The complexity of the operations with a WSN is characterized by constrained resources, dynamic network topology, and low level embedded OS APIs. WSN middleware provides a potential solution to bridge the gap and remove the impediments. In the early time of the research on WSN, people did not pay much attention to middleware because the simplicity of the early applications did not show much demand on the support from the middleware. Along with the rapid evolution of this area, the gap becomes increasingly obvious and hinders the popularity of WSN based applications.

WSN middleware helps the programmer develop applications in several ways. First, it provides appropriate system abstractions, so that the application programmer can focus on the application logic without caring too much about the lower level implementation details. Second, it provides reusable code services, such as code update, and data services, such as data filtering, so that the application programmer can deploy and execute the application without being troubled with complex and tedious functions. Third, it helps the programmer in network infrastructure management and adaptation by providing efficient resource services, e.g., power management. It also supports system integration, monitoring, as well as system security.

Although middleware is a well-established research area in distributed computing systems, WSN poses new challenges to middleware research. The traditional middleware techniques cannot be applied directly to WSNs. First, most distributed system middleware techniques aim at providing transparency abstractions by hiding the context information but WSN-based applications are usually required to be context-aware. Second, although many mobile computing middleware supports context awareness, their major concern is how to continuously satisfy the interests of individual mobile nodes in the presence of mobility. In contrast, WSN-based systems are data centric reflecting the whole application's interests. Thus, the locations and mobility of the sensor nodes should be handled by WSN middleware in a different way. For example, a node moving away from a phenomenon may choose to hand off the monitoring responsibility to a nearby node. Also, WSNs mostly use attribute-based addressing rather than relying on network-wide unique node addresses. Third, data aggregation in intermediate nodes of the forwarding path is desirable in a WSN but no such kind of support is provided in traditional distributed system middleware because of the end-to-end paradigm used. Finally, WSN requires the middleware to be light weight for implementation in sensor nodes with limited processing and energy resources. WSNs also have new requirements on hardware (e.g. various sensors and computing nodes), operating systems and routing protocols, as well as the applications.

7.1 MIDDLEWARE FOR WSN APPROACHES

Wireless sensor networks consist of a large number of small scale nodes capable of limited computation, wireless communication and sensing. WSN supports a wide range of applications like object tracking, infrastructure monitoring, habitat monitoring, battle field monitoring, health care monitoring etc.

Developing applications for WSN is a tedious job as the application developers have to meet considerable number of constraints due to the rigid integration of sensor nodes to the physical world. Designing a middleware is a novel approach for addressing these constraints wherein the middleware can act as binding software between applications and operating systems (OS).

The necessity of designing a middleware software for WSN is to bridge the gap between the high level requirements from applications and the complexity of the operations in the underlying network, there are some other issues which could be well addressed by designing a middleware and they are listed as follow:

1. Resource management at the middleware level is more easy and flexible compared to the OS layer level and application layer level because resource management at the OS level becomes platform dependent and will not be common for all applications if it is at the application layer level.
2. Adding security features is also more appropriate at the middleware level supporting multiple applications.
3. Integration of a WSN with other networks is possible with a middleware.
4. Middleware can provide run time environment for supporting and co coordinating multiple applications.

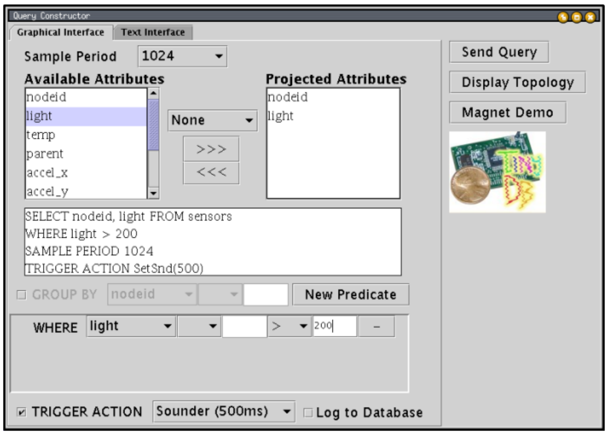
Most of the middleware developed for WSN are inspired by traditional middleware for conventional computer networks or in abstractions from other paradigms, like data-bases.

This characteristic makes possible classify them according to the paradigm that they follow, and they are presented this way as follows.

Database-Inspired Solutions

In the first group, database-inspired approach, the selected middleware were the classical COUGAR and TinyDB.

COUGAR and TinyDB are designed for use in relatively simple data collection applications, only supporting very simple in network selection and aggregation functions based on simple arithmetic operations. Both have a SQL-like query language with support to temporal and data streaming.



Sample Period: 1024

Available Attributes: nodeid, light, temp, parent, accel_x, accel_y

None

Projected Attributes: nodeid, light

```
SELECT nodeid, light FROM sensors
WHERE light > 200
SAMPLE PERIOD 1024
TRIGGER ACTION SetSnd(500)
```

□ GROUP BY nodeid

WHERE light > 200

☑ TRIGGER ACTION Sounder (500ms) □ Log to Database

TinyDB is more sophisticated than COUGAR in the energy saving by calculating the frequency of the sampling to answer queries and also by the use of a routing structure that helps the nodes to route in a energy-efficiently way. COUGAR uses a schema of leader nodes to aggregate data in the way back of data to respond queries.

A key limitation in this two middleware is the assumption that sensor nodes are largely homogenous. The data types/relations that will be used at every node must be agreed in advance. This is acceptable in a small size sensor network; however it represents a great limitation for networks with more nodes and represents really a problem if a graceful evolution is required. A noteworthy limitation of the sensor database approach is that they are not prepared to support rich sensor types, as surveillance cameras with image processing. Another remarkable comment is about the use of a SQL-like languages not being appropriate as sensor data capture observations and not facts.

Event-Based Solutions

Event-based approach is based on the idea of publish/subscribe, allowing a decoupling of event producers and subscribers, however it supports additional operations related to time and spatial conditions. The main drawback in the approach is the complexity involved in its implementation. Another problem is the assumption that all sensors report accurate data and that the set of sensors is homogeneous. DSWare introduced some handling regarding uncertainty of events by the notion of confidence when looking at event correlations.

Mires provides a more traditional publish/subscribe solution designed to run on TinyOS. In its architecture, sensors advertise the type of data that they can provide, while applications are able to select among these data, those in which they are interested. Sensors so publish the data to applications according to the subscriptions. However, no detail about how this protocol is implemented is provided.

7.2 REFERENCE MODEL OF WSN MIDDLEWARE

As shown in Figure 1, a complete WSN-middleware solution should include four major components: programming abstractions, system services, runtime support, and QoS mechanisms. Programming abstractions define the interface of the middleware to the application programmer. System services provide implementations to achieve the abstractions. Runtime support serves as an extension of the embedded operating system to support the middleware services. QoS mechanisms define the QoS constrains of the system.

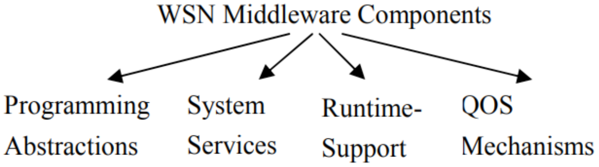


Figure 1. Major components of WSN middleware.

By analyzing the requirements of WSN-based applications and the characteristics of WSNs, we propose a reference framework, shown in Figure 2, to describe the organization and relationships of the above components. It should be mentioned that it is not necessary for a specific WSN-middleware to include all the components. Also, functions of several components may be combined together and implemented as one component.

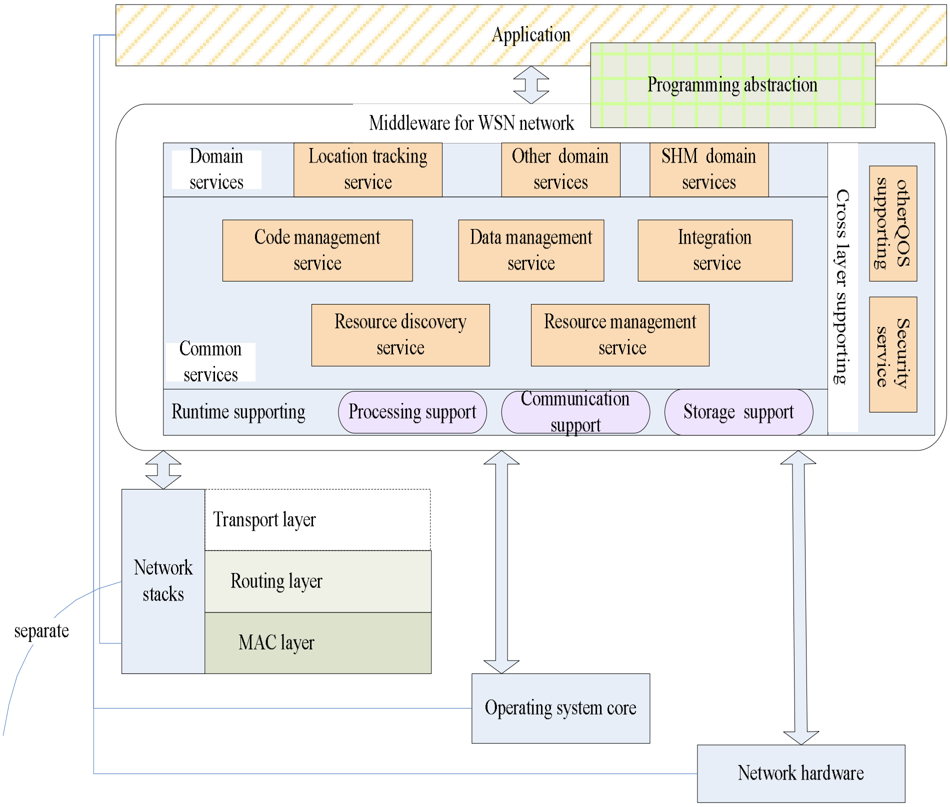


Figure 2. Reference model of WSN middleware.

In the deployment, the functions of WSN-middleware can be distributed to the sensor nodes, the sink nodes, and high level application terminals, as shown in Figure 3. The distributed middleware components located in different nodes of the network communicate with each other to achieve some common goals.

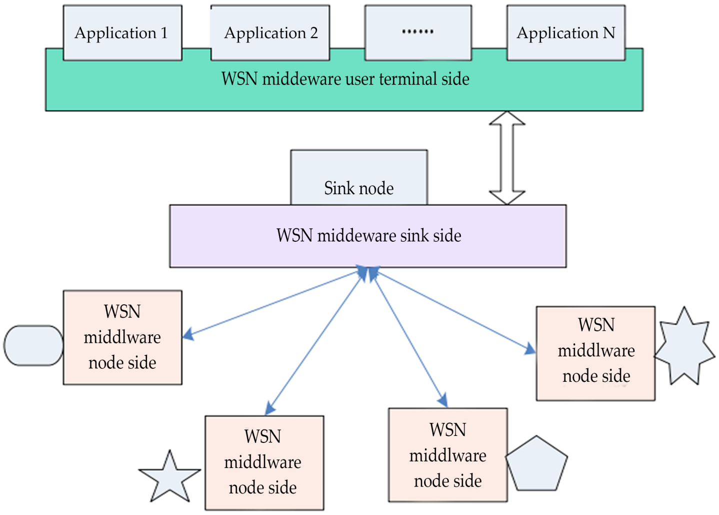


Figure 3. System architecture of WSN middleware.

7.2.1 Programming Abstractions

Programming abstractions is the foundation of WSN-middleware. It provides the high-level programming interfaces to the application programmer which separate the development of WSN based applications from the operations in the underlying WSN infrastructures. It also provides the basis of developing the desirable middleware services. Three aspects are involved when developing the programming abstractions: abstraction level, programming paradigm, and interface type.

Abstraction Level refers to how the application programmer views the system. Node level abstraction abstracts the WSN as a distributed system consisting of a collection of sensor nodes, and provides the programmer the support for programming the individual sensor nodes for their actions and cooperation. System level abstraction abstracts the WSN as a single virtual system and allows the programmer to express a single centralized program (global behavior) into subprograms that can execute on local nodes (nodal behavior), leaving only a small set of programming primitives for the programmer while making transparent the low-level concerns such as the distributed code generation, remote data access and management, and inter-node program flow coordination. Generally speaking, node level abstraction facilitates the development of applications with more flexibility and energy saving, and less communication and interpretation overhead. On the other hand, system level abstraction is easier to use because nodal behaviors can be generated automatically so the programmer can concentrate on the network-level actions, without worrying about how the sensor nodes collaborate with each other to perform the assigned task.

Programming paradigm refers to the model of programming the applications. It is often dependent on the applications. WSN applications can be classified in two dimensions: application Data collection feature and application dynamic feature. Data collections can be continuous, event-driven, or query-based. Application can be totally static and has some mobility characteristic, such as mobile target or mobile sink. Correspondingly, for different applications, WSN middleware may use different programming paradigms, such as database, mobile agent, and Publish/Subscribe (Pub/Sub). For example, the data base paradigm is often used for query-based data collection, while the Pub/Sub paradigm can be a good choice for event-driven applications. Mobile agent paradigm may be a choice for tracking mobile target applications.

Interface type refers to the style of the programming interface. As a matter of fact, programming abstraction is embodied as the programming interface. Descriptive interfaces provide SQL-like languages for data query, Rule-based declarative languages for command execution, or XML-based specification files for context configuration. On the contrary, imperative interfaces provide imperative programming languages for writing the code to interact with the WSN network. Descriptive interfaces usually require the interpretation of the queries and thus consume more resources, while imperative interfaces require the programmer to specify the logic of execution, and are more flexible but more difficult to use.

The consideration of adopting a particular abstraction level and selecting an appropriate programming paradigm and applicable interface depends on the specific application requirements and the underling WSN infrastructure. Middleware providing similar paradigms may share the implementation techniques. For example, the database-based paradigm is usually implemented with a descriptive interface, while the event-driven paradigm can be implemented either with an imperative interface by providing the handlers to be recalled or with a descriptive interface by providing an event description scheme.

7.2.2 System services

System services embody the functionalities and form the core of WSN-middleware. They are exposed to the application programmer through the abstraction interface, and provide the support for application deployment, execution, as well as sensor and network management. We classify the system services into two broad categories: common services and domain services.

Common services are the basic services shared by all WSN applications. They help manage the application information and the WSN infrastructure.

The functionalities provided by the common services include:

- *Code management*: responsible of code migrating and code updating in a deployed network,
- *Data management*: responsible of data acquisition, data storage, data synchronization, data analysis, and data mining,
- *Resource discovery*: responsible of discovering newly joined sensor nodes and detecting nodes becoming inaccessible either as a result of mobility or loss of battery power,
- *Resource management*: responsible of managing the node resources (e.g. energy, memory, A/D device, communication module) and network resource (e.g. topology, routing, system time),
- *Integration*: responsible of integrating WSN and its applications into other networks, such as the Internet and Grid, for broader use.

Domain services facilitate the development of applications in a specific domain. They can make use of the common services and add application oriented functions to provide domain specific services. For example, EnviroTrack is a WSN middleware that support environmental Target tracking. Impala is a middleware for the ZetbraBet project, a wildlife monitoring project. It has two layers: the upper layer contains the application specific protocols and functions, and the lower layer contains the common services such as code management. WSN-SHM middleware is designed for developing structural health monitoring applications which have the requirements of high frequency sampling and high resource consumption.

7.2.3 Runtime Support

Runtime support provides the underling execution environment of applications and can be seen as an extension of the embedded operating system which provides functions of scheduling of tasks, inter-process communication (IPC), memory control, and power control in terms of voltage scaling and component activation and inactivation. The need of runtime support in WSN middleware comes from the facts that the hardware and firmware of the sensor nodes may not always provide enough support for the implementation of the middleware services.

The functionalities of the runtime support in WSN middleware include local processing support, communication support, and storage support. More specifically, support is provided for multi-thread processing, smart task scheduling and synchronization of memory access.

Runtime support of WSN-middleware is always embodied as a virtual machine over a specific embedded operating system.

7.2.4 QoS Mechanism

Quality of Service (QoS) mechanisms is an advanced feature of WSN-middleware. Providing QoS support in WSN is still an open issue for research. QoS features are always cross layers and cross components, embodied in various functional services. For example, the data management service is required to be reliable and of high accuracy.

Typical parameters for expressing QoS of WSN network infrastructure include message delay, jitter, and loss, network bandwidth, throughput, and latency. Typical parameters for expressing QoS of WSN applications include data accuracy, aggregation delay, coverage, and system life time. Middleware acts as a broker between the applications and the network infrastructure. QoS support may translate and control the QoS metrics between the application level and the network level. If the QoS requirements from an application are not feasible to fulfill in the network, the middleware may negotiate a new QoS guarantee with both the application and the network. QoS support may also provide the implementation framework for simplifying the QoS-aware WSN application development using QoS assurance algorithms.

7.3 MIDDLEWARE SYSTEM SERVICES

Middleware systems are comprised of abstractions and services to facilitate the design, development, integration and deployment of distributed applications in heterogeneous networking environments.

7.3.1 Code management

A WSN application consists of pieces of code that execute on the sensor nodes. Code management provides services for code deployment, i.e., allocation and migration of code to sensor nodes. Code allocation determines a set of sensor nodes, on which the execution will be activated. Code migration transfers the code on a sensor node to another node. It not only helps conveniently re-task the network for network reprogramming (code updating), but also enables the data computation elements of an application to be re-located. Code can migrate to the nodes close to the area where relatively large amounts of data are collected, enabling potentially high energy saving, or migrate with the mobile phenomena. For example, the code of an application for fire alarm can be migrated from node to node along the path of fire spread.

Generally speaking, implementation of code allocation involves with checking conditions using comparisons. In SINA code allocation is implemented in a sensor execution environment (SEE), which compares SQTL script parameters with the attributes of sensor nodes and executes the script only if there is a match. In Cougar, code allocation is implemented by a query optimizer that determines the energy-efficient query routes. Code allocation services implemented by a query optimizer has good expressivity but brings network load, while the SEE approach has limited expressivity but good scalability. Another promising approach, as used in MiLAN, is to apply application-level QoS to control the code allocation in configuration adaptation. The approach enables the adaptation of the application operations based on the current application requirements, which can be adjusted depending on the output of the application itself. In this way the code allocation is adaptive to the changing conditions. However, the technique used in MiLAN requires a centralized control.

Code migration can be implemented at not only the middleware layer but also in the underlying embedded operating systems, as in BerthaOS and MagnetOS. However, because WSN OS does not support code interpretation, code migration implemented at the OS level is error prone and subject to malicious attacks.

At the middleware level, most techniques for task migration rely on the use of mobile code, moving the code to the data origins to process the data locally. Current implementations include code migration through mobile code and mobile Java object. An example of mobile code is mobile agent, which is an execution thread encapsulating the code as well as the state and data. Mobile agent makes migration decisions autonomously. The key of this approach is to make the application as modular as possible to facilitate their injection and distribution through the network. However, the nature of mobile agent code does not allow hardware het erogeneity. So, this approach is implemented on top of a VM for platform independency.

There is a trade-off between the complexity of the interpreter running on the nodes and the complexity of mobile code. Code migration services implemented by mobile code with TCL and SQTL have the advantages of small size and high dynamicity, but are suffered from the complexity in specification and high communication cost. Implementation based on mobile agent and mobile Java objects have good salability but high resource consumption. Code migration is very resource dissipative and should be used only when necessary.

To get more insights of the code management services, we take Agilla as an example of the implementation techniques. Agilla is a mobile agent based WSN middleware. The idea behind Agilla is to initially deploy a network without any application installed. Agents that implement the application behavior can later be injected, effectively reprogramming the network. Agilla marks the first time that multiple mobile agents and tuple spaces are used in a unified framework for WSNs.

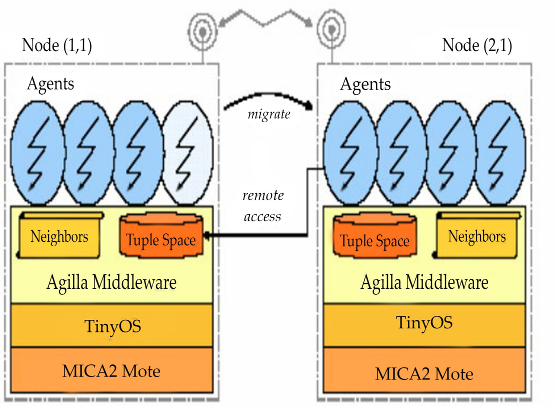


Figure 4. Agilla system model.

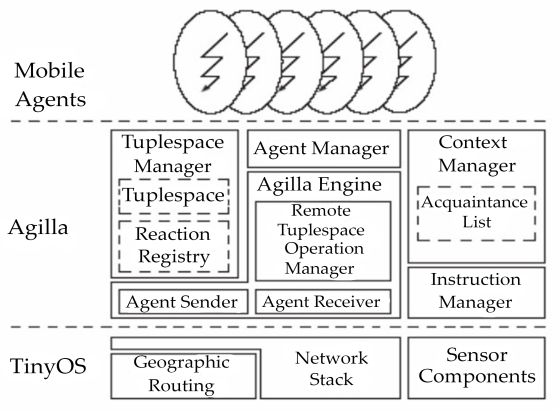


Figure 5. Agilla middleware architecture.

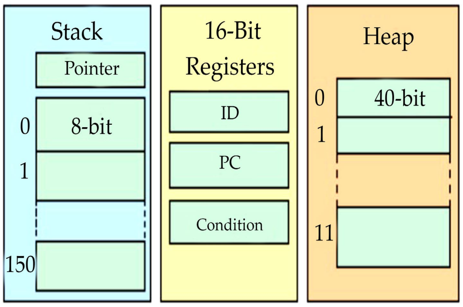


Figure 6. Agilla mobile agent architecture.

The Agilla system model is shown in Figure 4. Each sensor node supports multiple agents, and maintains a tuple space and a neighbor list. The tuple space is local and shared by the agents residing on the node. Special instructions are provided to allow the agents to remotely access another node's tuple space. The neighbor list contains the addresses of all the one-hop nodes. Agents can migrate carrying their code and state, but not their own tuple spaces.

Figure 5 shows the middleware architecture of Agilla. The tuple space manger implements the tuple space operations (e.g., out, inp and rdp) and reactions, and manages the contents of the local tuple space and reaction registry. The agent manager maintains each agent's context. It is responsible of allocating memory for an agent when it arrives and de-allocating it when the agent leaves or dies. The context manager determines the node's location as well as that of its neighbors. Instruction manager and Agilla engine provide runtime support. Instruction Manager is responsible of dynamic memory allocation, retrieving the next instruction to execute, and packing up the agent's code into the minimal number of messages. The Agilla engine controls the concurrent execution of all the agents on a sensor node.

Figure 6 shows the agent architecture. An agent consists of a stack, heap, and various registers. The heap is a random-access storage area that allows an agent to store variables. The registers contain the agent's ID, program counter (PC), and the condition code. The agent ID is unique to each agent and is maintained across migration operations. A cloned agent is assigned a new ID. The PC contains the address of the next instruction, and is used by the code manager to fetch the next instruction. When a reaction fires, the reaction manager changes the PC to point to the first instruction of the reaction's code. To allow an agent to resume execution from where it was when the reaction fired, the original PC is stored on the stack. The condition code records the execution status.

With regards to code allocation, Agilla use a reaction approach. Reactions are added to the tuple spaces, allowing an agent to tell Agilla that it is interested in tuples that match a particular template. The tuple space manager remembers the reactions registered by each agent by storing them within the reaction registry. Whenever a tuple is inserted, the registry is checked to see whether the new tuple matches a reaction's template. If so, the tuple space manager notifies the agent manager, which updates the agent's program counter to execute the reaction's code.

Code migration is implemented by moving or cloning an agent from one node to another. The tuple space manager packages up all reactions registered by an agent so they can be transferred along with the agent. When an agent moves, it carries its state and code and resumes executing on the new node. When it clones, it copies its state and code to another node and resumes executing on both the old and new nodes. The multi-hop migration is handled by the middleware and is transparent to the user.

7.3.2 Data Management

WSN applications are data centric. Here, data refers mainly to the sensed data. Sometimes it also refers to the network infrastructure information interested by the applications. Data management in WSN middleware provides services to applications for data acquisition, data processing, and data storage. The approaches to implementing the data management services depend much on the application data model.

Data acquisition

Data acquisition is an essential service for WSN applications, responsible of delivering the relevant and accurate data required by the application.

For the event based data model, data acquisition support is focused on the event definition, event register/cancel, event detection and event delivery. The application specifies the interest in certain state changes of the data. Upon detecting such an event, the middleware will help send event notification to interested applications. TinyDB, DSware, Mires, and Impala all support event-based data acquisition. DSware also supports compound event detection.

A typical approach to implementing event-based data acquisition is the Pub/Sub paradigm, which has two advantages in supporting event based data acquisition. First, it supports asynchronous communication. Second, it facilitates message exchanging between the sensor nodes and the sink node. The basic entities of Pub/Sub system are event subscriber and event publisher (sometimes event broker also). From the middleware's point of view, the event subscriber is the sink node and the event publishers are the sensor nodes.

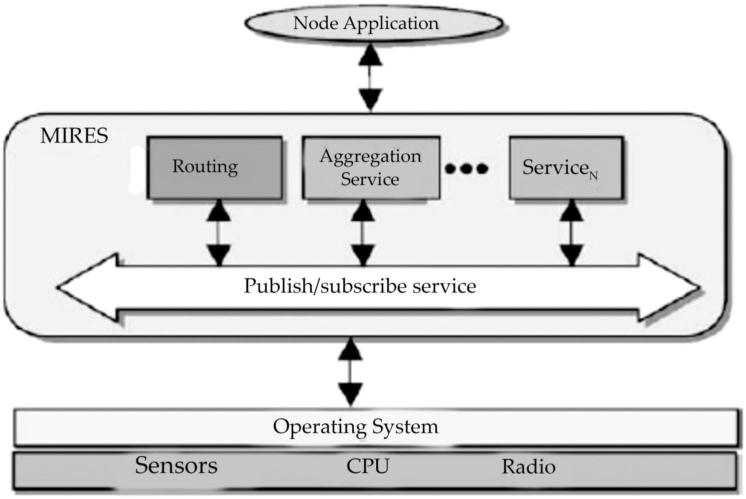


Figure 7. Mire's architecture.

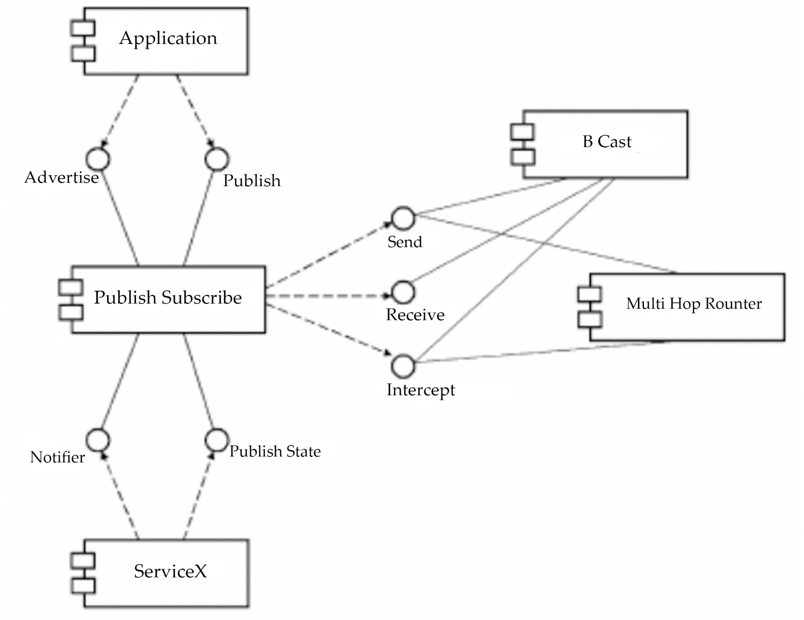


Figure 8. Mire's Pub/Sub component.

As an example of the Pub/Sub approach, let us have a look of Mires. Figure 7 and 8 show Mires' architecture and its Pub/Sub component structure respectively. Mire includes a core component, namely the Pub/Sub service, and some additional services. The communication between the sensor nodes consists of three phases. Initially, the sensor nodes in the network advertise their available topics (e.g., temperature and humidity) collected from the local sensors. Next, the advertised messages are routed to the sink node using a multi-hop routing algorithm. A user application connected to the sink node is able to select (i.e., subscribe) the desired advertised topics to be monitored. Finally, subscribe messages are broadcasted down to the network nodes. After receiving the subscribed topics, the sensor nodes are able to publish their collected data to the network. The Pub/Sub service maintains the topic list and the subscribing applications so as to marshal the right topic to the related application. In Mires, only the messages referring to the subscribed topics are sent, hence reducing the number of transmissions and energy consumption.

For query-based data model, data acquisition support is focused on the query processing model and methods. Middleware for query-based data model usually use a declarative interface, with global level abstraction and database programming model. Example systems are TinyDB, Cougar, and SensorWare. They leverage the techniques used in the traditional database system to implement data acquisition services, e.g., applying distributed query or CACQ (continuously adaptive continuous queries over streams)

TinyDB is a good example to illustrate the query-based approach. TinyDB is a query-processing system that extracts information from the data collected by the WSN using the underlying operating system TinyOS. TinyDB maintains a virtual database table, called SENSORS, whose columns contain information such as sensor type, sensor node identifier, and remaining battery power. The programmer can view the values of the SENSORS, and add new rows to it. Consider the following example. A user wants to be reported when the average temperature is above 80° F in any room on the third floor of a building monitored by sensors. The user inputs the following database query along with the rate at which the sensors are to collect the data:

SELECT AVG (temp) FROM sensors

(select rows from Sensors)

WHERE floor = 3

(at the 3rd floor)

GROUP BY room

(rows are grouped by room number)

AVG (temp) > 80F

(only groups with average temperature > 80F)

SAMPLE PERIOD 20 seconds

(perform every 20 seconds—rate of collection)

TinyDB uses a controlled-flooding approach to disseminate the queries throughout the network. The system maintains a routing tree (spanning tree) rooted at the end point (usually the user's physical location). Then, in a decentralized approach, every sensor node has its own query processor that processes and aggregates the sensor data and maintains the routing information. In every period, the parent node closer to the root agrees with its children on a time interval for listening to data from them.

Data processing

Generally speaking, there are three different approaches to support data processing in WSNs. In centralized processing, all the data are collected and then sent to a central node for processing. In node level distributed processing raw data collected in the sensor nodes are pre-processed to obtain partial results, which are then collected by the sink node for further processing to get the final result. In network level distributed processing final results are obtained through both node-level distributed processing and information exchange between the sensor nodes, and between sensor nodes and the sink node. In the extreme case, where every sensor node is involved with data processing, routing, and is aware of the final decision, it becomes completely distributed processing.

Given that the communication cost is much higher than the computation cost at a sensor node, WSN middleware should support in-network distributed data processing service, mostly through. data fusion / aggregation. Although in-network data processing services are also supported at a lower level by some firmware in terms of signal conditioning, and data fusion and data aggregation can also be supported at the MAC and routing layers, middleware support has the following distinctive features:

1) It is more independent of the underlying network protocols, so different strategies can be applied according to different data accuracy requirements from different applications or different network conditions.
2) It facilitates high level data analysis such as feature-based fusion and decision-based fusion.

For event-based data model, data aggregation/fusion can be implemented in separate services. An example is the aggregation service in Mires. In Mires, data aggregation is implemented in separate modules for functions such as AVG and SUM. The aggregation is executed by an "Aggregate Use" module that carries out an activity of de-multiplexing, passing requests for the correct aggregation module in accordance to its identifier. This way, the flexibility to add new aggregation functions is guaranteed, just requiring the creation of a module for the new function and adding the association between the function and an identifier to a configuration file.

In addition, for event-based data model, detecting the event boundary and determining the event area and its center should also be considered in WSN middleware.

For query-based data model, data aggregation/fusion services can be implemented by using the pipelining techniques, as used in TinyDB and SensorWare.

Another data processing service is data calibration for ensuring the synchronization between the sensor nodes. Some applications, e.g., seismographic or building health monitoring, require precise time synchronization among the readings on different sensor nodes. How to achieve time synchronization is an important function of the middleware.

Data storage

There are three approaches to implementing data storage support in WSNs. External storage stores the data in the base station out of the WSN. Local storage stores the data where it is generated, reducing communication but increasing the inquiry cost. Data centric storage provides a tradeoff between the previous two approaches. Data-centric storage is the most popular approach implemented in existing WSN middleware.

Let us look at Data Service Middleware (DSWare) as an example to show the data storage service implementation in WSN-middleware. As shown in Figure 9, DSWare is a specialized layer that implements various data services and, in doing so, provides a database like abstraction to WSN applications. Figure 10 shows the DSWare framework. The event detection component is responsible of providing the data acquisition service. The group management component provides the support for group-based decision and is responsible of data aggregation. The scheduling component schedules the services to all DSWare components with two scheduling options: energy-aware scheduling and real-time scheduling. Here, we focus on the data storage and caching components.

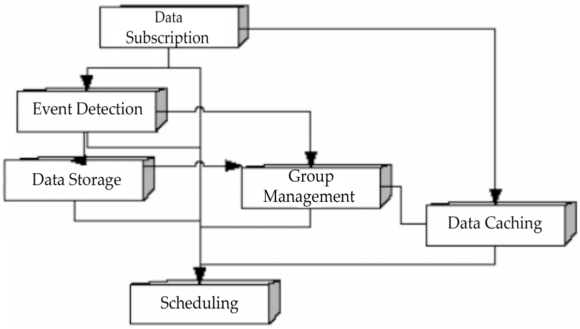


Figure 9. DSWare framework.

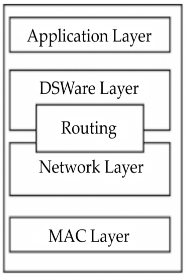


Figure 10. DSWare system model.

The Data Storage component in DSWare stores data according to the semantics associated with the data. It has a data look-up operation and provides fault tolerance should there be node failures. It also has operations for storing correlated data in geographically adjacent regions. This has two advantages: enabling data aggregation and making it possible for the system to perform in-network processing.

To facilitate data look-up, DSWare maps data to physical storage using two levels of hash functions. At the first level, the hash function maps a key, which is a unique identifier assigned to each data type, to a logical storage node in the overlay network. As a result of this operation, the storage nodes form a hierarchy at this level. The second level involves the mapping of single logical node to multiple physical nodes such that a base station performing a query operation has the data fetched from one of the physical locations. There is a big risk in mapping a given data type to a single node as this data could be lost as a result of node failure. Furthermore, mapping data to a single node in the sensor network causes bursts of traffic to the node which may lead to collision and higher rate power consumption. DSWare uses replication to store data in multiple physical sensor nodes that can be mapped onto a single logical node. Load balancing is achieved since queries can be directed to any one of the physical nodes and the lifetime of individual nodes is prolonged since power consumption is substantially reduced. With replication of data amongst multiple nodes come consistency issues. DSWare adopts "weak consistency "to avoid peak time traffic since only the newest data amongst nodes is bound to lack consistency. This new data is propagated to other nodes and the size of inconsistent data is bounded so that replication occurs when the workload in individual nodes is low.

Data Caching in DSWare provides multiple copies of data that are most requested. DSWare spreads the cached data over the network to achieve high availability and faster query execution. A feedback control scheme is used to dynamically decide whether or not copies should reside in frequently queried nodes. The scheme uses various inputs including proportion of periodic queries and average response time from data source to guide the nodes in making decisions about whether or not a copy should be kept. This component also monitors the usage of the copies to decide whether to increase or reduce the number of copies, or move them to a new location.

In conclusion, 1) Data management is an important topic in WSNs. One of the distinguished features that middleware offer in data management is the appropriate abstraction of data structure and operation. Without this abstraction, the developer has to manage the heterogeneous data and low level operation in the application. Various exiting data management algorithms can be implemented as reusable and alternative middleware services with certain of parameters. The middleware system can even automatically adjust the service parameters according to its current status. Application specific data management algorithms can be written based on those common data services. This also facilitates the development process. 2) Most existing WSN middleware provide some kind of data management services. However, high level in-network analysis services related to the WSN application domain, e.g. data mining, are not implemented yet and need more attention.

7.3.3 Resource and Information Discovery

Resources in a WSN usually refer to the sensor node hardware resource, e.g. energy, memory, A/D device, and communication module. The resource discovery service returns the data type that a discovered node can provide, the modes in which it can operate, and the transmission power level or residual energy level of a sensor node. On the hand, the information discovery service returns the information about the network topology, the network protocols, and the neighbors and the locations of the discovered nodes. The service can also be used to discover new nodes and find out when nodes become inaccessible as a result of either mobility or loss of battery power. However, many of the service features is not being available in existing WSN middleware yet

Compared to resource discovery in traditional networks which is involved with identifying and locating (relocating) the services and resources in the system, resource and information discovery services in WSN are more difficult to implement due to the lack of unique node ID and the lack of generic service specification, and because the services need to be provided in a power-aware way. Some existing WSN middleware systems adopt service discovery protocols from traditional computer network solutions, e.g., SLP and Bluetooth SDP. MiLAN is an example. Other systems, e.g., TinyLime, use tuple space to implement the resource discovery service. However, these implementations need Unique ID of the resource, but many WSNs are content based without Unique ID for sensor nodes

Although many localization algorithms have been developed for different kinds of systems, for example, Ultrasound, RF, and ultra-wideband, RSSI techniques are used for accurate localization via carefully placed beacons. Few existing WSN middleware has integrated location discovery service. To our opinion, this is mainly because the implementation of this kind of service depends very much on hardware and the underlying environment. For large scale use of WSN for pervasive computing, standard and adaptive location discovery services should be provided.

7.3.4 Resource Management

Resource and information discovery services have two main functions:

1) Providing the underlying network information to applications that are required to be reflective or adaptive (e.g. context-aware),
2) Providing the underlying network information to support adaptive resource management services.

Resource management in WSN middleware is mainly for providing common and reusable services to support the applications that have the requirements of self-organization. Resource management services are usually used for resource configuration at setup time and resource adaptation at runtime, and they are essential to ensure the QoS of WSN.

Resource management at the OS layer is platform-dependent, so changes at this level might affect different resource requirements of the applications running in a sensor node. On the other hand, application-level resource management imposes an extra burden on the application, and adaptation mechanisms developed at this level cannot be reused. In contrast, resource management at the middleware layer has more flexibility. Most existing WSN middleware provide services including cluster service, schedule service, and data routing service. These services are supported by finer granular services such as power level management, transmission level management, etc. These fine granular services should be supported and constrained by the underlying OS, the firmware, and the hardware. Otherwise, it is impossible for the middleware to provide the corresponding services.

The cluster service refers to the cluster member maintenance for layered WSN. For cluster service, many middleware systems, including EnviroTrack, MiLAN, DSWare, AutoSec, and SINA addressed the implementation issues according to different objectives. For examples, EnviroTrack provided the cluster member re-allocation service to re-define the clusters after deployment; the MiLAN and AutoSec provided automatically cluster organization service according to the QoS information getting from network infrastructure and WSN application. SINA and DSWare also provided automatically cluster organization service, but the objectives are to achieve appropriate clusters so as to facilitate the data aggregation process. Except for the above examples, the work reported in provides a function for generic cluster management of sensor nodes. The function arises either in terms of non-functional requirements (e.g., security, reliability) or according to dynamic system conditions. (e.g., power level, connectivity).

The schedule service refers to the node wakeup/sleep scheduling. It is used to reduce the energy consumption by allowing the sensor nodes to be put to sleep and to be taken up according to specific policy. For example, when not being allocated tasks a sensor node can sleep in order to save energy. Implementation of this service may make use of the services, such as sleep scheduling protocols in the MAC layer and CPU voltage scaling in the physical layer.

The data routing service can be implemented in several different ways. Some middleware such as Mate do not provide any specific routing management service, but provide architecture which allows the implementation of arbitrary routing protocols. For systems that provide routing management services, three main approaches can be identified. The first approach is implementing a new higher level routing protocol at the middleware level. An example is MagnetOS that implements a multi-hop routing protocol in a middleware component. The second is maintaining an overlay, and supporting routing mechanism, as well as routing reconfiguration on top of this overlay. For example, Mires makes use of a Pub-Sub mechanism to support the routing management. Owing to the loosely coupled interactions between the nodes in the Pub-Sub paradigm, it is very flexible to provide new kind of data routing implementation. The third approach is implementing a mechanism that allows for switching between different routing protocols, as what is done in Impala, or providing a mechanism that allows for the adaptation of different routing protocols, as what is done in MilAN.

Most existing WSN middleware adopts localized resource management. Policy based management has been shown to be a good approach to supporting the design of self-adaptive resource management Currently, resource management services in existing WSN middleware are tightly coupled with applications and generic resource management services need to be developed.

7.4 INTERNET-SCALE RESOURCE-INTENSIVE SENSOR SERVICES (IRIS)

IRIS is composed of a potentially global collection of sensing agents (SAs) and organizing agents (OAs). SAs collect and process data from their attached webcams or other sensors, while OAs provide facilities for querying recent and historical sensor data. Any Internet connected, PC-class device can play the role of an OA. Less capable PDA-class devices can act as SAs. Continued advances in microprocessing technology enable significant processing power and memory to be encapsulated in smaller and cheaper devices that may act as SAs.

Key features of IRIS include:

- IRIS provides simple APIs for orchestrating the SAs and OAs to collect, collaboratively process and archive sensor data while minimizing network data transfers.
- The user is presented with a logical view of the data as a single XML document, while physically the data is fragmented across any number of host nodes (data transparency).
- IRIS supports a large portion of XPATH, a standard XML query language, for querying the data in the system.
- IRIS handles issues of service discovery, query routing, semantic caching of responses and load balancing in a scalable manner for all services.

7.4.1 The IRIS Architecture

We describe the overall architecture of IRIS, its query processing features, and its caching and data consistency mechanisms.

Architecture

IRIS is composed of a dynamic collection of SAs and OAs. Nodes in the Internet participate as hosts for SAs and OAs by downloading and running IRIS modules. Sensor-based services are deployed by orchestrating a group of OAs dedicated to the service. These OAs are responsible for collecting and organizing the sensor data in a fashion that allows for a particular class of queries to be answered (e.g., queries about parking spaces). The OAs index, archive, aggregate, mine and cache data from the SAs to build a system-wide distributed database for that service. Having separate OA groups for distinct services enables each service to tailor the database schema, caching policies, data consistency mechanisms, and hierarchical indexing to the particular service. This does not restrict the placement of OAs, because multiple OAs can be hosted on the same node.

In contrast, SAs are shared by all services. An SA collects raw sensor data from a number of (possibly different types of) sensors. The types of sensors can range from webcams and microphones to temperature and pressure gauges. The focus of our design is on sensors that produce large volumes of data and require sophisticated processing, such as webcams. SAs with attached webcams include, as part of the IRIS module, Intel's open-source image-processing library. The sensor data is copied into a shared memory segment on the SA, for use by any number of sensor-based services.

OAs upload scripting code to any SA collecting sensor data of interest to the service, basically telling the SA to take its raw sensor feed, perform the specified processing steps, and send the distilled information to the OA. For video feeds, the script consists primarily of calls to the image processing library. Filtering data at the SAs prevents flooding the network with high bandwidth video feeds and is crucial to the scalability of the system. Even compressed video consumes considerable bandwidth, whereas aggressive filtering can reduce 10 minutes of video down to under a kilobyte of data, depending on the service. For example, the Parking Space Finder service distills the video down to a bit vector indicating which spots are empty.

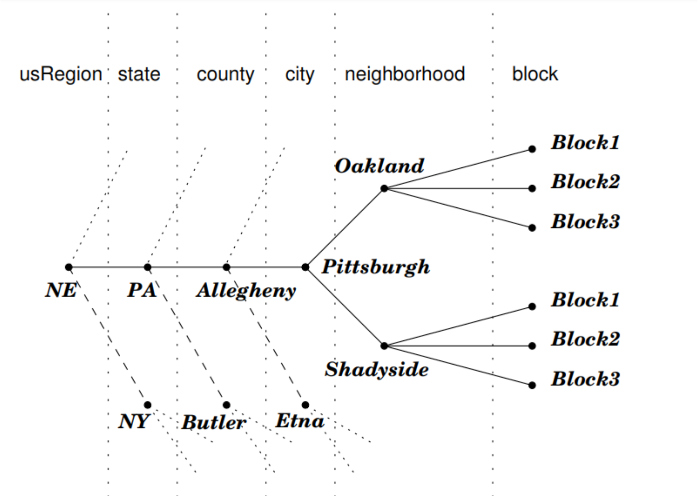


Figure 11. OA Hierarchy.

Query Processing

Central to IRIS is distributed query processing. Data is stored in XML databases associated with each OA. We envision a rich and evolving set of data types, aggregate fields, etc., best captured by self-describing tags – hence XML was a natural choice. Larger objects such as video frames are stored outside the XML databases; this enables inter-service sharing, as well as more efficient image and query processing.

Data for a particular service is organized hierarchically, with each OA owning a part of the hierarchy. An OA may also cache data from one or more of its descendants. A common hierarchy for OAs is geographic, because each sensor feed is fundamentally tied to a particular geographic location. Figure 11, for example, shows a geographical hierarchy for Parking Space Finder.

In IRIS, a query from a user anywhere in the world is first routed to its starting point. But how do we find the starting point OA, given the large number of OAs and the dynamic mapping of OAs to host machines? Our solution is to have DNS-style names for OAs that can be constructed from the queries themselves, to create a DNS server hierarchy identical to the OA hierarchy, and to use DNS lookups to determine the IP addresses of the desired OAs. For our example query, we construct the DNS-style name pittsburgh.allegheny.pa.ne.parking.intel-iris.net, perform a DNS lookup to get the IP address of the Pittsburgh OA, and route the query there.

Upon receiving a query, the starting point OA queries its local database and cache, and evaluates the result. If necessary, it gathers missing data by sending subqueries to its children OAs, who may recursively query their children, and so on. Finally the answers from the children are combined and the result is sent back to the user. Note that the children IP addresses are found using the same DNS-style approach, with most lookups served by the local host. The key technical challenge overcome in our approach is how to efficiently and correctly detect, for general XPATH queries, what parts of a query answer are missing from the local database, and where to find the missing parts.

XPATH queries supported. In our current prototype, we take the common approach of viewing an XML document as unordered, in that we ignore any ordering based solely on the linearization of the hierarchy into a sequential document. For example, although siblings may appear in the document in a particular order, we assume that siblings are unordered, as this matches our data model. Thus we focus on the unordered fragment of XPATH, ignoring the few operators such as position() or axes like following-siblings that are inappropriate for unordered data. We support the entire unordered fragment of XPATH 1.0.

Partial-Match Caching and Data Consistency

An OA may cache query result data from other OAs. Subsequent queries may use this cached data, even if the new query is not an exact match for the original query. For example, the query may use data for Oakland cached at the Pittsburgh OA, even though this data is only a partial match for the new query. Similarly, if distinct Oakland and Shadyside queries result in the data being cached at Pittsburgh, the query may use the merged cached data to immediately return an answer.

Due to delays in the network and the use of cached data, answers returned to users will not reflect the absolutely most recent data. Instead, queries specify a consistency criteria indicating a tolerance for stale data (or other types of approximation). For example, when heading towards a destination, it suffices to have a general idea of parking space availability. However, when arriving near the destination, exact spaces are desired. We store timestamps along with the data, so that an XPATH query specifying a tolerance is automatically routed to the data of appropriate freshness. In particular, each query will take advantage of cached data only if the data is sufficiently fresh.

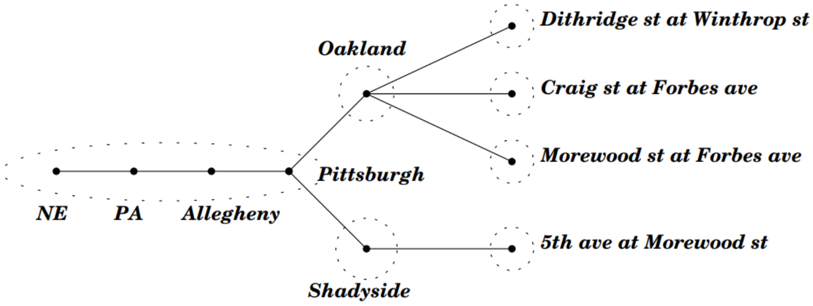


Figure 12. The hierarchy used in the demonstration and the mapping of the hierarchy onto the OAs.

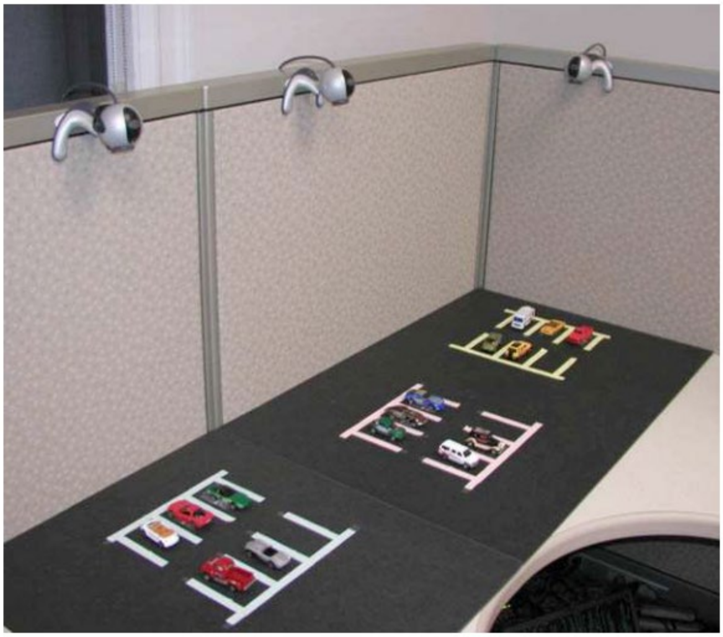


Figure 13. Webcams monitoring toy parking lots.

7.4.2 A Parking Space Finder Service

The service that we demonstrate in this demo is that of a parking space finder. This service utilizes webcams that are monitoring parking spaces to gather information about the availability of the parking spaces.

Sensing Agents

We use 4 cameras that are monitoring toy parking spaces set up as part of our demo. These cameras are attached to 1.6 GHz laptop machines that process the video feed, and perform image processing to decide whether a parking spot is full or empty. Figure 14 shows the actual locations of the four parking lots that we simulate using the above setup. These are parking lots near Intel Research Pittsburgh.

Organizing Agents

The organizing agents that we use for this demo are 7 PCs scattered throughout the Intel research labs. We show the part of the hierarchy that is used in this demonstration. This logical hierarchy is mapped onto the 7 machines as follows:

(1) The four blocks corresponding to the parking lots are mapped onto one OA each,

(2) The two neighborhoods, Oakland and Shadyside, are mapped onto one OA each, and

(3) The rest of the nodes in the hierarchy are mapped onto one OA.

Web Frontend

The web frontend for this service essentially presents the user with a form that the user can fill out to specify her destination, and also other constraints that she might have (e.g., that the parking spot must be covered). Currently, we only allow the user to pick from 5 destinations near the parking lots using a dropdown menu. Once the user specifies her criteria and submits the query, the frontend finds the nearest available parking spot that satisfies the user’s constraints using IRIS, and then uses Yahoo Maps Service to find driving directions to that parking space from the user’s current location. These driving directions are then displayed to the user.

The driving directions are continuously updated as the user drives towards the destination, if the availability of the parking spot changes, or if a closer parking spot satisfying the constraint is available. We envision that a car navigation system will repeatadly and periodically ask the query as the user nears the destination. Lacking that, we currently simulate such a behavior by resubmitting the query periodically by assuming that the user has reached the next intersection along the route.

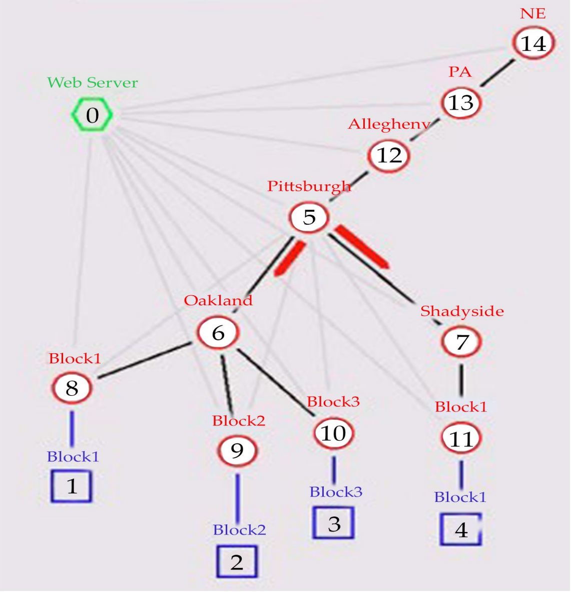


Figure 14. A modified version of NAM is used to show the messages during a query execution.

Logging and replaying messages

We also demonstrate a mechanism that we have built for logging and replaying the messages exchanged by the web frontend and by the OAs. The collected log information during execution of a query is used to lazily replay the messages that were sent during the execution of the query. We use the NAM network simulator to show these messages. NAM is part of the popular open-source network simulator, ns, with a graphical display that shows the configuration of the network under consideration, and uses animation to show messages being communicated in the network.

A series of XPATH queries of increasing complexity are used to demonstrate visually various aspects of our system such as routing to the starting point, recursive query processing, partial-match caching, and query-based consistency.

7.5 MILAN: MIDDLEWARE LINKING APPLICATIONS AND NETWORKS

For several decades, distributed computing has been both an enabling and a challenging environment in which to build applications. Initially, the major difficulty in implementing such systems was simply exchanging data across distances and among heterogeneous components. Today these problems are essentially solved, and research is turning its focus to higher level concerns such as improved fault tolerance through replication, optimal data access via distributed object placement, and methods of enabling high level communication abstractions such as event dispatching and remote invocation. The end result of this research into distributed systems is an expanding set of middleware platforms that reside above the operating system and below the application, abstracting lower level functionality such as network connectivity and providing an abstract coordination interface to the application programmer.

Often in distributed systems, the nature of the network has a major impact on the application performance. This is especially true when the underlying network is primarily wireless and/or the system endpoints are mobile and constrained by battery lifetime. In these environments, local network connectivity changes over time and must be closely monitored and managed to best serve the needs of the applications. However, multiple applications utilizing the network may have different criteria for how to best utilize the available resources. For example, consider a security application and an entertainment application accessing both audio and low-quality video from two different wireless sources. Suppose a new node joins the network offering high-quality video, but the network cannot support the transmission of all three streams. The goal of the security application may be to receive the high-quality video stream, while the goal of the entertainment application may be to keep the low-quality video and audio streams. This creates two problems that cannot be easily solved using conventional middleware systems: how should the needs of each application that utilizes the distributed network resources be implemented, and how should the conflict created by the different needs of the applications be resolved?

Generally, the scenario space we target is one with many low-level devices offering services in which some fundamental resource limitation is pushed, e.g., network bandwidth is exceeded and/or component energy is scarce. This brings out two core issues of distributed systems that are not addressed in conventional middleware. First, rather than an application reacting to the changing network environment, it is important for the application to actually control the environment to optimize its performance. Second, if there are multiple applications sharing a common set of network components, it is important to develop a method of resolving conflicting needs that maximizes an overall utility. While it is possible to build applications to address these issues in each unique instance, a properly designed middleware system will both speed the development of multiple applications in similar domains, as well as increase the dependability and reliability of applications by reusing the set of well-tested management components of the middleware.

The uniqueness of our approach is the integration of network control into the middleware, enabling application-directed network reconfiguration. For this reason, we call our system Milan: Middleware Linking Applications and Networks. From the application, Milan receives a set of performance specifications with respect to different system components as well as information specifying how different applications should interact. From the network, Milan monitors for available components and overall resources such as power consumption and bandwidth. Combining this information in some optimal manner, Milan continuously adapts the network configuration to best meet the applications' needs and balance performance for cost.

7.5.1 Application Scenarios

Although wireless networks are becoming common, the range and types of applications for wireless networks largely follow those of applications on wired networks, namely, setting up point-to-point connections between end-hosts. However, new application scenarios are being proposed that exploit some of the unique properties of wireless networks and distributed systems. For example, in sensor networks, the components typically cooperate to accomplish a common goal (e.g., monitoring a region), rather than compete for available resources. In multimedia applications, the components can take advantage of the broadcast nature of radio technology in order to more efficiently distribute information to multiple recipients, rather than create independent connections. In applications of this new style, the applications themselves should dictate which nodes are active and how they access the channel.

7.5.2 Environmental Surveillance

Consider an environment where multiple sensors (e.g., acoustic, seismic, video) are distributed throughout an area such as a battlefield. A surveillance application can be designed on top of this sensor network to provide information to an end-user about the environment. Suppose, however, that the channel linking all the sensors to the application software cannot handle the transmission of all the sensors' data to the end-user, due to bandwidth constraints of the wireless channel or energy constraints of the individual sensors. Based on data obtained from the sensors that are active at any time, the application knows which sensors' data it needs to maximize performance. For example, suppose initially one sensor in every square meter is sending data to the application and all other sensors are inactive. If the application discovers, through analyzing the data from the active sensors, that an event of interest is occurring in part of the area being surveyed, it needs the ability to control and activate other sensors in that area in order to maximize performance.

To accomplish this with today's technology, the application would need to track the location of all of the sensors, individually enable and disable sensors based on the collected data, as well as ensure that the enabled sensors do not send data at rates that are in excess of the network resources. We believe that through a well-defined API, an application should dynamically declare its needs (e.g., information from high activity areas) with respect to the low-level components (e.g., sensors), and the middleware should assume the task of managing the sensor components, thereby allowing independence of the application from whatever network it executes on top of.

7.5.3 Virtual Campus

Many ubiquitous computing efforts are focused on supporting university students and faculty with computer-based applications. These include digital whiteboards, note sharing, exam taking, viewing live and taped lectures, support for collaborative projects, etc. Many times, these applications exploit wireless connections among various battery powered devices. Our interest comes not from the design of the applications, but from their interactions on a common network. For example, if one student is watching and listening to a taped lecture (e.g., using an 802.11 WLAN) and a second student listening to a live lecture moves into the same area and must share the bandwidth, the network needs to be reconfigured to support both students' applications. Possible solutions include dropping the video but retaining the full quality audio for both students or delaying the taped lecture until resources become available. These system resource allocation choices require a careful balance between the user characteristics (e.g., students vs. faculty), the application characteristics (e.g., live vs. taped), and the current network resources

Supporting the seamless integration of multiple applications and managing the network constraints is not easy with existing middleware platforms. We propose that distributed applications be built independently, but augmented with a minimal set of hooks that allow them to specify their needs to the middleware (e.g., first reduce video quality, but if the reduced quality video cannot be supported, then drop the video but keep the audio, otherwise pause the lecture, etc.). With these hooks and user-supplied utility metrics specifying the desired interaction of different applications, the middleware can manipulate the evolving system to best support faculty and students and the various applications running in the virtual campus.

7.5.4 Milan Solution Strategy

To support applications that need to trade performance for energy cost, we are developing a new middleware, named Milan (Middleware Linking Applications and Networks), which accepts performance needs from the application, monitors network conditions, and optimizes the network configuration on behalf of the application. Our approach is based on a new technique of representing the application needs as a specialized graph, designing network feasibility templates for various underlying networks (e.g., Bluetooth and 802.11), and constantly analyzing current conditions and affecting the network to balance application utility and energy cost.

Unlike traditional middleware that sits between the application and the operating system, Milan, which intends to control the network, has an architecture that extends into the network protocol stack as shown in Figure 15. The interface to the application and the low-level components allows the application to provide a graph specifying the utilities of the low-level components that may be available over time, as well as a mechanism for Milan to control the low-level components. As Milan is intended to sit on top of multiple physical networks, an abstraction layer converts Milan commands to protocol-specific commands that are passed through the usual network protocol stack.

Initial design explores the tradeoff between performance and network cost in the context of a single, centralized application that takes data from multiple distributed sensors. Specifically, we use the heart monitor as a test application, where the utility of the application is measured in terms of reliability of measuring certain variables.

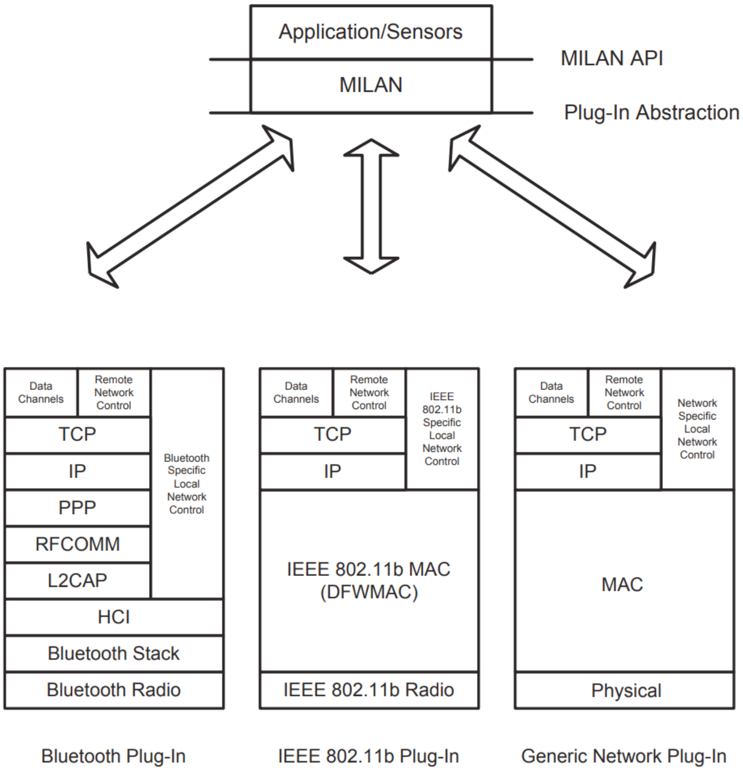


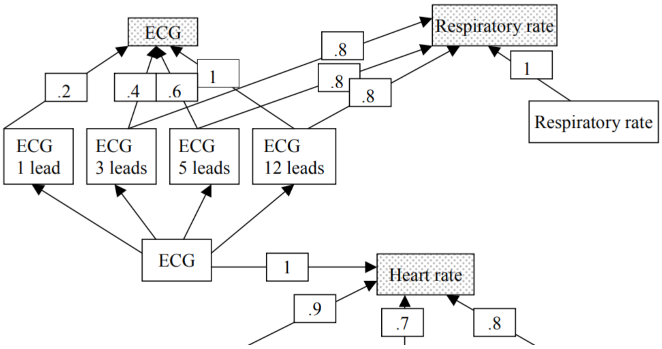
Figure 15. Milan protocol stack.

Application Performance

Consider the heart monitor application, where several variables must be calculated based on data from sensors. For example, the heart rate, respiratory rate, blood oxygen level, blood flow, blood pressure, ECG signal, and position and activity of the person being monitored are all important variables in determining if the heart is healthy or developing problems. If this application relies on a single sensor, for example a blood pressure sensor, it would have a certain reliability in characterizing each of the above-stated variables. For example, a blood pressure sensor directly measures blood pressure and therefore provides 100% reliability in determining this variable, but it indirectly measures variables like heart rate and blood flow and therefore provides less than 100% reliability in determining these parameters. The reliability of these variables would be improved by including data from additional sensors such as ECG, heart rate, and blood oxygen level sensors. Figure 16 shows an example of the different parameters that are important to monitor to determine the condition of the heart, as well as the sensors that can provide direct or indirect measurements of these variables. Lines between a sensor and a variable mean that the sensor can either directly or indirectly measure that variable, and the numbers on the lines represent example reliabilities in using that sensor's data to measure the variable.

Ideally we would want to provide as much data to the heart monitor application as possible to increase its reliability1. There are, however, several drawbacks to such an approach. For one, the lifetime of the system may be reduced if all sensors transmit all the time. Instead, the lifetime may be extended by requiring data from only a subset of available sensors, slightly decreasing the overall reliability but allowing some nodes to save energy. Furthermore, the network capacity may not support the transfer of all of the data for the heart monitor. Thus, it is important to devise an automatic way of dynamically choosing an appropriate subset of sensors to achieve the required reliability while minimizing cost and staying within network resource constraints.

In general, an application knows how it performs given data from different combinations of low-level components. This information must be transmitted to Milan. We propose using a graph-based approach to allow the application to specify this performance information. In the graph, nodes with links emanating from them represent the low-level components (e.g., sensors) and nodes with links ending at them represent the variables the application is trying to measure using sensor data. The weights of the links represent how accurately the sensor data at the tail of the link can determine the variable at the head of the link. Figure 16 shows an example of an application graph for the heart monitoring application.



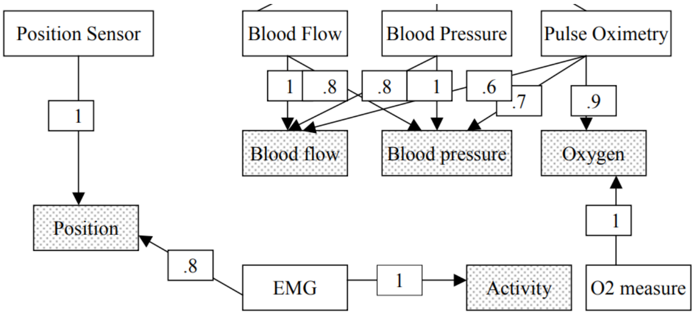


Figure 16. Example application performance graph for a heart monitoring application.

This graph must be extended if sensor data can be fused and the fused data used to determine the variables. In this case, we can add a node for every combination of sensors whose data can be fused. However, this approach adds considerably to the complexity of the graph presented by the application to Milan since the number of fused data sets can grow exponentially. Hence, we are currently exploring other ways to represent fused data.

Finally, it is possible that the contribution of each low-level component to the application performance will change over time with regard to the context of the application. This information must be conveyed to Milan to ensure the highest application performance over time. Initially for the heart monitor application we consider three states for the application corresponding to whether the user is healthy, unhealthy, or in between. The application provides reliability information for each of the three states (e.g., a different reliability graph for each state), and part of the Milan API allows the application to signal a change from one state to another.

In addition to the application graph(s), the application must specify minimum performance (e.g., reliability) bounds that guide Milan in choosing an appropriate set of sensors. Using the application graph that specifies the application variables' reliabilities and the application-specified minimum reliability at a given time, Milan can choose which sensors to use to meet or exceed the application's specified reliability while considering power costs and bandwidth constraints.

For example, suppose the application sends Milan the following request for variable reliabilities:

- Heart rate must be measured with reliability ≥ 0.5
- Blood oxygen level must be measured with reliability ≥ 0.7
- Blood pressure must be measured with reliability ≥ 0.8

SUMMARY

- A WSN consists a collection of sensor nodes and a sink node connected through wireless channels, and can be used to build distributed systems for data collection and processing, covering the functions of on-field signal sensing and processing, in-network data aggregation, and self-organized wireless communication.
- Middleware refers to software and tools that can help hide the complexity and heterogeneity of the underlying hardware and network platforms, ease the management of system resources, and increase the predictability of application executions.
- Wireless sensor networks consist of a large number of small scale nodes capable of limited computation, wireless communication and sensing.
- WSN supports a wide range of applications like object tracking, infrastructure monitoring, habitat monitoring, battle field monitoring, health care monitoring etc.
- Event-based approach is based on the idea of publish/subscribe, allowing a decoupling of event producers and subscribers, however it supports additional operations related to time and spatial conditions.
- Programming abstractions is the foundation of WSN-middleware. It provides the high-level programming interfaces to the application programmer which separate the development of WSN based applications from the operations in the underlying WSN infrastructures.
- System services embody the functionalities and form the core of WSN-middleware. They are exposed to the application programmer through the abstraction interface, and provide the support for application deployment, execution, as well as sensor and network management.
- Middleware systems are comprised of abstractions and services to facilitate the design, development, integration and deployment of distributed applications in heterogeneous networking environments.
- Data acquisition is an essential service for WSN applications, responsible of delivering the relevant and accurate data required by the application.

- Resources in a WSN usually refer to the sensor node hardware resource, e.g. energy, memory, A/D device, and communication module.
- IRIS is composed of a potentially global collection of sensing agents (SAs) and organizing agents (OAs).

REFERENCES

1. A. Dey and G. Abowd. CybreMinder: A Context-Aware System for Supporting Reminders. In Proceedings of the Second International Symposion on Handheld and Ubiquitous Computing, pages 172–186, September 2000.
2. A.L. Murphy, G.P. Picco, and G.-C. Roman. Lime: A Middleware for Physical and Logical Mobility. In Proceedings of the 21st International Conference on Distributed Computing Systems, pages 524–533, April 2001.
3. A.L. Murphy, G.P. Picco, and G.-C. Roman. Lime: A Middleware for Physical and Logical Mobility. In Proc. of the 21st Int. Conf. on Distributed Computing Systems, Orland, USA, May 2001, pp. 524-533.
4. Bartolome Rubio, Manuel Diaz, Jose M. Troya. Programming Approaches and Challenges for Wireless Sensor Networks. In Proc. of the 2nd International Conf. on Systems and Networks Communications (ICSNC07) Cap Esterel, French Riviera, France, August 25-31, 2007, pp36.
5. C. Chong, S.P. Kumar. Sensor Networks: Evolution, Opportunities, and Challenges. In Proc. of the IEEE, 91(8), August 2003.
6. C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks. Procedings of ACM Mobicom ’00), Aug. 2000.
7. Cecilia Mascolo, Stephen Hailes. Survey of Middleware for Networked Embedded Systems. Technical Report for project: Reconfigurable Ubiquitous Networked Embedded Systems, University College London, 2005.
8. D. Box, D. Ehnebuske, G. Kakivaya, A. Layman, N. Mendelsohn, H.F. Nielsen, S. Thatte, and D. Winer. Simple Object Access Protocol (SOAP) 1.1. Technical report, W3C, May 2000.
9. D.B. Johnson, D.A. Maltz, Y.-C. Hu, and J.G. Jetcheva. The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR). Internet Draft, February 2002. IETF Mobile Ad Hoc Networking Working Group.
10. IF Akyildiz, W Su, Y Sankarasubramaniam, E Cayirci. A Survey on Sensor Networks. IEEE Communications Magazine, 2002, 40(8): 102–114.

11. Isaac Green and Randal C. Nelson. Tracking Objects using Recognition. Technical Report 765, University of Rochester, Computer Science Department, February 2002.

12. J. Bray and C. Sturman. Bluetooth 1.1: Connect without Cables. Prentice Hall, 2001.

13. J. F. Allen, G. Ferguson, and A. Stent. An architecture for More Realistic Conversational Systems. In Proceedings of Intelligent User Interfaces 2001 (IUI-01), Jan. 2001.

14. J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister. System Architecture Directions for Network Sensors. In Proceedings of the 9th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), pages 93–104, Cambridge, MA, USA, November 2000.

15. J. Hill, R. Szewczyk, A. Woo, S.Hollar, D. Culler, K. Pister. System Architecture Directions for Networked Sensors. In Proc. of the 9th Int'l Conf. Architectural Support for Programming Languages and Operating Systems (ASPLOS-IX), New York, NY, USA, ACM Press, 2000, pp. 93–104

16. J.C.D. Conway, C.J.N. Coelho, D.C. da Silva, A.O. Fernandes, L.C.G. Andrade, and H.S. Carvalho. Wearable Computer as a Multi-parametric Monitor for Physiological Signals. In Proceedings of the IEEE International Symposium on Bioinformatics and Bioengineering (BIBE), pages 236–242, 2000.

17. K. Lee. IEEE 1451: A Standard in Support of Smart Transducer Networking. In IEEE Instruments and Measurements Technology Conference, May 2000. http://ieee1451.nist.gov/IEEE1451.pdf.

18. Karen Henricksen, Ricky Robinson, A survey of middleware for sensor networks: state-of-the-art and future directions. In Proc. of the Int'l workshop on Middleware for sensor networks table of contents, Melbourne, Australia, 2006, pp: 60 – 65.

19. Licia Capra, Wolfgang Emmerich, Cecilia Mascolo. Middleware for Mobile Computing. Technical Report. Department of Computer Science, University College London, 2005.

20. Matthew Wolenetz, Rajnish Kumar, Junsuk Shin. Umakishore Ramachandran. Middleware Guidelines for Future Sensor Networks. Technical Report 30332–0280, College of Computing Georgia Institute of Technology Atlanta, Georgia, 2004.

CHAPTER 8

TIME SYNCHRONIZATION AND LOCALIZATION

INTRODUCTION

Localization is an important application in wireless communications, sensor networks, radar and sonar. It has been intensively studied by the signal processing community in the past few decades, especially after the Federal Communications Commission (FCC) issued the order of emergency 911 (E-911). Conventional source localization are based on time of arrival (TOA), time difference of arrival (TDOA), angle of arrival (AOA) or received signal strength (RSS). The TOA and TDOA based approaches strongly depends on the assumption that time synchronization has already been achieved.

On the other hand, time synchronization are traditionally studied by the computer science community from protocol design point of view. Many time synchronization protocols have been proposed for computer networks and wireless sensor networks, such as the Network Time Protocol (NTP), Timingsync Protocol for Sensor Networks (TPSN) and Reference Broadcast Synchronization (RBS). Recently, Noh et al mathematically evaluated the performance of two-way message exchange time synchronization method in wireless sensor networks.

8.1 TIME SYNCHRONIZATION

Time synchronization in all networks either wired or wireless is important. It allows for successful communication between nodes on the network. It is, however, particularly vital for wireless networks. Synchronization in wireless nodes allows for a TDMA algorithm to be utilized over a multi-hop wireless network. Wireless time synchronization is used for many different purposes including location, proximity, energy efficiency, and mobility to name a few.

In sensor networks when the nodes are deployed, their exact location is not known so time synchronization is used to determine their location. Also time stamped messages will be transmitted among the nodes in order to determine their relative proximity to one another. Time synchronization is used to save energy; it will allow the nodes to sleep for a given time and then awaken periodically to receive a beacon signal. Many wireless nodes are battery powered, so energy efficient protocols are necessary. Lastly, having common timing between nodes will allow for the determination of the speed of a moving node.

The need for synchronization is apparent. Besides its many uses like determining location, proximity, or speed, it is also needed because hardware clocks are not perfect. There are variations in oscillators, which the clocks may drift and durations of time intervals of events will not be observed the same between nodes. The concept of time and time synchronization is needed, especially in wireless networks.

8.1.1 Wired Network Synchronization

For a wired network, two methods of time synchronization are most common. Network Time Protocol and Global Positioning System (GPS) are both used for synchronization. Neither protocol is useful for wireless synchronization. Both require resources not available in wireless networks.

The Network Time Protocol requires an extremely accurate clock, usually a server with an atomic clock. The client computer wanting to synchronize with the server will send a UDP packet requesting the time information. The server will then return the timing information and as a result the computers would be synchronized. Because of many wireless devices are powered by batteries, a server with an atomic clock is impractical for a wireless network.

GPS requires the wireless device to communicate with satellites in order to synchronize. This requires a GPS receiver in each wireless device. Again because of power constraints, this is impractical for wireless networks. Also sensor networks consist of inexpensive wireless nodes. A GPS receiver on each wireless node would be expensive and therefore unfeasible. The time accuracy of GPS depends on how many satellites the receiver can communicate with at a given time. This will not always be the same, so the time accuracy will vary. Furthermore Global Positioning System devices depend on line of sight communication to the satellite, which may not always be available where wireless networks are deployed.

The constraints of wireless networks do not allow for traditional wired network time synchronization protocols. Wireless networks are limited to size, power, and complexity. Neither the Network Time Protocol nor GPS were designed for such constraints.

8.1.2 Wireless Network Synchronization

The definition of time synchronization does not necessarily mean that all clocks are perfectly matched across the network. This would be the strictest form of synchronization as well as the most difficult to implement. Precise clock synchronization is not always essential, so protocols from lenient to strict are available to meet one's needs.

There are three basic types of synchronization methods for wireless networks. The first is relative timing and is the simplest. It relies on the ordering of messages and events. The basic idea is to be able to determine if event 1 occurred before event 2. Comparing the local clocks to determine the order is all that is needed. Clock synchronization is not important.

The next method is relative timing in which the network clocks are independent of each other and the nodes keep track of drift and offset. Usually a node keeps information about its drift and offset in correspondence to neighboring nodes. The nodes have the ability to synchronize their local time with another nodes local time at any instant. Most synchronization protocols use this method.

The last method is global synchronization where there is a constant global timescale throughout the network. This is obviously the most complex and the toughest to implement. Very few synchronizing algorithms use this method particularly because this type of synchronization usually is not necessary.

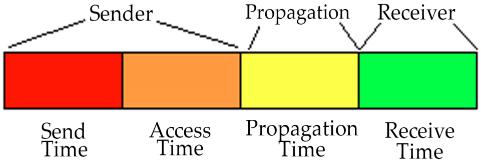


Figure 1: Breakdown of packet delay components.

As shown in figure 1, all the wireless synchronization schemes have four basic packet delay components: send time, access time, propagation time, and receive time. The send time is that of the sender constructing the time message to transmit on the network. The access time is that of the MAC layer delay in accessing the network. This could be waiting to transmit in a TDMA protocol. The time for the bits to by physically transmitted on the medium is considered the propagation time. Finally, the receive time is the receiving node processing the message and transferring it to the host. The major problem of time synchronization is not only that this packet delay exists, but also being able to predict the time spent on each can be difficult. Eliminating any of these will greatly increase the performance of the synchronization technique.

As illustrated there are many different variations of time synchronization or wireless networks. They range from very complex and difficult to implement to simpler and easy to implement. No matter the scheme used, all synchronization methods have the four basic components: send time, access time, propagation time, and receive time.

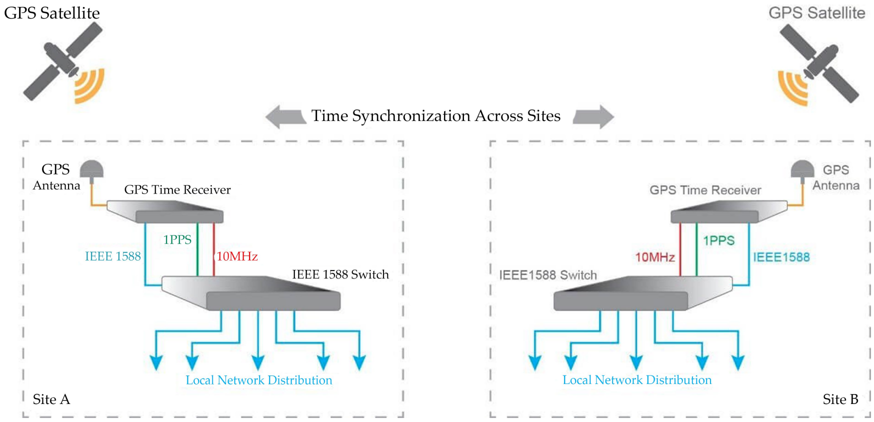
Time of day, frequency, and phase are all-important elements in synchronizing time. Time is measured by clocks, which can simply be defined as a device with a stable source frequency and a counter. Frequency is the measure of a repeating event within a period of time, normally stated in Hertz or the number of events in a second. Phase synchronization is when two separate repeating events happen at the same point in time. Accurate time synchronization in any application therefore involves the distribution of the time of day, frequency, and phase between devices.

All network devices contain components that count time, typically based on crystal oscillators that output an electrical signal with a precise frequency. When synchronized time between devices is important, clocks can therefore track the time with good accuracy. However, when precise time synchronization is required, it is found that even identical devices still lose synchronization over time. This reality is due to slight physical differences between crystal oscillators and temperature variations that affect the exact output frequency and therefore the clock time. To maintain accurate time between network devices requires continuous synchronization from a reference time source that has a more accurate, reliable clock.

The need for time synchronization in radio broadcasts, telecommunications, and test and measurement applications has a long history. Time of day codes, frequency and phase signals in various formats can be found integrated into all kinds of equipment. The following table lists some of the more common timing signals and standards that are in use, as well as current network time protocols.

Name	Description
1PPS (1 Pulse per Second)	An electrical pulse signal aligned to the start of each second. Accurate to 12 picoseconds to a few microseconds per second depending on the generating source.
10 MHz	A precise reference frequency signal used for synchronization.
TOD (Time of Day)	An interface that combines an RS-422 serial connection with a 1PPS signal.
BITS (Building-Integrated Timing Supply)	A system that distributes timing signals within a building over T1/E1 connections.
NTP (Network Time Protocol)	Standard protocol for the distribution of network time. Accurate to within tens of milliseconds.
IEEE 1588 (Precision Time Protocol)	Precision time synchronization over networks. Accuracy ranges from 10 ns to 100 ns.
GPS (Global Positioning System)	High-precision time synchronization from global satellites. Accuracy of about 100 nanoseconds.

The time synchronization method used depends mostly on the distance between the equipment that requires synchronization. A connection distance not only involves delay, but also degradation of the signal quality. Therefore, the physical signal connections of 1PPS, 10 MHz, and TOD, are limited to equipment connections in the same rack or room. BITS signals are used for connections within the same building. Both NTP and IEEE 1588 are protocols that provide time synchronization over Ethernet networks, which can span a whole range of distances. For much larger distances, GPS time has become the standard for providing precise synchronization between distant locations. Typically, a GPS device serves as a reference master clock at a location and provides time synchronization signals to other equipment and the local network.



When working with multiple computers or devices it is important to consider time synchronization. Information collected and distributed between devices will frequently be tagged with the time it was acquired. In order to properly utilize this information, it is important that times are synchronized with each other.

8.1.3 Ways to Synchronize Time Across Multiple Devices

There are many ways to synchronize time across multiple devices. Ultimately, the decision on which method to use will be based on the situation. Below are some of the most common techniques.

NTP

The Network Time Protocol, or NTP, is the most common time synchronization mechanism in use today. NTP is commonly used to synchronize the clocks of computers with a local network and also over the internet. NTP can usually maintain time to within tens of milliseconds over the public Internet and can achieve better than one millisecond accuracy in local area networks under ideal conditions.

NTP utilizes one or more servers (available on the network) and NTP client software on the device to synchronize. All versions of Windows Server include NTP server software and can optionally act as NTP servers for the local network.

If the public internet is available from all the computers in the network, and the 10-millisecond accuracy is acceptable, simply configure each computer to synchronize with one of the many available NTP servers available on the internet. This can be accomplished from the Windows

Control Panel or the Windows Clock display on the system tray.

If 10 millisecond accuracy is not adequate, or the public internet is not available, configure one of the computers in the network as an NTP server. This is done using the Windows Control Panel and is slightly different depending on the version of Windows. Then configure all the other computers to synchronize to the local server, similar to the same way you would configure them to use an internet time server.

PTP

The Precision Time Protocol, or PTP, can achieve clock accuracy in the sub-microsecond range on local networks. Note that Windows is not a real time system and does not necessarily require the accuracies of PTP. The main motivation for synchronizing Windows computers with PTP is to leverage existing infrastructure and/or synchronize with other real time devices on the network.

A network using PTP to perform time synchronization will have a PTP reference (called the PTP grandmaster) which is usually a hardware device of some type. To allow Windows computers to synchronize to PTP time, a PTP client application will need to be installed and configured. Installation and configuration of a PTP client is dependent upon the specific client being used.

GPS

Another option to provide time synchronization is by using Global Positioning System or GPS. The GPS system has a built-in clock signal accurate to 10 nanoseconds, although most GPS receivers will only provide an accuracy of 100 nanoseconds to 1 microsecond. If a GPS receiver is available, the Windows clock can be synchronized to the GPS time. Configuring by GPS varies and is dependent on the software being used to read the GPS signal from the receiver.

8.1.4 The Precision Time Protocol – IEEE 1588

For many years, the Network Time Protocol (NTP) has been the standard method for time synchronization across networks and is still widely used today. Although NTP is capable of millisecond accuracy, this was not accurate enough for some applications and in 2002 a more accurate network synchronization standard was released, called IEEE 1588 or the

Precision Time Protocol (PTP). In 2008, the PTP protocol was revised to be even more accurate, often referred to as IEEE 1588v2 or PTPv2, providing a potential accuracy down to the nanosecond level. The PTP synchronization described in this Technology Brief refers to the IEEE 1588v2 protocol.

The current PTP protocol provides fault-tolerant synchronization among clocks embedded in devices across a network. PTP uses what is called the Best Master Clock algorithm to determine the most accurate clock in a network, and then synchronizes all other clocks to the "grandmaster" clock. The grandmaster clock sends "sync" packets with embedded timestamps to "slave" clocks across the network. By accurately measuring the network delays between the grandmaster and slave clocks, precise offsets are determined to keep the slave clocks in synchronization with the grandmaster. If a grandmaster clock is no longer available on the network, the Best Master Clock algorithm defines which is the new grandmaster clock and adjusts other clocks accordingly.

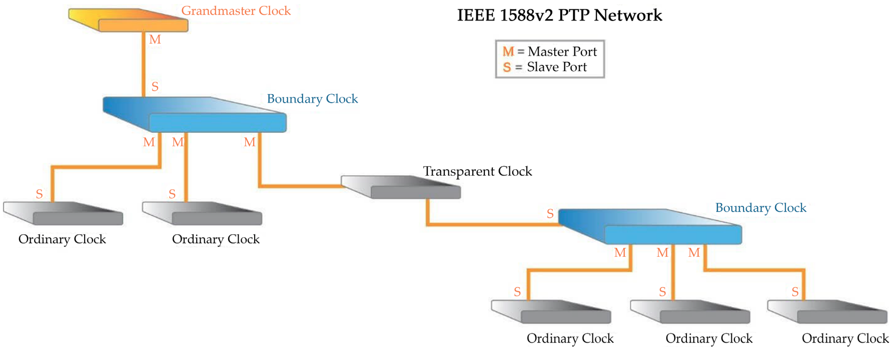
Network switches and other devices that include IEEE 1588v2 support have the ability at the hardware level to timestamp packets as they ingress and egress network ports. To prevent uncertain delays within a switch from causing inaccuracies in the time synchronization, the timestamps are added to packets between the MAC and PHY layer, exactly when a packet enters or leaves a port. For devices where there is a delay between the software sending a packet and it leaving a port, the extra delay time is sent in a follow-up packet. Once all network delays have been determined from the packet timestamps, a slave clock time can be precisely adjusted to the grandmaster clock time.

For networks that have a larger PTP domain with many switches and connected devices, the IEEE 1588v2 protocol also defines a hierarchy of clock types that help ensure accurate time synchronization across the network.

- **Ordinary Clock:** A clock device that has a single port connection to the network. This clock type can function as grandmaster or slave in the PTP domain.
- **Boundary Clock:** A clock that provides multiple connections to the network. One slave port will synchronize time with an upstream PTP clock, and other ports may serve as master ports to other downstream slave clocks. The connected slave clocks synchronize time directly with the boundary clock rather than the PTP domain grandmaster clock.

- **Transparent Clock:** A network device that does not synchronize its time but processes PTP messages and corrects for forwarding delays through the device.

Using the clock type hierarchy in a PTP network essentially removes or compensates for jitter effects and internal delays created in Ethernet switches and maintains time synchronization precision to at least the sub-microsecond level.



8.1.5 Time Synchronization to the Sub-Microsecond Level

Although time synchronization has been important for many years in a multitude of applications, the growing challenges and requirements of telecommunications and networking have given rise to impressive levels of accuracy, even over vast global distances. With GPS global satellite time being synchronized to atomic clock time, it provides an extremely precise time at any location where a GPS receiver can use a combination of 1PPS and 10MHz signals along with a PTP-capable network to synchronize any number of devices down to sub-microsecond accuracy.

8.2 SYNCHRONIZATION PROTOCOLS FOR WSNS

There are many synchronization protocols, many of which do not differ much from each other. As with any protocol, the basic idea is always there, but improving on the disadvantages is a constant evolution. Three protocols will be discussed at length: Reference Broadcast Synchronization (RBS), Timing-sync Protocol for Sensor Networks (TPSN), and Flooding Time Synchronization Protocol (FTSP). These three protocols are the major timing protocols currently in use for wireless networks. There are other synchronization protocols, but these three represent a good illustration of the different types of protocols. These three cover sender to receiver synchronization as well as receiver to receiver. Also, they cover single hop and multi hop synchronization schemes.

8.2.1 Reference Broadcast Synchronization

Many of the time synchronization protocols use a sender to receiver synchronization method where the sender will transmit the timestamp information and the receiver will synchronize. RBS is different because it uses receiver to receiver synchronization. The idea is that a third party will broadcast a beacon to all the receivers. The beacon does not contain any timing information; instead the receivers will compare their clocks to one another to calculate their relative phase offsets. The timing is based on when the node receives the reference beacon. The simplest form of RBS is one broadcast beacon and two receivers. The timing packet will be broadcasted to the two receivers. The receivers will record when the packet was received according to their local clocks. Then, the two receivers will exchange their timing information and be able to calculate the offset. This is enough information to retain a local timescale.

RBS can be expanded from the simplest form of one broadcast and two receivers to synchronization between n receivers, where n is greater than two. This may require more than one broadcast to be sent. Increasing the broadcasts will increase the precision of the synchronization.

RBS differs from the traditional sender to receiver synchronization by using receiver to receiver synchronization. The reference beacon is broadcasted across all nodes. Once it is received, the receivers note their local time and then exchange timing information with their neighboring nodes. The nodes will then be able to calculate their offset.

Advantages of RBS

The main advantage of RBS is that it eliminates the uncertainty of the sender by removing the sender from the critical path. By removing the sender, the only uncertainty is the propagation and receive time. The propagation time is negligible in networks where the range is relatively small. It is claimed that the reference beacon will arrive at all the receiving nodes instantaneously. By removing the sender and propagation uncertainty the only room for error is the receiver uncertainty. Figure 2 illustrates this concept.

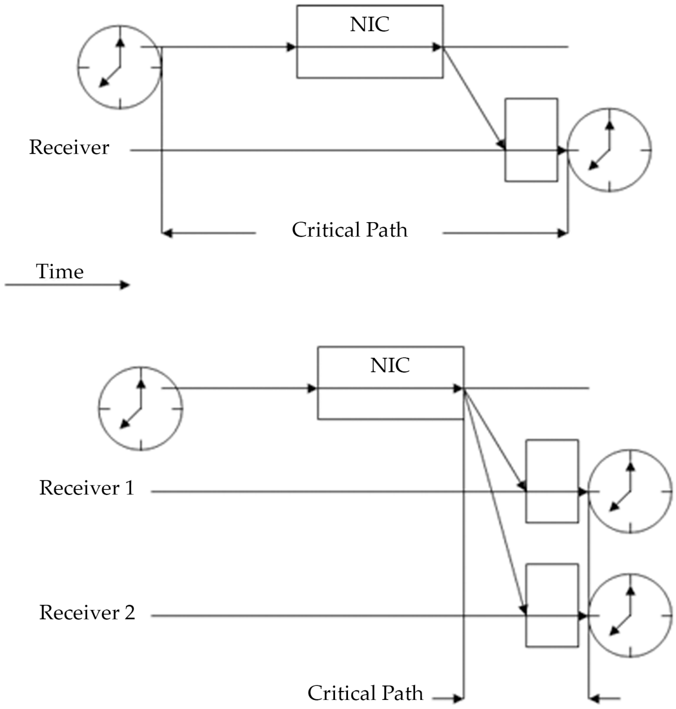


Figure 2: Comparison of a traditional synchronization system with RBS.

As seen here, the critical path in a traditional system, which is the top diagram, includes the sender. Since RBS is a receiver to receiver synchronization the sender is removed from the critical path. The critical path on contains the propagation and the receiver uncertainty. If, however, the transmission range is relatively small, then we can eliminate the propagation time and the critical path only contains the uncertainty of the receiver.

8.2.2 Timing-sync Protocol for Sensor Networks

TPSN is a traditional *sender-receiver* based synchronization that uses a tree to organize the network topology. The concept is broken up into two phases, the level discovery phase and the synchronization phase. The level discovery phase creates the hierarchical topology of the network in which each node is assigned a level. Only one node resides on level zero, the root node. In the synchronization phase all *i* level nodes will synchronize with *i-1* level nodes. This will synchronize all nodes with the root node.

Level Discovery Phase

The level discovery phase is run on network deployment. First, the root node should be assigned. If one node was equipped with a GPS receiver, then that could be the root node and all nodes on the network would be synced to the world time. If not, then any node can be the root node and other nodes can periodically take over the functionality of the root node to share the responsibility.

Once the root node is determined, it will initiate the level discovery. The root, level zero, node will send out the *level_discovery* packet to its neighboring nodes. Included in the *level_discovery* packet is the identity and level of the sending node. The neighbors of the root node will then assign themselves as level one. They will in turn send out the *level_discovery* packet to their neighboring nodes. This process will continue until all nodes have received the *level_discovery* packet and are assign a level.

Once again all nodes are assigned a level to create a tree type topology. The root node is level zero continuing down the tree with level one and so on. All nodes of level *i* will broadcast the *level_discovery* with all nodes of level *i-1*. This is maintained until all nodes are assigned a level.

Synchronization Phase

The basic concept of the synchronization phase is two-way communications between two nodes. As mentioned before this is a sender to receiver communication. Similar to the level discovery phase, the synchronization phase begins at the root node and propagates through the network.

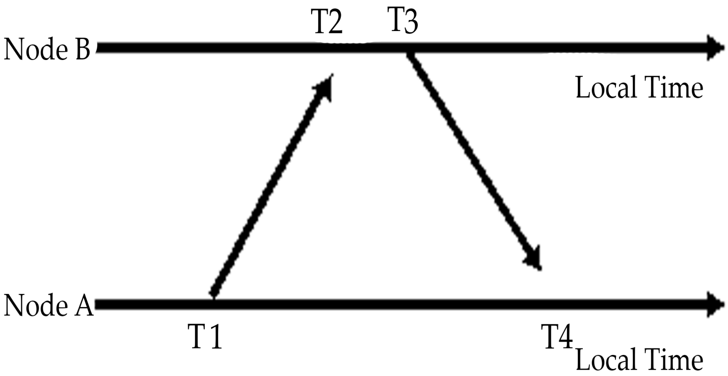


Figure 3: Two-way communication between nodes.

Figure 3 illustrates the two-way messaging between a pair of nodes. This messaging can synchronize a pair of nodes by following this method. The times T1, T2, T3, and T4 are all measured times. Node A will send the *synchronization_pulse* packet at time T1 to Node B. This packet will contain Node A's level and the time T1 when it was sent. Node B will receive the packet at time T2. Time T3 is when Node B sends the *acknowledgment_packet* to Node A. That packet will contain the level number of Node B as well as times T1, T2, and T3. By knowing the drift, Node A can correct its clock and successfully synchronize to Node B. This is the basic communication for TPSN.

The synchronization process is again initiated by the root node. It broadcasts a *time_sync* packet to the level one nodes. These nodes will wait a random amount of time before initiating the two-way messaging. The root node will send the acknowledgment and the level one nodes will adjust their clocks to be synchronized with the root nodes.

The level two node will be able to hear the level one nodes communication since at least one level one node is a neighbor of a level two node. On hearing this communication the level two nodes will wait a random period of time before initiating the two-way messaging with the level one nodes. This process will continue until all nodes are synchronized to the root node.

Again the synchronization process executes much the same as the level discovery phase. All communication begins with the root node broadcasting information to the level 1 nodes. This communication propagates through the tree until all level *i-1* nodes are synchronized with the level *i* nodes. At this point all nodes will be synchronized with the root node.

Advantages of TPSN

Any synchronization packet has the four delays discussed earlier: send time, access time, propagation time, and receive time. Eliminating any of these would be a plus. Although TPSN does not eliminate the uncertainty of the sender it does, however, minimize it. Also, TPSN is designed to be a multi-hop protocol; so transmission range is not an issue.

Unlike RBS, TPSN has uncertainty in the sender. They attempt to reduce this non-determinism by time stamping packets in the MAC layer. It is claimed that the sender's uncertainty contributes very little to the total synchronization error. By reducing the uncertainty with low level time stamping, it is claimed that TPSN has a 2 to 1 better precision than RBS and that the sender to receiver synchronization is superior to the receiver to receiver synchronization.

RBS also is limited by the transmission range. It was stated that RBS can ignore the propagation time if the range of transmission was relatively small. If it is a large multi-hop network, this is not the case. RBS would have to send more reference beacons for the node to synchronize. TPSN on the other hand was designed for multi-hop networks. Their protocol uses the tree based scheme so the timing information can accurately propagate through the network.

The sender to receiver synchronization method is claimed to be more precise than the receiver to receiver synchronization. Also TPSN is designed for multi-hop networks, where RBS works best on single hop networks. So, the transmission range is not a factor with TPSN.

8.2.3 Flooding Time Synchronization Protocol

Another form of sender to receiver synchronization is FTSP. This protocol is similar to TPSN, but it improves on the disadvantages to TPSN. It is similar in the fact that it has a structure with a root node and that all nodes are synchronized to the root.

The root node will transmit the time synchronization information with a single radio message to all participating receivers. The message contains the sender's time stamp of the global time at transmission. The receiver notes its local time when the message is received. Having both the sender's transmission time and the reception time, the receiver can estimate the clock offset. The message is MAC layer time stamped, as in TPSN, on both the sending and receiving side. To keep high precision compensation for clock drift is needed. FTSP uses linear regression for this.

FTSP was designed for large multi-hop networks. The root is elected dynamically and periodically reelected and is responsible for keeping the global time of the network. The receiving nodes will synchronize themselves to the root node and will organize in an ad hoc fashion to communicate the timing information amongst all nodes. The network structure is mesh type topology instead of a tree topology as in TPSN.

Advantages of FTSP

There are several advantages to FTSP, which it has improved on TPSN. Although TPSN did provide a protocol for a multi-hop network, it did not handle topology changes well. TPSN would have to reinitiate the level discovery phase if the root node changed or the topology changes. This would induce more network traffic and create additional overhead.

FTSP is robust in that is utilizes the flooding of synchronization messages to combat link and node failure. The flooding also provides the ability for dynamic topology changes. The protocol specifies the root node will be periodically reelected, so a dynamic topology is necessary. Like TPSN, FTSP also provides MAC layer time stamping which greatly increases the precision and reduces jitter. This will eliminate all but the propagation time error. It utilizes the multiple time stampings and linear regression to estimate clock drift and offset.

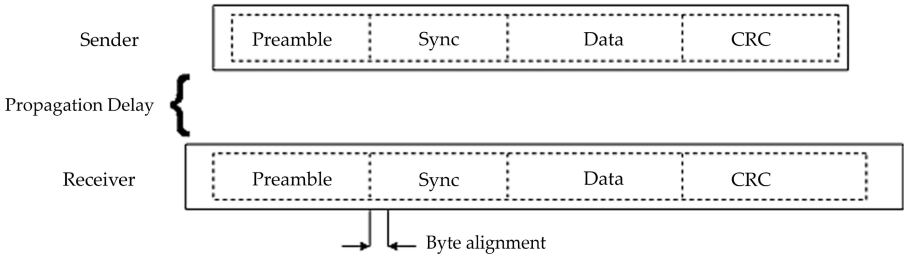


Figure 4: Data packets transmitted with FTSP.

The data packets transmitted with FTSP are constructed as shown in figure 4. There is a preamble then sync bytes followed by the data then finally the CRC. The dashed lines in the figure indicate the actual bytes in the packet and the solid line indicate the bytes in the buffer. When the sender is transmitting the preamble bytes, the receiver adjusts to the carrier frequency. Once the sync bits are received, the receiver can calculate the bit offset needed to accurately recreate the message. The time stamps are located at the boundaries of the sync bytes.

Allowing for dynamic topology changes, robustness for node and link failure, and MAC layer time stamping for precision are the major advantages of FTSP. It provides a low bandwidth flooding protocol to provide a network wide synchronization where all nodes are synchronized to the root node.

8.2.4 Attacks on Synchronization Protocols

There is a common problem among the three protocols presented. They were all developed to be energy efficient, precise, robust, and so on, but none of them were developed with security in mind. As in all computer protocols security is always an issue and attacks on protocols is inevitable.

In all synchronization attacks, the goal is to somehow convince nodes that their neighboring nodes are at a different time then they really are. Since global synchronization is the goal for some protocols and they rely on the neighboring nodes to pass the synchronization information on, compromising a node would disrupt the global synchronization.

Attacks on RBS, TPSN, and FTSP

For RBS, an attack on the synchronization can be executed easily. RBS works by receiver to receiver synchronization in which both nodes receive the reference beacon and then calculate their offset with one another. An attack would be as simple as compromising one of the nodes with an incorrect time. The non compromised node will then calculate an incorrect offset during the exchange period.

Remember TPSN is a sender to receiver tree based protocol with two phases, the level discovery phase and the synchronization phase. Both of the phases are initiated by the root node. In the synchronization phase the level number and the time are both sent through the tree. An attack would simply be to compromise a non-root node with the incorrect time. This will propagate through the tree and the closer the compromised node is to the root node, the more incorrect synchronization will occur.

Also a node could lie about its level. That would cause other nodes to request synchronization information in which it could give inaccurate information. That node also could refuse to participate in the level discovery phase, which could eliminate its children from the network.

The fundamental problem in FTSP is that it allows for any node to elect itself the root after a period of time of not receiving the synchronization information. A corrupt node could claim itself to be the root and now the other nodes will respond to its timing information instead of the correct information from the real root node. The will of course propagate through the network until all nodes have incorrectly calculated their skew and offset.

Since none of the protocols were designed with security in mind. Attacks on the synchronization are easily executed by following the rules of the protocol. In the sender to receiver synchronization, an attack will institute more damage because it will propagate through the network.

Countermeasures for Attacks

There are two major types of synchronization protocols. The single hop protocols, RBS, and the multi-hop protocols, TPSN and FTSP. In either case, the goal is to authenticate the synchronization messaging. Redundancy as well as nodes refusing to pass on bad information are other ways to combat synchronization attacks.

For single hop networks, the challenge for synchronization security is to make sure the sending node is not compromised to send out erroneous timing information. This can be accomplished by an authentication process. Either an authentication process or the use of a different private key between the sending node and each receiving node should be used for security.

In the multi-hop case an attack on a node close to the root could compromise a large portion of the network. The use of private keys in this case could also be used, but there are a few other idea. For FTSP, redundancy could be introduced so that it does not calculate its timing from just one neighbor, but from several. It could then determine if there is a corrupt node. If a node was suspicious that it was receiving bad synchronization data, it could cease retransmission of the data. This would stop the desynchronization from propagating throughout the network.

Once again, none of the protocols discussed were designed with security in mind. Therefore it is easy to compromise a node's timing and have the erroneous timing propagate through the network, especially on multi-hop networks. Authentication, redundancy, and refusal to transmit corrupt synchronization information are ways to combat attacks. The tradeoff being that these countermeasures require overhead and will induce more network traffic, but it may be a small price to pay to keep synchronization attacks from compromising the network.

8.2.5 An Industry Case

This section represents an implementation example of a time synchronization protocol. An automation facility at a Swedish mining company is using a ZigBee, IEEE 802.15.4, wireless sensor network. The goal was to have a simple timing algorithm that was energy efficient because their sensor nodes were battery powered. Their idea was to put the nodes to sleep when not in use to conserve energy.

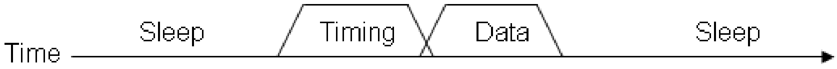


Figure 5: Synchronization phases.

The algorithm they developed was divided into three phases: timing phase, data phase, and sleep phase. The timing phase is when the node achieves or maintains synchronization. The data phase is when the data is transmitted over the network. Finally, the sleep phase is when the node enters a low power state where the radio is turned off. Figure 5 illustrates this.

The timing phase starts with the hub transmitting a single hop synchronization messages. In the message is *real_time* and *hop_count*. Initially they are both zero to indicate to the network the beginning of a frame. The nodes on the next hop will retransmit the message, after waiting a random period between 0 and 125 ms, to the next hop nodes until all nodes on the network have received the synchronization message. The *hop_count* is incremented on every hop and the *real_time* is the nodes elapsed time since the initialization. They are careful that all nodes do not retransmit if the network density is high to prevent flooding.

The company wants to further improve this algorithm. Currently every node is treated identically. Realistically, this is not the case since nodes on the edge of the network do not need the ability to route. Also they would like to trim the timing and data phases to as small as possible.

Again, they developed an synchronization algorithm that was a sender to receiver algorithm. They had a strict energy requirement. It was similar to TPSN in that the synchronization message started at the hub and propagated throughout the network one hop at a time. The nodes would also wait a random amount of time before retransmitting the synchronization message to avoid collisions much like TPSN.

In all wireless networks, the major problem for synchronization protocols is the variance in the send time, access time, propagation time, and the receive time. Elimination or the ability to accurately predict any of these greatly increases the effectiveness of the synchronization protocol. Discussion on RBS, TPSN, and FTSP was provided with each protocol's advantages. Finally a industrial case was presented and details of their protocol was given.

RBS is a broadcast protocol utilizing receiver to receiver synchronization. The designated root node would broadcast a beacon. Multiple nodes would receive the beacon simultaneously. The receivers would then note their local time upon reception and then compare with neighboring nodes. From this they would be able to calculate their phase offset. The main advantage is that this protocol removes the non-determinism from the sender. The major disadvantage is that it was not designed for large multi-hop networks.

Next was TPSN which was a sender to receiver synchronization. This protocol has two major phases, the level discovery phase and the synchronization phase. In the level discovery phase each node is assigned a level with the root node being the only node at level zero. Next is the synchronization phase where the nodes at level one would initiate the two-way messaging with the root node. This process will propagate throughout the network. Both phases are initiated from the root node. The major advantage to this is that it provides a 2 to 1 improvement on precision. The major disadvantage was that it did not allow for dynamic topology changes.

The final protocol discussed was FTSP, which was another sender to receiver synchronization protocol. FTSP broadcasts it timing information to all nodes that are able to receive the message. Those nodes in turn will calculate their offset from the global time. The receiving node will also calculate its clock skew using linear regression. The major advantage is that it is robust to compensate for node and link failures. It also allows for a dynamic topology.

These protocols were designed with performance in mind and did not take into account for security. It was shown that synchronization attacks on all these protocols were possible. Authentication, redundancy, and the refusal to pass on corrupt timing information were the countermeasure discussed.

Finally an industry case was presented where a Swedish automation facility is using a ZigBee sensor network and needed an energy efficient timing algorithm. They developed their own algorithm, but it performed much like TPSN. The hub node would start the timing and would transmit the message one hop. The receiving node(s) would then in turn retransmit the message. The nodes would enter a low power sleep mode to conserve energy.

8.3 LOCALIZATION

Localization is extensively used in Wireless Sensor Networks (WSNs) to identify the current location of the sensor nodes. A WSN consist of thousands of nodes that make the installation of GPS on each sensor node expensive and moreover GPS will not provide exact localization results in an indoor environment. Manually configuring location reference on each sensor node is also not possible in the case of dense network. This gives rise to a problem where the sensor nodes must identify its current location without using any special hardware like GPS and without the help of manual configuration. Localization techniques makes the deployment of WSNs economical. Most of the localization techniques are carried out with the help of anchor node or beacon node, which knows its present location. Based on the location information provided by the anchor node or beacon node, other nodes localize themselves.

8.3.1 Localization Techniques

Measurement techniques in WSN localization can be broadly classified into three categories: AOA measurements, distance related measurements.

Angle of Arrival (AoA)

AOA is defined as the angle between the propagation direction of an incident wave and some reference direction, which is known as orientation. Range information is obtained by estimating and mapping relative angles between neighbors make use of AoA for localization. AoA estimates the angle at which signals are received and use simple geometric relationships to calculate node positions. When the orientation is 0 or pointing to the North, the AOA is absolute, otherwise, relative. One common approach to obtain AOA measurements is to use an antenna array on each sensor node.

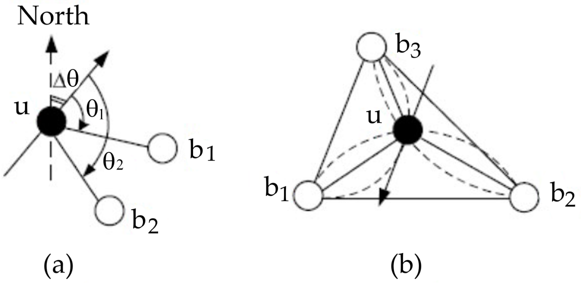


Figure 6: Triangulation in AOA localization: (a) Localization with orientation information; (b) Localization without orientation information.

AoA schemes are described where sensor nodes are forwarding their bearings with respect to anchors, i.e. nodes which is assumed to know their own coordinates and orientations. Unfortunately, these methods require a strong cooperation between neighbor nodes, and they are prone to error accumulations. Anchor nodes with adaptive antennas are used to communicate with sensors located in different parts of a network. A similar concept assumes a single anchor in the center of a network sending an angle bearing. The other nodes calculate their coordinates with the aid of the bearing and some extra information from their neighbors. However, both these solutions also need some RSS data. The position of a sensor node is determined as an intersection of antenna sectors of different anchor nodes. More precise algorithms assume that sensors can receive exact AoA information from anchors. This can be accomplished if the anchors have directional antennas rotating with a constant angular speed. The sensors can estimate the AoA of the signal registering the time when the rotating beacon has the strongest power. However, the anchors with unrealistic radiation patterns are analyzed, the radio noise is not taken into consideration and the calculations are possible only for three anchors.The rotating antennas are too large for tiny anchor nodes. Generally, the main challenge of the AoA localization schemes for WSNs is the difficulty in achieving good accuracy while keeping the system simple and feasible to implement in pocket- size devices.

Distance related measurements

Distance related measurements include propagation time based measurements, time of arrival (ToA) measurements, and time difference-of-arrival (TDoA) measurements.

Time of Arrival (ToA)

TOA is a widely used technology to perform localization. To obtain range information using ToA, the signal propagation time from source to destination is measured. A GPS is the most basic example that uses ToA. To use ToA for range estimation, a system needs to be synchronous, which necessitates use of expensive hardware for precise clock synchronization with the satellite. It is for example used in radar systems. The basic TOA technology describes the reference nodes and the blindfolded node, co-operating to determine the inter-node distances by using timing results. The blindfolded node will send a message to each of the reference nodes to measure the distance. The moment the blindfolded node transmits a message, it attaches a timestamp (t1), indicating the clock time in the blindfolded node at the start of the data transmission. At arrival of the message at the reference node, the clock time in the reference node is stored as timestamp (t2). The difference between timestamp (t1) and (t2) indicates the time needed for the signal to travel from the blindfolded to the reference node through the air. The propagation time can be directly translated into distance, based on the known signal propagation speed. These methods can be applied to many different signals, such as RF, acoustic, infrared and ultrasound. TDoA methods are impressively accurate under line-of-sight conditions. But this line-of-sight condition is difficult to meet in some environments. Furthermore, the speed of sound in air varies with air temperature and humidity, which introduce inaccuracy into distance estimation. Acoustic signals also show multi-path propagation effects that may impact the accuracy of signal detection.

Time Difference of Arrival (TDoA

To obtain the range information using TDoA, an ultrasound is used to estimate the distance between the node and the source. Like ToA, TDoA necessitates the use of special hardware, rendering it too expensive for WSNs.The propagation time can be directly translated into distance, based on the known signal propagation speed. These methods can be applied to many different signals, such as RF, acoustic, infrared and ultrasound. TDoA methods are impressively accurate under line-of-sight conditions. But this line-of-sight condition is difficult to meet in some environments. Furthermore, the speed of sound in air varies with air temperature and humidity, which introduce inaccuracy into distance estimation. Acoustic signals also show multi-path propagation effects that may impact the accuracy of signal detection.

Received Signal Strength Indicator (RSSI)

The energy of radio signal is an electromagnetic wave, which decreases as it propagates in space. As the signal propagates, its energy decreases with distance. RADAR is one of the first to make use of RSSI. RSSI has also been employed for range estimation, when the signal is received and have an estimate of the distance between sender and receiver. RSSI measures the power of the signal at the receiver and based on the known transmit power, the effective propagation loss can be calculated. After this by using theoretical and empirical models, this signal loss can be translated into a distance estimate, as used for RF signals. RSSI is a relatively cheap solution without any extra devices, as all sensor nodes are likely to have radios.

Limitations for RSSI approach

Radio propagation is affected by fading and shadowing. For an indoor application, thus will have an even more important effect, as the radio signal can be affected by the surrounding environment, and the reflections will create a multipath solution. The walls and the furniture will of course work as obstacles for the radio signal, but those are permanent obstacles. The received power at the reference distance for the model would vary. The performance, however, is not as good as other ranging techniques due to the multipath propagation of radio signals.

8.3.2 Range-Free and Range-Based Localization

Range-based and range-free techniques are discussed deeply in this section.

Range-Free Methods

Range-free methods are distance vector (DV) hop, hop terrain, centroid system, APIT, and gradient algorithm. Range-free methods use radio connectivity to communicate between nodes to infer their location. In range-free schemes, distance measurement, angle of arrival, and special hardware are not used.

DV Hop

DV hop estimates range between nodes using hop count. At least three anchor nodes broadcast coordinates with hop count across the network. The information propagates across the network from neighbor to neighbor node. When neighbor node receives such information, hop count is incremented by one. In this way, unlocalized node can find number of hops away from anchor node. All anchor nodes calculate shortest path from other nodes, and unlocalized nodes also calculate shortest path from all anchor nodes. Average hop distance formula is calculated as follows: distance between two nodes/number of hops.

Unknown nodes use triangulation method to estimate their positions from three or more anchor nodes using hop count to measure shortest distance.

Hop Terrain

Hop terrain is similar to DV hop method in finding the distance between anchor node and unlocalized node. There are two parts in the method. In the first part, unlocalized node estimates its position from anchor node by using average hop distance formula which is *distance between two nodes/ total number ofhops.* This is initial position estimation. After initial position estimation, the second part executes, in which initial estimated position is broadcast to neighbor nodes. Neighbor nodes receive this information with distance information. A node refines its position until final position is met by using least square method.

Centroid System

Centroid system uses proximity-based grained localization algorithm that uses multiple anchor nodes, which broadcast their locations with (X_i,Y_i) coordinates. After receiving information, unlocalized nodes estimate their positions. Anchor nodes are randomly deployed in the network area, and they localize themselves through GPS receiver. Node localizes itself after receiving anchor node beacon signals using the following formula :

$$(X_{\text{est}}, Y_{\text{est}}) = \left(\frac{X_i, + \cdots +, X_n}{N}, \frac{Y_i, + \cdots +, Y_n}{N}\right),$$

where X_{est} and Y_{est} are the estimated locations of unlocalized node.

APIT

In APIT (approximate point in triangulation) scheme, anchor nodes get location information from GPS or transmitters. Unlocalized node gets location information from overlapping triangles. The area is divided into overlapping triangles. In APIT, the following four steps are included.

- Unlocalized nodes maintain table after receiving beacon messages from anchor nodes. The table contains information of anchor ID, location, and signal strength.
- Unlocalized nodes select any three anchor nodes from area and check whether they are in triangle form. This test is called PIT (point in triangulation) test.
- PIT test continue until accuracy of unlocalized node location is found by combination of any three anchor nodes.

- At the end, center of gravity (COG) is calculated, which is intersection of all triangles where an unlocalized node is placed to find its estimated position.

Gradient Algorithm

In gradient algorithm, multilateration is used by unlocalized node to get its location. Gradient starts by anchor nodes and helps unlocalized nodes to estimate their positions from three anchor nodes by using multilateration. It also uses hop count value which is initially set to 0 and incremented when it propagates to other neighboring nodes. Every sensor node takes information of the shortest path from anchor nodes. Gradient algorithm follows fes steps such as the following:

i. In the first step, anchor node broadcasts beacon message containing its coordinate and hop count value.

ii. In the second step, unlocalized node calculates shortest path between itself and the anchor node from which it receives beacon signals. To calculate estimated distance between anchor node and unlocalized node, the following mathematical equation is used :

$$D_{ji} = h_{j,Ai} d_{\text{hop}},$$

where d_{ho}p is the estimated distance covered by one hop.

iii. In the third step, error equation is used to get minimum error in which node calculates its coordinate by using multilateration as follows:

$$E_j = \sum_{i=1}^{n} \left(d_{ji} - d^{ji} \right),$$

where d_{ji} is the estimated distance computed through gradient propagation.

Range-Based Localization

Range-based schemes are distance-estimation- and angle-estimation-based techniques. Important techniques used in range-based localization are received signal strength indication (RSSI), angle of arrival (AOA), time difference of arrival (TDOA), and time of arrival (TOA).

Received Signal Strength Indication (RSSI)

In RSSI, distance between transmitter and receiver is estimated by measuring signal strength at the receiver. Propagation loss is also calculated, and it is converted into distance estimation. As the distance between transmitter and receiver is increased, power of signal strength is decreased. This is measured by RSSI using the following equation:

$$P_r(d) = \frac{p_t G_t G_r \lambda 2}{(4\lambda)^2 d^2},$$

where P_t = transmitted power, G_t = transmitter antenna gain, G_r = receiver antenna gain, and λ = wavelength of the transmitter signal in meters.

Angle of Arrival (AOA)

Unlocalized node location can be estimated using angle of two anchors signals. These are the angles at which the anchors signals are received by the unlocalized nodes. Unlocalized nodes use triangulation method to estimate their locations.

Time Difference of Arrival (TDOA)

In this technique, the time difference of arrival radio and ultrasound signal is used. Each node is equipped with microphone and speaker. Anchor node sends signals and waits for some fixed amount of time which is t_{delay}, then it generates *"chirps"* with the help of speaker. These signals are received by unlocalized node at t_{radio} time. When unlocalized node receives anchor's radio signals, it turns on microphone. When microphone detects chirps sent by anchor node, unlocalized node saves the time t_{sound}. Unlocalized node uses this time information for calculating the distance between anchor and itself using the following equation :

$$d = \left(s_{\text{radio}} - s_{\text{sound}}\right) * \left(t_{\text{sound}} - t_{\text{radio}} - t_{\text{delay}}\right).$$

Time of Arrival (TOA)

In TOA, speed of wavelength and time of radio signals travelling between anchor node and unlocalized node is measured to estimate the location of unlocalized node. GPS uses TOA, and it is a highly accurate technique; however, it requires high processing capability.

We generated some interesting results by comparing few localization techniques. The results are based on our observations and analysis.

Figure 7 shows cost of four localization techniques, and it is observed that GPS- and TOA-based systems are more expensive as compared with DV hop and RSSI.

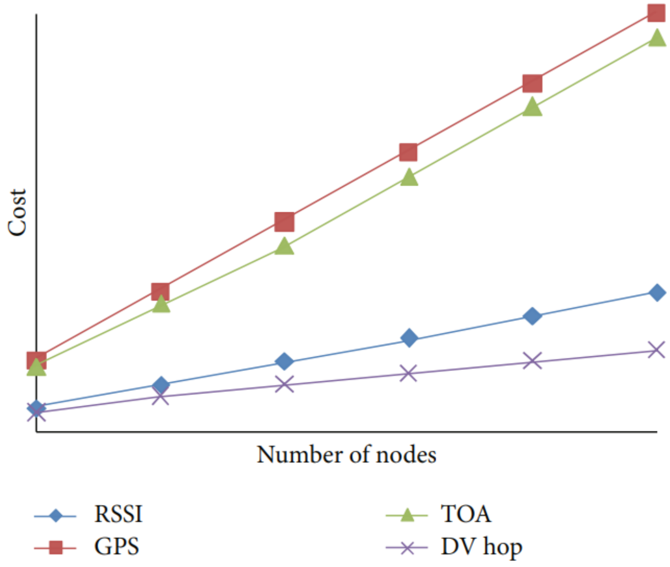


Figure 7: Cost analysis of localization techniques.

Figure 8 represents accuracy comparison of different localization techniques. It is observed that localization mechanisms equipped with GPS systems are highly accurate.

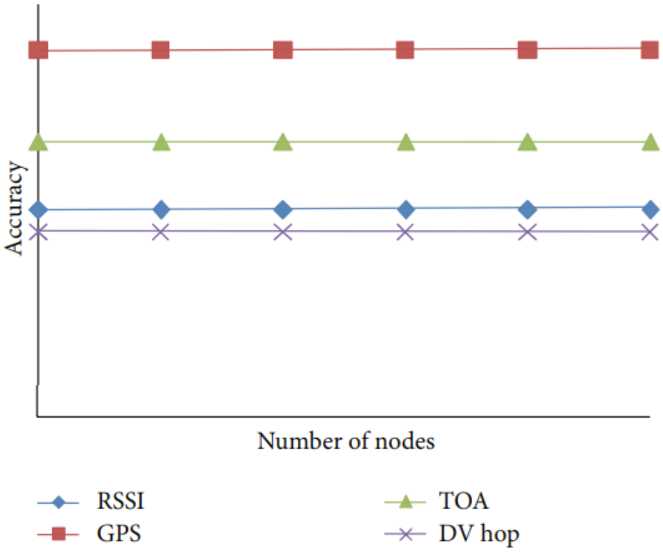


Figure 8: Accuracy comparison of different localization mechanisms.

Such mechanisms are needed for WSNs, which are energy efficient. Figure 9 shows comparison of energy efficiency of different localization mechanisms. GPS-based localization mechanisms are less energy efficient while RSSI-based mechanisms are highly energy efficient.

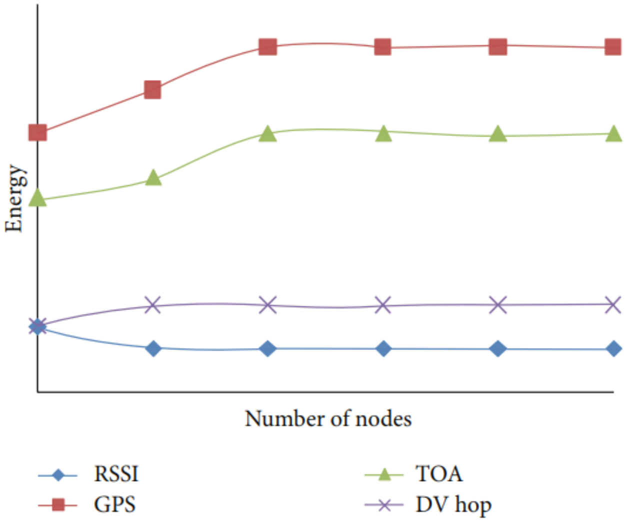


Figure 9: Energy efficiency comparison of different localization mechanisms.

8.3.3 Localization Algorithm

Many localization algorithms have been proposed for WSNs. Localization algorithms can be categorized as

- Centralized vs. Distributed : based on their computational organization
- Range-Free vs. Range-Based : to determining the location of a sensor node
- Anchor-Based vs. Anchor-Free : based on whether or not external reference nodes (i.e., anchors) are needed
- Individual vs. Collaborative Localization : calculating node position

Centralized vs. Distributed

Localization algorithms can be categorized as centralized or distributed algorithms based on their computational organization.

In Centralized algorithms

Sensor nodes send data to a central location where computation is performed and the location of each node is determined and sent back to the nodes. In certain networks where centralized information architecture already exists, such as road traffic monitoring and control, environmental monitoring, health monitoring, and precision agriculture monitoring networks, the measurement data of all the nodes in the network are collected in a central processor unit. In such a network, it is convenient to use a centralized localization scheme. Once feasible to implement, the main motive behind the interest in centralized localization schemes is the likelihood of providing more accurate location estimates than those provided by distributed algorithms. Centralized localization is basically migration of inter-node ranging and connectivity data to a sufficiently powerful central base station and then the migration of resulting locations back to respective nodes .

There exist three main approaches for designing centralized distance-based localization algorithms: multidimensional scaling (MDS), linear programming and stochastic optimization approaches. The advantage of centralize algorithms are that it eliminates the problem of computation in each node, at the same time the limitations lie in the communication cost of moving data back to the base station.The high communication costs and intrinsic delay.In most cases, costs increase as the number of nodes in the network increases, thus making centralized algorithms inefficient for large networks.

- MDS-MAP

MDS-MAP consists of three steps.

Step 1: the scheme computes shortest paths between all pairs of nodes in the region of consideration by the use of all pair shortest path algorithm such as Dijkstras or Floyds algorithm. The shortest path distances are used to construct the distance matrix for MDS.

Step 2: the classical MDS is applied to the distance matrix, retaining the first 2 (or 3) largest eigenvalues and eigenvectors to construct a 2-D (or 3-D) relative map that gives a location for each node. Although these locations may be accurate relative to one another, the entire map will be arbitrarily rotated and flipped relative to the true node positions.

Step 3: Based on the position of sufficient anchor nodes (3 or more for 2-D, 4 or more for 3-D), transform the relative map to an absolute map based on the absolute positions of anchors which includes scaling, rotation, and reflection. The goal is to minimize the sum of squares of the errors between the true positions of the anchors and their transformed positions in the MDS map.

The advantage of this scheme is that it does not need anchor or beacon nodes to start with. It builds a relative map of the nodes even without anchor nodes and next with three or more anchor nodes; the relative map is transformed into absolute coordinates. This method works well in situations with low ratios of anchor nodes .

A drawback of MDS-MAP is that it requires global information of the network and centralized computation.

In Distributed algorithms

This algorithm distributes the computational load across the network to decrease delay and to minimize the amount of inter-sensor communication have been introduced. Each node determines its location by communication with its neighboring nodes. Generally, distributed algorithms are more robust and energy efficient since each node determines its own location locally with the help of its neighbors, without the need to send and receive location information to and from a central server. Distributed algorithms however can be more complex to implement and at times may not be possible due to the limited computational capabilities of sensor nodes .Similarly to the centralized ones, the distributed distance based localization approaches can be obtained as an extension of the distributed connectivity-based localization algorithm to incorporate the available inter-sensor distance information.In Distributed localizations all the relevant computations are done on the sensor nodes themselves and the nodes communicate with each other to get their positions in a network. Distributed localizations can be categorized into different classes.

- Beacon-based distributed algorithms: Beacon- based distributed algorithms start with some group of beacons and nodes in the network to obtain a distance measurement to a few beacons, and then use these measurements to determine their own location.
- Relaxation-based distributed algorithms: In relaxation-based distributed algorithms use a coarse algorithm to roughly localize nodes in the network. This coarse algorithm is followed by a refinement step, which typically involves each node adjusting its position to approximate the optimal solution.
- Coordinate system stitching based distributed algorithms: In Coordinate system stitching the network is divided into small overlapping sub regions, each of which creates an optimal local

map, then merge the local maps into a single global map.

- Hybrid localization algorithms: Hybrid localization schemes use two different localization techniques such as proximity based map (PDM) and Ad-hoc Positioning System (APS) to reduce communication and computation cost
- Interferometric ranging based localization: Radio interferometric positioning exploits interfering radio waves emitted from two locations at slightly different frequencies to obtain the necessary ranging information for localization.
- Error propagation aware localization: When sensors communicate with each other, error propagation can be caused due to the undesirable wireless environment, such as channel fading and noise corruption. To suppress error propagation various schemes are proposed, like error propagation aware (EWA) algorithm .

Range-Free vs. Range-Based

Range Free

Range-free techniques use connectivity information between neighboring nodes to estimate the nodes position. Range-free techniques do not require any additional hardware and use proximity information to estimate the location of the nodes in a WSN, and thus have limited precision .Range-free localization never tries to estimate the absolute point to-point distance based on received signal strength or other features of the received communication signal like time, angle, etc. This greatly simplifies the design of hardware, making range- free methods very appealing and a cost-effective alternative for localization in WSNs. Amorphous localization, Centroid localization, APIT, DV-Hop localization, SeRLoc and ROCRSSI are some examples of range-free localization techniques .

Range Based

Range-based techniques require ranging information that can be used to estimate the distance between two neighboring nodes. Range-based techniques use range measurements such as time of arrival (ToA), angle of arrival (AoA), received signal strength indicator (RSSI), and time difference of arrival (TDoA) to measure the distances between the nodes in order to estimate the location of the sensors.In range-based localization, the location of a node is computed relative to other nodes in its vicinity. Range-based localization depends on the assumption that the absolute distance between a sender and a receiver can be estimated by one or more features of the communication signal from the sender to the receiver. The accuracy of such an estimation, however, is subject to the transmission medium and surrounding environment. Range based techniques usually rely on complex hardware which is not feasible for WSNs since sensor nodes are highly resource-constrained and have to be produced at throwaway prices as they are deployed in large numbers. The range methods exploit information about the distance to neighboring nodes. Although the distances cannot be measured directly they can, at least theoretically, be derived from measures of the time-of- flight for a packet between nodes, or from the signal attenuation. The simplest range method is to require knowledge about the distances to three nodes with known positions (called anchors or beacons depending on the literature), and then use triangulation.

Anchor-Based vs. Anchor-Free

Localization algorithms for WSNs, based on whether or not external reference nodes (i.e., anchors) are needed. These nodes usually either have a GPS receiver installed on them or know their position by manual configuration. They are used by other nodes as reference nodes in order to provide coordinates in the absolute reference system being used.

Anchor-based algorithms

Anchor nodes are used to rotate, translate and sometimes scale a relative coordinate system so that it coincides with an absolute coordinate system. In anchor based algorithms, a fraction of the nodes must be anchor nodes or at least a minimum number of anchor nodes are required for adequate results. At least three non collinear anchor nodes for 2- dimensional spaces and four non coplanar anchor nodes for 3-dimensional spaces are required. The final coordinate assignments of the sensor nodes are valid with respect to a global coordinate system or any other coordinate system being used. A drawback to anchor- based algorithms is that another positioning system is required to determine the anchor node positions. Therefore, if the other positioning system is unavailable, for instance, for GPS-based anchors located in areas where there is no clear view of the sky, the algorithm may not function properly. Another drawback to anchor- based algorithms is that anchor nodes are expensive as they usually require a GPS receiver to be mounted on them. Therefore, algorithms that require many anchor nodes are not very cost-effective. Location information can also be hard-coded into anchor nodes, however, careful deployment of anchor nodes is required, which may be very expensive or even impossible in inaccessible terrains.

Anchor-free localization algorithms

They do not require anchor nodes. These algorithms provide only relative node locations, i.e., node locations that reflect the position of the sensor nodes relative to each other. For some applications, such relative coordinates are sufficient. For example, in geographic routing protocols, the next forwarding node is usually chosen based on a distance metric that requires the next hop to be physically closer to the destination, a criteria that can be evaluated based on relative coordinates only.

Individual vs. Collaborative Localization

When a node has enough information about distances and/or angles and positions, it can compute its own position using one of the methods trilateration, multilateration, and triangulation. Several other methods can be used to compute the position of a node includes probabilistic approaches, bounding box, and the central position. The choice of which method to be used also impacts the final performance of the localization system. Such a choice depends on the information available and the processors limitations. Localization protocols also differ in their basic approach to calculating node position. In one class of protocols, nodes individually determine their location, using information collected from other nodes, typically involving trilateration, triangulation, or multilateration.

In a straightforward way, direct reach of at least three anchor nodes is needed for a node to compute its location coordinates. In computing the position using any of the above methods, algorithms often employ iterations, to start from the anchor nodes in the network and to propagate to all other free nodes calculating their positions. One of the problems of this approach is its low success ratio when the network connectivity level is not very high or when not enough well-separated anchor nodes exist in the network. To localize all the nodes, these algorithms quite often require that 20%-40% of the total nodes in the network be anchor nodes, unless anchor nodes can increase their signal range. To solve the problem of demanding large numbers of anchor nodes, some approaches apply limited flooding to allow reach of anchor nodes in multiple hops, and to use approximation of shortest distances over communication paths as the Euclidean distance.

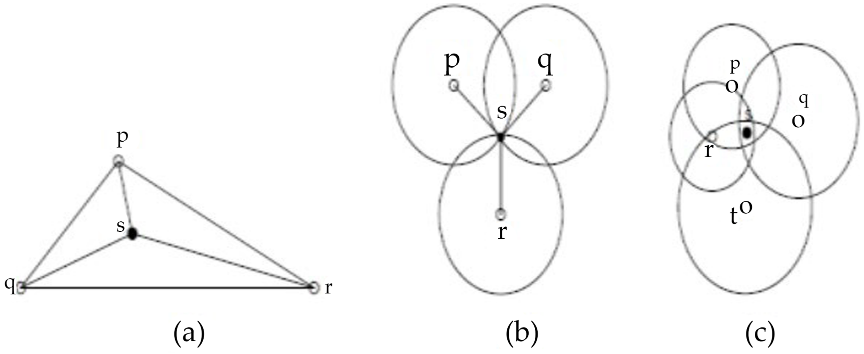


Figure 10: a) Triangulation, b) Trileteration, c) Multilateration

Triangulation

A large number of localization algorithms fall into this class. In simple terms, the triangulation method involves gathering Angle of Arrival (AoA) measurements at the sensor node from at least three sources. Then using the AoA references, simple geometric relationships and properties are applied to compute the location of the sensor node. In triangulation information about angles is used instead of distances. Position computation can be done remotely or by the node itself, the latter is more common in WSNs. The unknown node estimates its angle to each of the three reference nodes and, based on these angles and the positions of the reference nodes (which form a triangle), computes its own position using simple trigonometrically relationships .Triangulation method is used when the direction of the node instead of the distance is estimated, as in AoA systems. The node positions are calculated in this case by using the trigonometry laws of sines and cosines.

Trilateration

Trilateration is a method of determining the relative positions of objects using the geometry of triangles similar to triangulation. Unlike triangulation, which uses AoA measurements to calculate a subjects location, trilateration involves gathering a number of reference tuples of the form (x; y; d). In this tuple, d represents an estimated distance between the source providing the location reference from (x; y) and the sensor node. To accurately and uniquely determine the relative location of a point on a 2D plane using trilateration, a minimum of 3 reference points are needed .Trilateration is the most basic and intuitive method. To estimate its position using trilateration, a node needs to know the positions of three reference nodes and its distance from each of these nodes. In real-world applications the distance estimation inaccuracies as well as the inaccurate position information of reference nodes make it difficult to compute a position. In order to do localization in a network, generally some beacons, also known as anchor notes should be set up. These beacons know exactly their coordinates. If a node with unknown coordinate can measure by some approaches the distances away from these beacons, the node can calculate its coordinate using trilateration algorithm. The geometrical representation of trilateration is illustrated in the graph above. If d_1, d_2, and d_3 are accurate, N_k will locate at the intersection point of three circles; if d_1, d_2, and d_3 have some noise with them, N_k will locate at the intersection region of the three circles. The trilateration algorithm can be converted into linear equations.

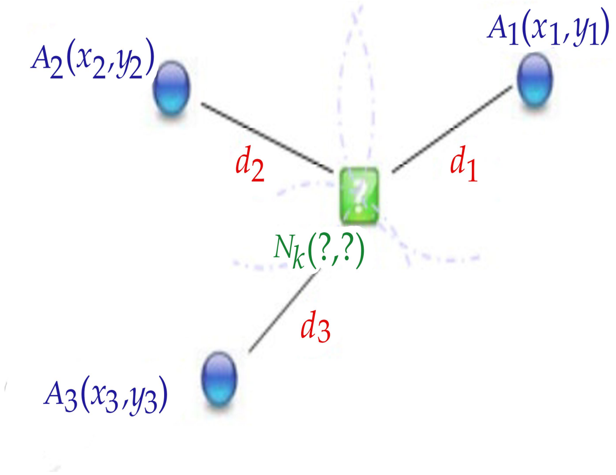


Figure 11: The geometrical meaning of trilateration: if N_k knows the exact distances to A_1, A_2, and A_3, it can calculate its coordinate.

Multilateration

Multilateration is the process of localization by solving for the mathematical intersection of multiple hyperbolas based on the Time Difference of Arrival (TDoA). In multilateration, the TDoA of a signal emitted from the object to three or more receivers is computed accurately with tightly synchronized clocks. When a large number of receivers are used, more than 4 nodes, then the localization problem can be posed as an optimization problem that can be solved using, among others, a least squares method. When a larger number of reference points are available, multilateration can be considered to compute the nodes position. The number of floating point operations needed to compute a position depends on the method used to solve the system of equations.

SUMMARY

- Time synchronization in all networks either wired or wireless is important. It allows for successful communication between nodes on the network. It is, however, particularly vital for wireless networks.
- For a wired network, two methods of time synchronization are most common. Network Time Protocol and Global Positioning System (GPS) are both used for synchronization. Neither protocol is useful for wireless synchronization. Both require resources not available in wireless networks.
- The definition of time synchronization does not necessarily mean that all clocks are perfectly matched across the network. This would be the strictest form of synchronization as well as the most difficult to implement. Precise clock synchronization is not always essential, so protocols from lenient to strict are available to meet one's needs.
- The time synchronization method used depends mostly on the distance between the equipment that requires synchronization. A connection distance not only involves delay, but also degradation of the signal quality.
- The Network Time Protocol, or NTP, is the most common time synchronization mechanism in use today. NTP is commonly used to synchronize the clocks of computers with a local network and also over the internet. NTP can usually maintain time to within tens of milliseconds over the public Internet and can achieve better than one millisecond accuracy in local area networks under ideal conditions.
- The Precision Time Protocol, or PTP, can achieve clock accuracy in the sub-microsecond range on local networks. Note that Windows is not a real time system and does not necessarily require the accuracies of PTP. The main motivation for synchronizing Windows computers with PTP is to leverage existing infrastructure and/or synchronize with other real time devices on the network.
- Another option to provide time synchronization is by using Global Positioning System or GPS. The GPS system has a built-in clock signal accurate to 10 nanoseconds, although most GPS receivers will only provide an accuracy of 100 nanoseconds to 1 microsecond.
- Localization is extensively used in Wireless Sensor Networks (WSNs) to identify the current location of the sensor nodes.

- Measurement techniques in WSN localization can be broadly classified into three categories: AOA measurements, distance related measurements
- Range-free methods are distance vector (DV) hop, hop terrain, centroid system, APIT, and gradient algorithm.
- Range-based schemes are distance-estimation- and angle-estimation-based techniques. Important techniques used in range-based localization are received signal strength indication (RSSI), angle of arrival (AOA), time difference of arrival (TDOA), and time of arrival (TOA).

REFERENCES

1. Aakvaag, N., Mathiesen, M., Thonet, G. "Timing and Power Issues in Wireless Sensor Networks - An industrial Test Case.", *Parallel Processing. ICPP 2005 Workshops* Page(s):419 - 426.
2. Cox, D., Jovanov, E., Milenkovic, A., "Time Synchronization for ZigBee Networks.", *System Theory, SSST '05. Proceedings of the Thirty-Seventh Southwestern Symposium* p. 135 - 138
3. Elson, J., Estrin, D. (2002). "Fine-Grained Network Time Synchronization using Reference Broadcast.", *The Fifth Symposium on Operating Systems Design and Implementation (OSDI)*, p. 147-163
4. Ganeriwal, S., Kumar, R., Srivastava, M. (2003). "Timing-Sync Protocol for Sensor Networks.", *The First ACM Conference on Embedded Networked Sensor Systems (SenSys)*, p. 138-149
5. Manzo, M., Roosta, T., Sastry, S. "Time Synchronization Attacks in Sensor Networks." *Proceedings of the 3rd ACM workshop on Security of ad hoc and sensor networks SASN.*
6. Maroti, M., Kusy, B., Simon, G., Ledeczi, A. "The Flooding Synchronization Protocol.", *Proc. Of the Second ACM Conference on Embedded Networked Sensor Systems (SenSys).* http://www.isis.vanderbilt.edu/publications/archive/Maroti_M_11_3_2004_The_Floodi.pdf
7. Sheu, J., Chao, C., Sun, C., "A Clock Synchronization Algorithm for Multi-hop Wireless Ad Hoc Networks.", *Distributed Computing Systems, Proceedings. 24th International Conference* p. 574 - 581
8. Sivrikaya, F.; Yener, B. (2004). "Time synchronization in sensor networks: A Survey", *Network, IEEE Volume 18, Issue 4* Page(s):45 - 50, http://www.cs.rpi.edu/~sivrif/academic/papers/IEEEnetwork04.pdf
9. Tian, Z., Luo, X., Giannakis, G.B., "Cross-layer sensor network synchronization.", *Signals, Systems and Computers. Conference Record of the Thirty-Eighth Asilomar Conference,* Volume 1, 7-10, p. 1276 - 1280.

INDEX

3GE Collection on Computer Science:
Wireless and Sensor Systems

Wireless sensor technology has been recognized as one of the emerging technologies of this century widely used for intelligent data sensing. WSNs have become an integral part of diverse applications such as environmental monitoring, military surveillance, and medicine by providing feasible communication, reliable inspection, and performing applications. WSNs are composed of a large number of sensor nodes which are densely deployed and wirelessly communicated to send and receive environmental information. A wireless sensor network (WSN) is composed of several sensor nodes, where the main objective of a sensor node is to collect information from its surrounding environment and transmit it to one or more points of centralized control, called base stations or sinks, for further analysis and processing. With the development of network and communication technology, the inconvenience of wiring is solved with WSN into people's life; especially it has wide perspective and practicability in the area of remote sensing, industrial automation control, and domestic appliance and so on. WSN has good functions of data collection, transmission, and processing. It has many advantages compared to traditional wired network, for example, convenient organizing network, small influence to environment, low power dissipation, low cost, etc. At present, near field wireless communication technology has been used widely, especially Bluetooth, wireless local area network (WLAN), infrared, etc.

A complete overview of wireless sensor network technology is given in this book. Wireless sensor network technology has become one of technological basic needs of us. Wireless sensor networks (WSNs) have grown considerably in recent years and have a significant potential in different applications including health, environment, and military. Despite their powerful capabilities, the successful development of WSN is still a challenging task. In current real-world WSN deployments, several programming approaches have been proposed, which focus on low-level system issues. In order to simplify the design of the WSN and abstract from technical low-level details, high-level approaches have been recognized and several solutions have been proposed. The book explores many fields such as wireless networks and communications, protocols, distributed algorithms, signal processing, embedded systems, and information management.

ISBN 978-1-98467-703-7
00000
9 781984 677037

BIBLIOTEX
Digital Library

www.3ge-learning.com
email: info@3ge-learning.com