

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/283824268>

Introduction to Wireless Sensor Networks

Chapter · January 2015

DOI: 10.1007/978-1-4939-2468-4_1

CITATIONS

9

READS

6,319

2 authors, including:



Mohammad Salahuddin

University of Waterloo

45 PUBLICATIONS 1,055 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



On the Evolution of Content Delivery Networks [View project](#)



Enhanced Vehicular Applications [View project](#)

Book Title:

Wireless Sensor and Mobile Ad-Hoc Networks: Vehicular and Space Applications

D. BenHaddou and A. Al-Fuqaha (Eds.), pp. 3-32, Springer, New York, 2015.

Chapter Title:

Introduction to Wireless Sensor Networks

Zille Huma Kamal and Mohammad Ali Salahuddin

Table of Contents

Abstract.....	3
1. Introduction to Wireless Sensor Networks (WSN).....	4
1.1 Wireless Sensor Network Nodes and Their Characteristics	5
1.2 Characteristics of Wireless Sensor Networks (WSN).....	7
2.0 Communication Patterns and Protocols in WSN	10
2.1 Communication Patterns	10
2.1.1 Data Delivery Models	11
2.2 Network Architecture of Wireless Sensor Networks	13
2.2.1 Transport layer protocols	14
2.2.2 Routing in Network Layer	15
2.2.3 Medium Access Control (MAC) protocols at Data Link Layer.....	18
3.0 Wireless Sensor Network Applications and Problem Space.....	20
4.0 Programming WSNs	22
4.1 TinyOS: An Operating System for WSN Motes.....	23
5.0 Summary	25
References and Bibliography	27

Abstract

Wireless Sensor Networks enjoy great benefits due to their low-cost, small-scale factor, smart sensor nodes. Not only can they be employed in cumbersome and dangerous areas of interest, for monitoring or controlling the region, but they can also be deployed to automate mundane tasks. Early sensory units were expensive and lacked the computational and communicational capabilities of current smart sensor nodes, which can now sense, process, store and forward data, all being powered by a battery.

Myriad applications exist that leverage Wireless Sensor Networks as low-cost solutions for observing the habitat and environment, from military and civilian surveillance and target detection and tracking applications, to precision farming and agriculture, patient monitoring in health care, residential applications like energy management, for safety and efficiency in vehicular networks to outer space explorations.

The diversity of the applications of Wireless Sensor Networks imposes varying design, implementation and performance requirements on the Wireless Sensor Networks. Therefore, for a thorough understanding of the different design and implementation techniques, we must understand the inherent characteristics of Wireless Sensor Networks and their smart sensor nodes. This intrinsic nature of the application specific Wireless Sensor Networks makes classification and taxonomy delineation difficult and cumbersome.

In this chapter, we will delineate the inherent characteristics of the Wireless Sensor Networks and their smart sensor nodes. Then, we will discuss the data delivery models and traffic patterns that instigate the design and development of novel network architecture protocols for WSN and distinguish them from its peers in other infrastructureless computing paradigms.

We compare WSN with its peers, with respect to, the problem space of WSN applications, followed by a brief overview of the challenges in programming WSN motes. Then, we present an overview of TinyOS, an operating system for WSN motes, and conclude with an overview of the challenges and limitations of WSNs.

1. Introduction to Wireless Sensor Networks (WSN)

Originally, sensors were electromechanical detectors for measuring physical quantities. Their first use can be traced back to 1933, in the first room thermostats [1]. Early microelectromechanical systems (MEMS) consisted of a multi-chip, where a sensor and its electronics and mechanics were housed on separate chips and packages. This resulted in larger size, more cost and lower yield of the sensor [2]. Recent advances in microelectromechanical systems (MEMS) and integrated circuits (IC) have enabled the development of small-scale sensors and the integration of its actuators and electronics into one cost-effective high-performance chip. Over the past decade, these sensors have evolved into smart sensors, which now include an on-board processor, memory and transceiver, all in a small-scale factor, powered by a battery source. These smart sensors constitute a node in the Wireless Sensor Network.

The benefit of the small-scale node is two-fold, first, the low production cost and second, the easy and low installation cost. Currently, the price of a Wireless Sensor Network node ranges in hundreds to thousands of dollars, however, it is envisioned to reduce to a couple of dollars with advances in technology and mass production [3]. The small size and low cost of the node allows an ease in the installation process, where nodes can be randomly air-dropped or precisely placed, based on the application.

These low cost nodes with processing, storage and sensing capabilities, coupled with the attractive infrastructureless networking capabilities of the transceiver, market the Wireless Sensor Network as a powerful, low cost solution for numerous problems in diverse areas of research. They offer collaboration in a distributed environment for sensing and processing information. Vital information is routed through a multi-hop ad hoc network to a Wireless Sensor Network sink, a collector of data or to a base station, which acts as a gateway to a fixed infrastructure. Variations of Wireless Sensor Networks (WSN) include nodes with actuators that comprise a Wireless Sensor and Actuator Network (WSAN) ([4], [5]), mobile nodes in Wireless Sensor Networks [3], or nodes capable of handling multimedia content like audio and video streaming or still images in a Wireless Multimedia Sensor Network (WMSN) [6].

Myriad applications exist that leverage Wireless Sensor Networks as low-cost solutions for observing the environment, for example, in military and civilian surveillance and target detection and tracking applications (e.g. [7], [8], [9], [10]), for monitoring and controlling industrial processes [11], for precision farming and agriculture [12], environment and habitat monitoring ([13], [14]) patient monitoring in health care ([15], [16]), residential applications like energy management [17], in vehicular networks for safety [18] and efficiency [19] and even for outer space explorations [20].

In the following sections, we will discuss the various characteristics of WSN nodes and the intrinsic nature of WSN, followed by a discussion of the traffic patterns and network architecture

protocols for WSN and their taxonomy. We will conclude with a summary of the challenges and limitations of WSN.

1.1 Wireless Sensor Network Nodes and Their Characteristics

Technological advances in MEMS and Integrated Circuits (IC) instigate the widespread availability of low-cost, small-scale sensors, which evolved into smart, battery powered sensors, with processing and communicating capabilities. These smart sensors constitute the Wireless Sensor Network node, referred to as motes, hereafter, in honor of the first Wireless Sensor Network nodes, which were University of Berkley's Rene and Mica Motes [21]. Figure 1, illustrates a typical WSN mote with its fundamental units. Minimally, a mote consists of a battery-operated unit with single or multiple sensors, a processing unit with storage and a transceiver. Typically, expansion slots exist or can be attached to expand the mote to include other application specific units, such as, global positioning system (GPS) for localization, or power harvesting units from solar or wind energy, or complementary metal-oxide semiconductor (CMOS) chips for multimedia capabilities, etc.

These motes are placed on programming boards, interfaced with a computer, so that the WSN application and its operating system can be flushed into the memory of the mote. At this time, the motes can also be programmed with a specific identification number and, or group identification number. Various motes can also be programmed without a physical connection to the computer, known as Over The Air (OTA) programming.

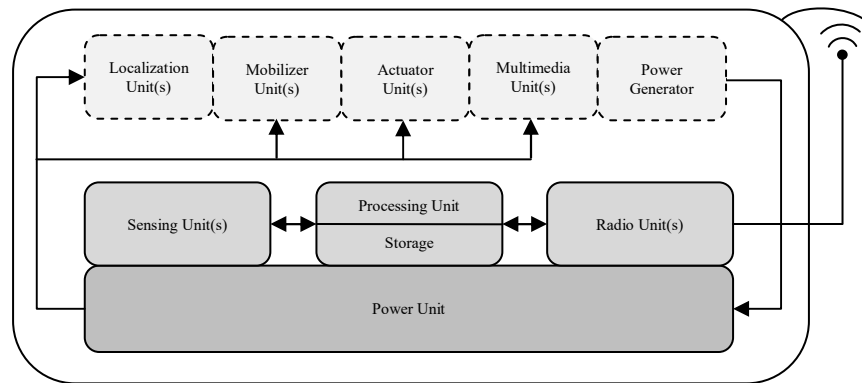


Figure 1. A typical WSN Mote with its fundamental units

WSN motes can vary greatly, with respect to the size, their cost, processing power, communication range, protocols and operating systems. WSN motes can be as large as a shoebox, e.g. Sensoria Wireless Integrated Network Sensors (WINS) Next Generation (NG) 2.0 [22], or as miniature as a coin, like Moteiv Corporation's Tmote Mini [23], but typical WSN mote dimensions are in the order of a couple of centimeters [24].

WSN motes are typically equipped with multiple sensors for sensitivity to various environmental factors, e.g. mechanical, thermal, biological, chemical, optical, and magnetic, etc. Often motes have

expansion slots that enable them to be equipped with mechanical actuators, wheels for mobility or CMOS chips or microphone for multimedia capabilities. The processors used in these motes can range from ultra-low-power 8 bit processors to more powerful 32 bit processors, similarly, memory space can vary from a couple of kilobytes to the order of megabytes [25].

Motes are equipped with short-range radio frequency (RF) transceivers to enable WSN applications [25] and to ease the query and retrieval of data from the WSN [26]. The use of a short range radio directly influences the antenna size, since short range radio waves have higher frequencies and shorter wavelengths they can be received and transmitted by small compact antennas. Through the high frequencies of 800-1000 MHz, IEEE 802.15.4 or the 2.4GHz Bluetooth, the low-power radios offer varying bandwidths [25]. The low transmission range of the motes instigates a multi-hop WSN. Though current WSN motes are equipped with RF radios, communication via infrared, ultrasound and inductive fields [27] has also been explored [25].

WSN motes can also be equipped with power generation units, that harvest power ambient energy sources such as solar [28], mechanical and thermal [29]. Table I presents a comparison of some WSN motes used in academia and industry w.r.t to these characteristics ([25], [24], [30]).

We will now consider the characteristics of Wireless Sensor Networks, with respect to the various application and environmental aspects.

Table I. Comparison of some WSN Motes

Platform	CPU	Clock (MHz)	RAM/Flash/EEPROM	Radio Transceiver	BW (kbps)	Freq. (MHz)	OS
AWAIRS 1	Intel StrongARM SA1100	59-206	1M/4M	Conexant RDSSS9M	100	900	MicroC/OS
μ AMPS	Intel StrongARM SA1100	59-206	1M/4M	National LMX3162	1000	2400	μ OS
Dot	Atmel Atmega 163	8	1K/16K/32K	RFM TR1000	10	916.5	TinyOS
BT Node	Atmel Atmega 128L	8	4K/128K/4K	ZV4002 BT/CC1000	1000	2400	TinyOS
Smart-its	PIC 18F252	8	3K/48K/64K	Radiometrix	64	433	Smart-its
Mica2	Atmel Atmega 128L	8	4K/128K/512K	Chipcon CC1000	38.4	900	TinyOS
Mica2Dot	Atmel Atmega 128L	4	4K/128K/512K	Chipcon CC1000	38.4	900	TinyOS
iBadge	Atmel Atmega 128L	8	4K/128K	Ericsson ROK101007 BT	1000	2400	Palos
CENS Medusa MK2	Atmel Atmega 128L/Atmel AT91FR4081	4/40	4K/32K/136K/1M	RFM TR1000	10	916	Palos
iMote	Zeevo ZV4002 (ARM)	12-48	64K/512K	Zeevo BT	720	2400	TinyOS
U3	PIC 18F452	0.031-8	1K/32K/256	CDC-TR-02B	100	315	Pavenet
RFRAIN	Chipcon CC1010 (8051)	3-24	2K/32K	Chipcon CC1010	76.8	0.3-1000	RFRAIN Libraries
Nymph	Atmel Atmega 128L	4	4K/128K/512K	Chipcon CC1000	38.4	900	Mantis
Telos	TI MSP430F149	8	2K/60K/512K	Chipcon CC2420	250	2400	TinyOS
MicaZ	Atmel Atmega 128L	8	4K/128K	Chipcon CC2420	250	2400	TinyOS
Particle2/29	PIC 18F6720	20	4K/128K/512K	RFM TR1001	125	868.35	Smart-its
eyesIFXv2	TI MSP430F1611	8	10K/48K	Infineon TDA 5250	64	868	TinyOS
iMote2	Intel PXA 271	13-104	256K/32M	Chipcon CC2420	250	2400	TinyOS

Platform	CPU	Clock (MHz)	RAM/Flash/EEPROM	Radio Transceiver	BW (kbps)	Freq. (MHz)	OS
TelosB/Tmote Sky	TI MSP430F1611	8	10K/48K/1M	Chipcon CC2420	250	2400	TinyOS
Ember RF Module	Atmel Atmega 128L	8	4K/128K	Ember 250	250	2400	EmberNet
XYZ Sensor node	OKI ML67Q500x (ARM/THUMB)	1.8-57.6	4K/256K/512K	Chipcon CC2420	250	2400	SOS
Ant	TI MSP430F1232	8	256/8K	Nordic nRF24AP1	1000	2400	Ant
ProSpeckz II	Cypress CY8C2764	12	256/16K	Chipcon CC2420	250	2400	Speckle net
Fleck	Atmel Atmega 128L	8	4K/128K/512K	Nordic nRF903	76.8	902-928	TinyOS
Sun Spot	Atmel AT91FR40162S	75	256K/2M	Chipcon CC2420	250	2400	Squawk VM (Java)
ECO	nRF24E1 (8051)	16	4K/512/32K	Nordic nRF24E1	1000	2400	-
SHIMMER	TI MSP430F1611	4/8	10K/2G	WML-C46A BT/CC2420	250	2400	TinyOS
IRIS	Atmel Atmega 1281	8	8K/640K/4K	Atmel ATRF230	250	2400	TinyOS
WaspMote	Atmel Atmega 1281	8	8K/128K/4K	eUnistone 31308/2	38.4	2400	-
				Zig Bee		2400/868 / 900	
				WiFi – 802.11 b/g		2400	
				GSM - SIM900 (SIMCom)		850/900/1800/1900	
SquidBee	ATmega8 / ATmega168	16	1K/8or16K/512K	MaxStream	250	2400	-

1.2 Characteristics of Wireless Sensor Networks (WSN)

Wireless Sensor Networks are deployed in a region of interest over a period of time. Since the motes have a short-range radio and a small coverage area, Wireless Sensor Networks typically contain a large number of motes. These motes form multi-hop networks and collaborate with each other to maintain connectivity and coverage. Apart from the traditional concerns on collaboration and communication in an ad hoc environment, WSN are also plagued with power and energy management concerns, due to the battery operated motes. In this section, we will delineate the various design criteria of application specific WSNs.

Region of Interest – The region of interest in WSN applications can be divided into those that are dangerous or isolated versus those that are mundane and cumbersome. In both scenarios, there is little or no infrastructure ([3], [26]). For example, volcano monitoring [13] a dangerous task for humans can be accomplished using low-cost motes that are deployed in the region. These motes are not only expendable but also provide critical data analysis and lifesaving information. However, a WSN used in life rhythm analysis for the elderly [31], can automate the mundane process of gathering vital signs and provides continuous statistics, rather than at fixed intervals.

Modes of Deployment – There are two distinct mote deployment strategies, random and precise. In random deployment, motes are randomly distributed like nodes in wireless ad hoc networks. This

deployment could be rocket ejected or aircraft sown [32], e.g. in the aerial spread of motes for volcano monitoring [13]. Precise deployment usually consists of manual or pre-planned placement of motes, e.g. habitat monitoring on Skomer Island [33]. The deployment strategies used affects the structure of WSN and can have an impact on coverage of the region and cost of deployment [26].

Organization and Architecture –WSN can be organized into two typical structures, flat or hierarchical. Figure 2 illustrates the flat and hierarchical WSN. In a flat network, all motes in the network have the same role and importance, whereas, in a hierarchical organization, motes are clustered or organized into groups with different motes playing different roles, such as general purpose sensing motes or data aggregators or forwarders, etc. As we climb up the hierarchy, the mote's functionality increases with respect to its cost, size, processing power, storage size, etc.

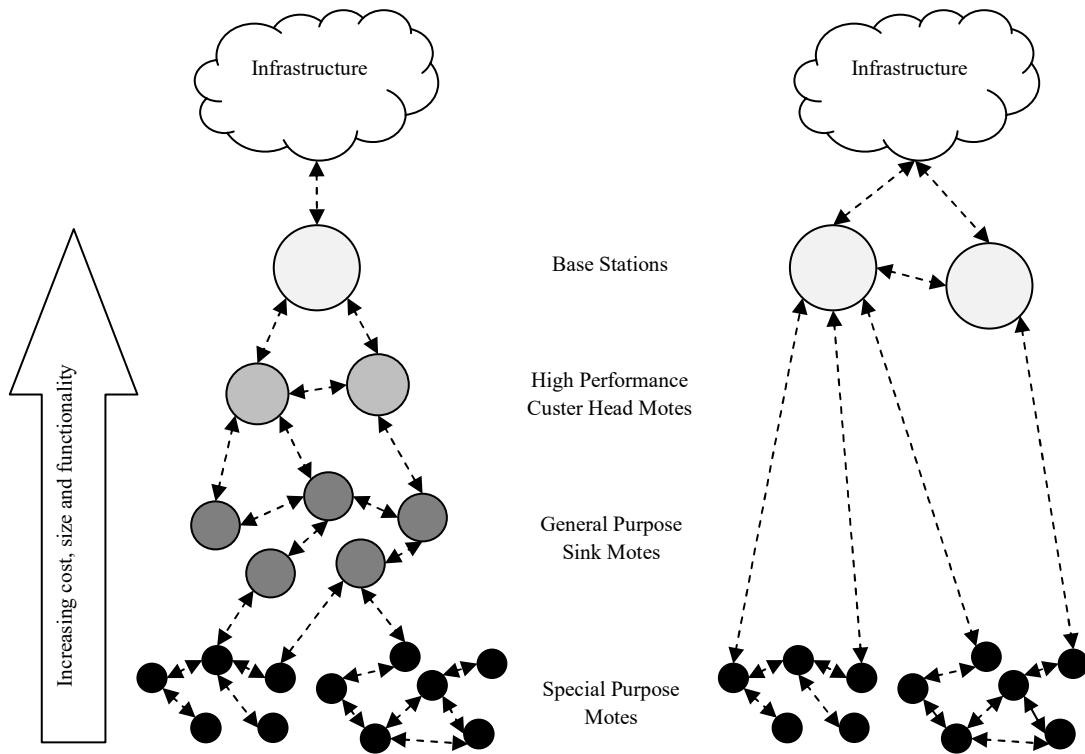


Figure 2. WSN Architecture, hierarchical (left) and flat (right)

For best results, WSN deployment should consist of a hierarchical organization of heterogeneous motes, integrating a larger number of low-power general purpose motes with smaller number of specialized or high-performance motes [24]. Furthermore, these motes can be equipped with actuators [4] or consist of mobile motes [3].

In WSN architectures, single or multiple base stations preside over the entire WSN. These base stations act as a gateway between the WSN and other fixed infrastructure, e.g. Internet. The base stations are typically, high-performance, human operated devices with rechargeable power source, e.g. laptops interfaced with a mote. Lower down the hierarchy, would be specialized motes with slightly better

configurations than the low-cost, low-power general purpose motes at the bottom of the hierarchy or pyramid. The special purpose motes offer data aggregation, or fusion, and more complex computational and communicational power, making them ideal sink nodes in the WSN. The general purpose motes at the bottom of the hierarchy typically act as data collectors and forwarders, as illustrated in Figure 2.

Multiple WSN can be interconnected to form a global WSN, referred to as global sensing infrastructure [34], as illustrated in Figure 3.

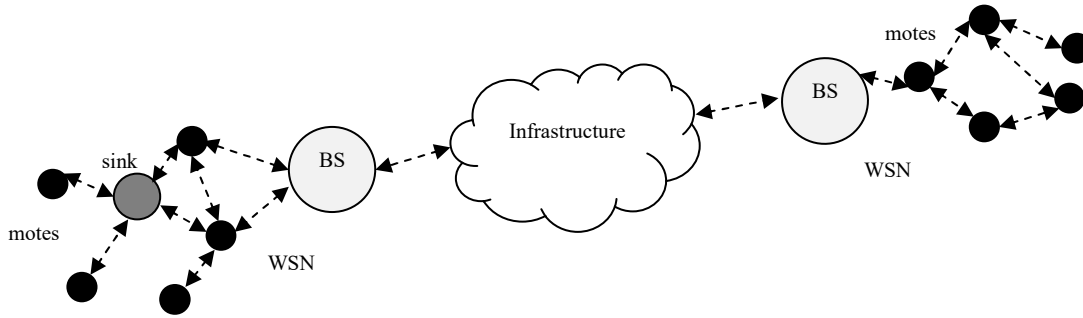


Figure 3. Global WSN Infrastructure

Lifetime of the WSN – WSN are deployed with the intention of providing long-term data collection at previously unimaginable, scales and resolutions [35]. Typical WSN habitat monitoring applications benefit from long-term data that help decipher data trends and are necessary to detect significant change in habitat [36]. Thus, the lifetime of a WSN is a fundamental characteristic. It is bounded by the finite power source of the battery operated motes and the nature of the deployed region, and infeasible or cumbersome task of replacing and disposing these batteries. This necessitates power management in hardware and software components of the motes ([37], [3]) and in WSN protocols [38]. Reinforcement strategies are also being explored in WSN through power harvesting techniques, from solar [28], mechanical and thermal [29] sources.

Ad hoc Communication – The lack of infrastructure in WSN is an intrinsic property of other communication systems like Bluetooth and mobile ad hoc networks (MANETs) [3]. Bluetooth employs discovery protocols and MANETs use robust and dynamic configuration protocols to establish and maintain the network infrastructure. Though these systems are the closest peers of WSN, there are fundamental differences that distinguish WSN from its peers [3]. First, WSN has larger number of motes with smaller transmission power and radio range, whereas Bluetooth and MANETs have smaller number of nodes with larger transmission power and longer radio ranges. Second, WSN and MANETs suffer from dynamic topology changes due to mote/node mobility and drop-out, but rate of mobility and drop-out is slower in WSN. Lastly, WSN motes have a finite, non-rechargeable power source, whereas, Bluetooth and MANET nodes can be recharged. The critical power management in WSN imposes challenging requirements in WSN protocols and inhibits the direct use of existing Bluetooth and MANET protocols in WSN [3]. Therefore, WSN needs power-aware protocols for a large-scale, autonomous and

heterogeneous network of low-processing power and low-transmission rate nodes. These protocols are needed for establishing and maintaining a secure network infrastructure through self-discovery, configuration, and healing, routing and inherent fault-tolerance.

Self-configuration and organization – Various types of self-discovery algorithms exist for use in WSN, some are adaptations from Bluetooth discovery mechanism and others developed specially for WSN. Algorithms delineated in [39], [40], [41] and their like consist of a suite of protocols responsible for discovering nodes and organizing the wireless infrastructure, despite topology changes and node failures.

Connectivity and coverage – Establishing and maintaining connectivity and coverage is an essential step in the deployment of a WSN. It is important to determine the initial number of nodes required to maintain connectivity and coverage in the region of interest, in face of node failure and dropout. It is generally assumed that node communication and sensing range are uniform and unidirectional, usually depicted as circular regions (cf. Section 2.1.1 Figure 4).

Furthermore, it is imperative for the configuration algorithms to maintain connectivity in the face of dynamic topology changes. Berman et al [42] propose dominating k -connectivity, such that in the face of upto k node failures, connectivity to a WSN sink is still maintained. Abbasi et al [43] present a recovery technique for preserving connectivity in the face of node failure. Others ([44], [45], [46], [47], etc) have also studied the matter of connectivity and coverage in WSN in great detail.

2.0 Communication Patterns and Protocols in WSN

After the WSN configuration and organization algorithms have setup the WSN infrastructure for routing and communication. WSN nodes can collaboratively perform the application-specific distributed tasks across the WSN. Over the lifetime of the WSN, human operated base stations, which are essentially gateways to fixed infrastructure, e.g. Internet backbone, are used to query and retrieve data from the WSN.

2.1 Communication Patterns

There are two different approaches in sampling and retrieving data from the WSN, namely, the push and pull approach. In a push approach, WSN nodes are programmed to autonomously sample the environment at fixed intervals and *push* their data into the network. They push data towards data collectors like the base station or sink. On the other hand, in the pull approach, nodes wait for an explicit command from the base station or sink to start sampling [48]. Both approaches have their own advantages and drawbacks. Obviously, push approach suffers from continuous sampling, bottleneck at sink or base station but enjoys low latency query response [48]. Whereas, in the pull approach, there is an increase in

the number of messages required for querying and there is a longer delay in query response, but conserves power by explicit sampling rather than continuous sampling [48].

WSN applications typically utilized the push approach to conserve energy, when radio communication was more expensive than sensing on a mote [48], however, advances in radio technologies have significant improvements in the power consumption, resulting in ultra-low power radios, e.g. ZigBee radio [48]. Bose and Abdelsalam [48] propose a hybrid push-pull approach to sampling and retrieving data from a WSN to leverage the benefits of both techniques. The push, pull or hybrid approach yields three distinct network traffic patterns in WSN. The push approach causes a many-to-one communication pattern, where motes are pushing data towards the sink or base station. In the pull approach, before motes-to-sink communication occurs, the sink or base station sends the explicit command to start sensing, which is one-to-many communication. When motes are collaboratively communicating, to self-configure, localize, or for data fusion at multiple sinks, many-to-many communication occurs [49].

Apart from these traffic patterns in WSN, Tilak *et al* [50] decompose communication paradigms in WSN into two categories, that is, application-based and infrastructure-based communication. Application-based communication refers to getting the sensed data from nodes to application user, whereas, infrastructure-based protocols are those that are used to provide the underlying primitives to achieve the application-based communication.

We will discuss data delivery models from the application and infrastructure based communication perspectives, followed by a discussion of the network architecture in WSN.

2.1.1 Data Delivery Models

From the aspect of infrastructure based communication protocols the flow of data packets are based on four distinct data delivery models unicast, broadcast, multicast [50], or many-to-many. Unicast is one-to-one communication between two motes, versus, the one-to-many (i.e. all motes in the communication range of a mote) communication of broadcast. The motes within the communication range of a mote are often referred to as one-hop neighbor or simply neighbors. In multi-hop communication, data is routed between motes, using unicast communication.

Multicast pertains to communication of motes in a group. Motes are categorized into groups by the application and only process those packets that contain a group identification number that matches the motes group identification number. These WSN groups can form a connected or disconnected sub-network. In a disconnected group, multi-hop communication would be used to enable multicast data delivery models. If there is a geographic overlay on the multicast WSN and packets are transmitted to motes based on their specific geographical location [51], then this specialized form of multicast data

delivery is known as Geocasting. Multicast data delivery model can also be specialized to behave in a many-to-one data delivery manner.

Finally, many-to-many data delivery models results in the presence of multiple sinks or gateways accessing the WSN for data and information [49]. These data delivery models in WSN are illustrated in Figure 4.

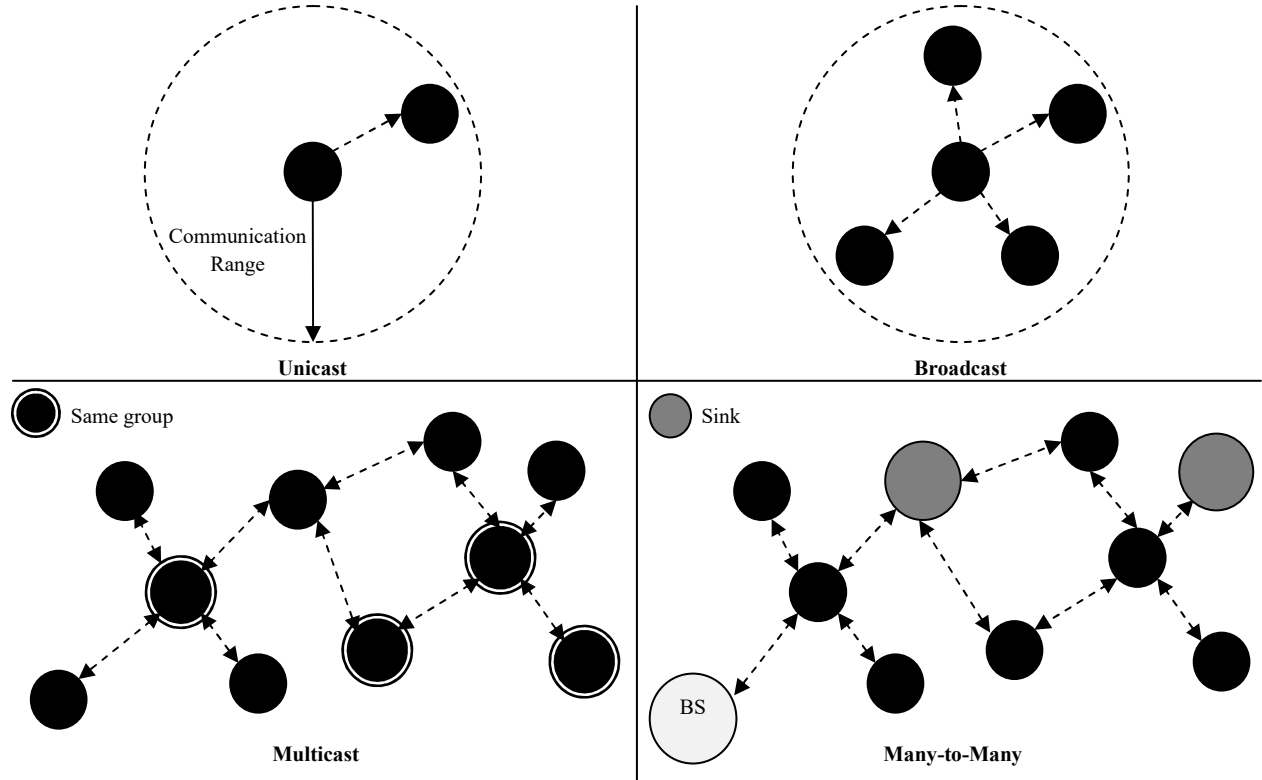


Figure 4. WSN infrastructure based data delivery models, unicast, broadcast, multicast and many-to-many

From the perspective of the application based communication protocols, the data delivery models consist of continuous, event-driven, query-driven or a hybrid approach of these ([50], [52]), as illustrated in Figure 5. In continuous, also known as periodic, data delivery models, data is transmitted from the motes to the sink or gateways at periodic intervals. In event-driven models, the motes are sensitive to one or more physical factors of the environment, when the sensed value (sensor readings) meets or exceeds a predetermined threshold, an event is triggered. The triggered motes propagate their sensor readings back to the sink(s) or gateways. In query-driven data delivery models, user initiates a query and network motes which meet the query criteria transmit their sensor readings. In the hybrid approach, one or more of the data delivery models are used together. For example, in a volcano monitoring application, the motes would be primarily event-driven. However, occasionally the user may want to poll or query the WSN for current seismic or temperature readings.

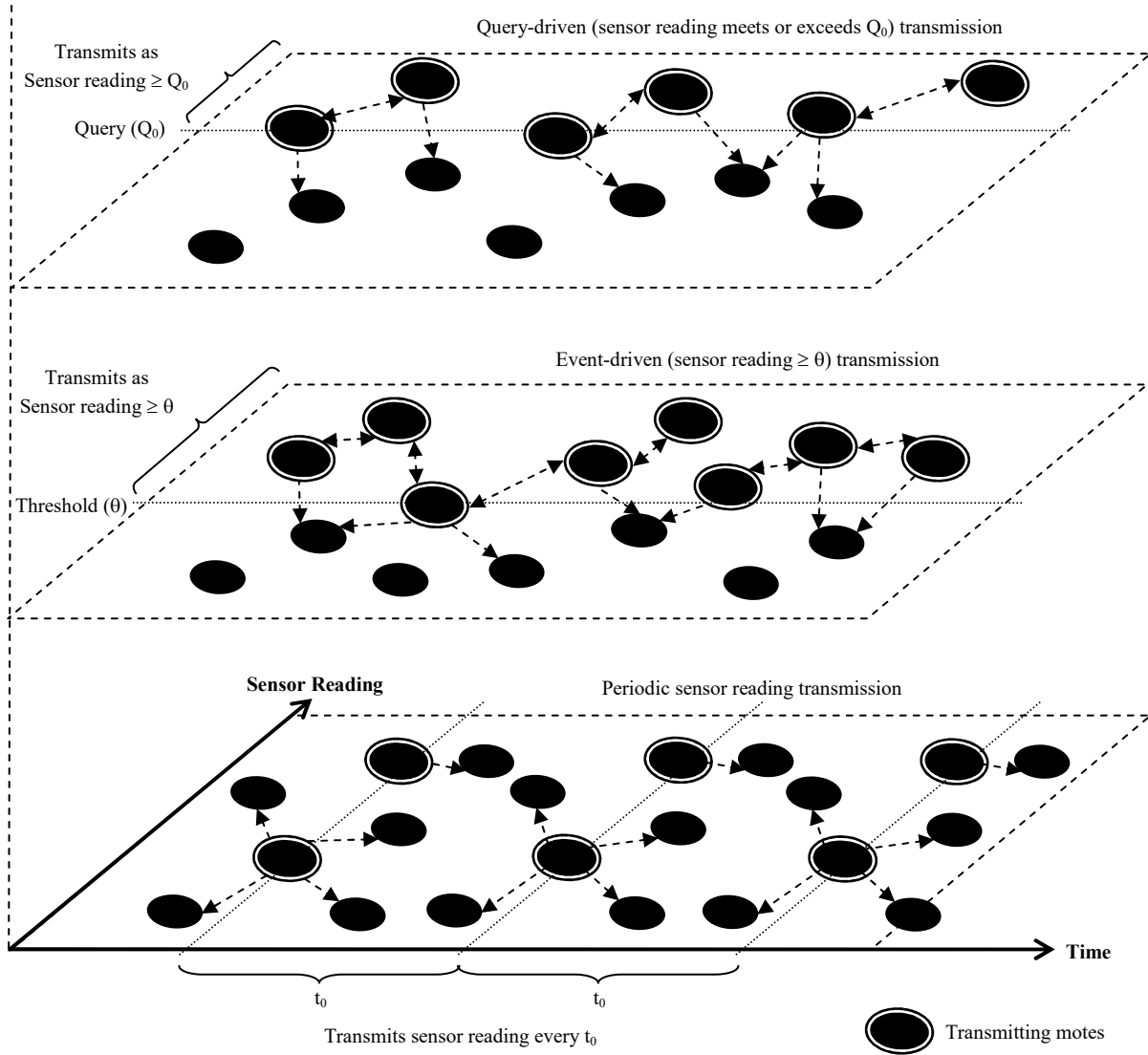


Figure 5. WSN application based data delivery models in WSN, periodic, event-driven and query-driven

2.2 Network Architecture of Wireless Sensor Networks

The inherent characteristics of the nodes in a WSN, such as limited processing, memory and communication capabilities, impose design and performance criteria on the routing and communication protocols in WSN. Communication protocols are used to dictate, manage and control all aspects of communication, from the lowest layer of accessing the physical medium to higher layers responsible for end-to-end transmission of packets and routing of data, to the top most application layer for data and packet formation. Figure 6, depicts an overview of the network architecture in WSN.

This is essentially different from typical network models due to the need for power management across all layers of the model. Furthermore, due to the constrained nature of nodes and the application-specific nature of WSN, there is no clear demarcation in the network architecture layers, as in traditional

network architecture. Often, fundamental WSN communication protocols for querying, routing and data delivery are integrated under data dissemination and aggregation techniques in WSN [53]. These limitations instigate the use of a cross layer approach in developing network protocols for wireless sensor networks.

In the following sections, we will discuss the various concepts of the transport, network and data link layers with respect to wireless sensor networks.

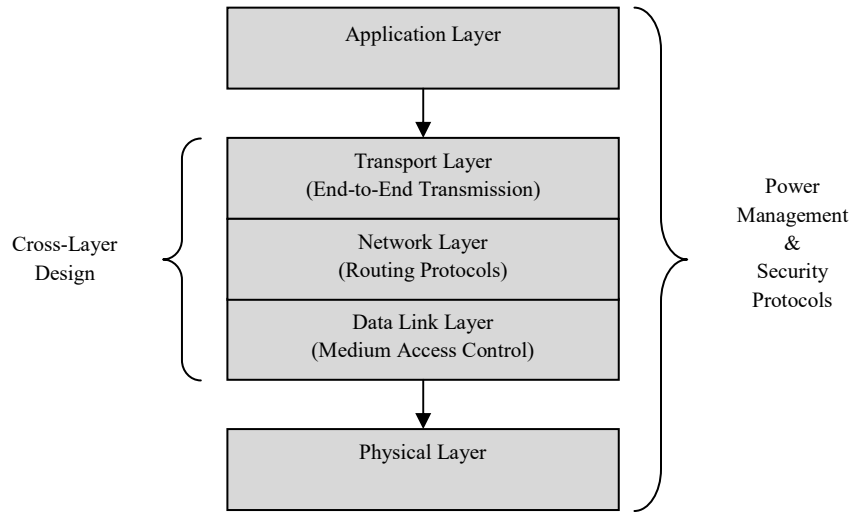


Figure 6. Network Architecture in WSN

2.2.1 Transport layer protocols

Transport layer protocols are responsible for end-to-end transmission of data packets. These protocols include provisions for reliable or unreliable data delivery and for centralized or distributed congestion control. It is imperative that these protocols be designed carefully, since they can quickly overwhelm the constrained WSN. As WSN applications mature, it is important to design and implement efficient protocols for the network architecture to ensure reliability in a data driven network.

Primitives are built into transport layer protocols for reliable data delivery that prompt retransmission of packet(s) in the event of packet drop. This can be achieved by using packet sequence numbers, a series of acknowledgements to confirm delivery of packets, or time-out or round-trip intervals to access packet loss. These primitives allow senders and receivers to account for packets received and packets missed. Such primitives are crucial in WSN applications, which are essentially data driven networks.

Furthermore, transport layer protocols are also responsible for congestion controlled. This is achieved by installing primitives that control transmission rate of motes, to ensure that the rate at which data packets are being transmitted does not overwhelm the underlying network or create a bottleneck at

the base station. Table II classifies some of the transport layer protocols developed for WSN with respect to these primitives ([54], [55], [56]).

Table II. Classification of some Transport Layer Protocols for WSN

WSN Transport Layer Protocols	Reliable	Unreliable	Congestion Control		
			Distributed	Centralized	None
Flush	×		×		
STCP	×		×		
Hop	×		×		
RCRT	×			×	
Wisdén	×				×
Tenet	×				×
RMST	×				×
WRCP		×	×		
IFRC		×	×		
Fusion		×	×		
CODA		×	×		
QCRA		×		×	
ESRT		×		×	
Surge (TinyOS)		×			×
CTP		×			×
RBC		×			×
CentRoute		×			×
Koala		×			×
XLP	×		×		
CRRT	×			×	
CTCP	×		×		
ERTP	×				×
GARUDA	×				×
DTSN	×				×
PHTCCP		×	×		

2.2.2 Routing in Network Layer

As in Wireless Computing, routing protocols can be decomposed into reactive, proactive, cooperative [57] or hybrid protocols. Proactive routing protocols are also known as table-driven protocols since each node maintains a route table for all destinations at all times. Reactive protocols are on-demand routing protocols, which only discover routes to a destination when it is required, outdated or invalid. The obvious tradeoffs include data delivery latency, routing protocol traffic overhead and storage requirements. Data delivery latency is low in networks using proactive routing protocols, since path information is readily available. However, there is overhead incurred by the motes in the WSN, to discover and maintain paths to all destinations, even those that may never be used in the WSN. Routing protocol traffic overhead is minimized in reactive networks, when there is light network traffic and little changes to the network topology. Moreover, storage requirements of proactive routing protocols increase proportionally to the size of the network. In cooperative routing protocols, data is sent to a central entity that can further process the data and disseminate it accordingly [57].

Routing protocols can also be distinguished based on the communication entities involved in the routing protocol. In this perspective, there is node-centric and data-centric routing. In node-centric routing, data is routed between nodes, based on the node addresses. However, in data-centric routing,

attribute based addressing is used, where nodes query for an attribute and only those nodes respond that can satisfy the query. For example, a mote can query for temperature greater than 72°C , and those motes that have temperature readings more than 72°C will respond.

Routing protocols are also designed for specific WSN topologies, classified as flat and hierarchical topologies, as illustrated in Figure 2. On top of these WSN topologies, routing can also be geographical in nature, where packets are routed to motes closest to destination

WSN routing protocols can also be based on different application defined criteria or operation based attributes [57]. These operation based attributes can include, multipath, negotiation-based, query-based, QoS parameters (reliability, data integrity, energy efficiency), coherent [57] based routing techniques to increase reliability, security, etc attributes.

Figure 7 depicts the various characteristics of WSN routing protocols, which can be integrated to design an efficient application specific routing protocol for a WSN.

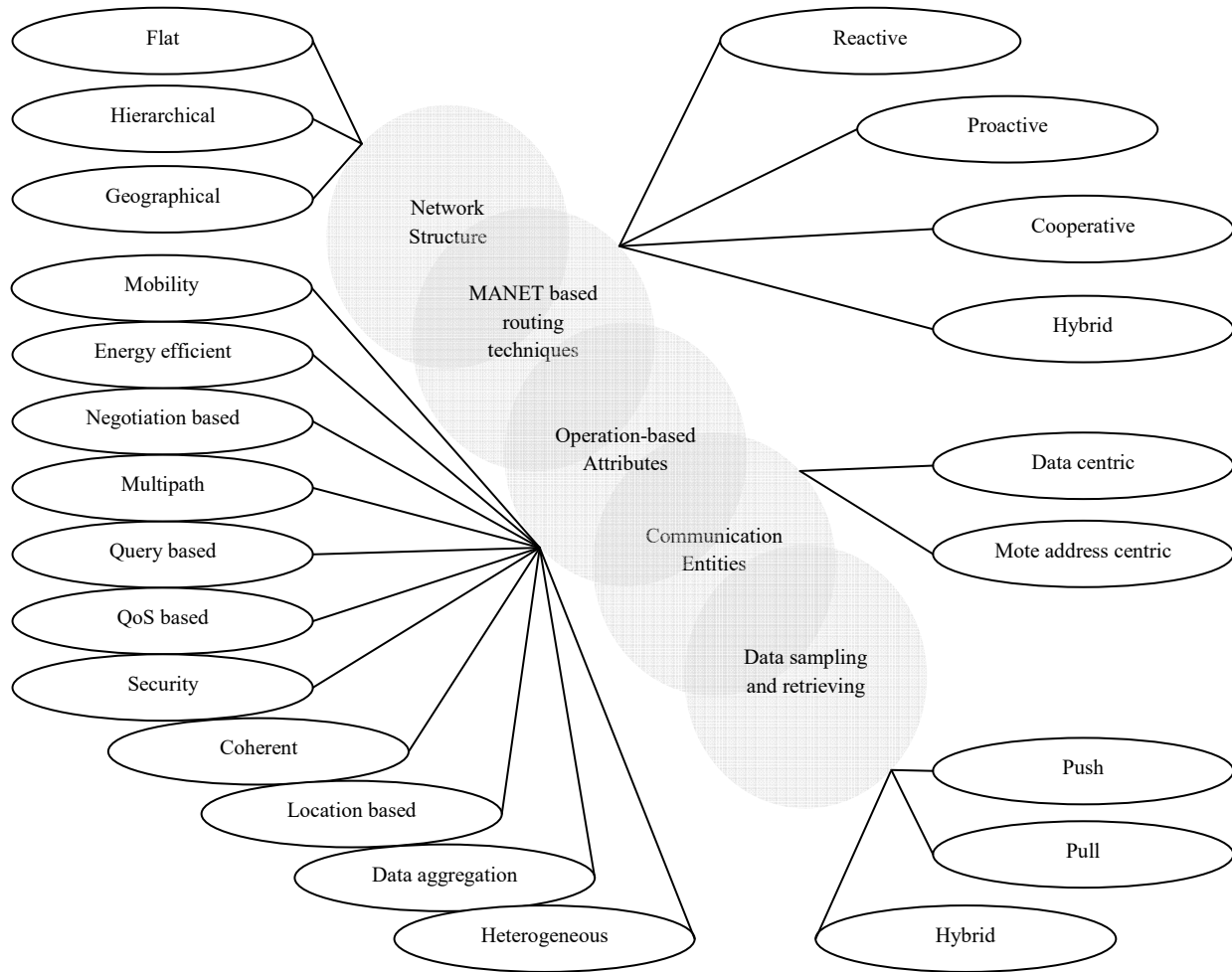


Figure 7. WSN routing protocol characteristics

Multipath features in routing protocols can have a two-fold benefit for WSN. Multipath routing can reduce frequency of updating routes, balance traffic load, increase rate of data delivery [58]. This consequentially increases the lifetime and reliability of WSN. Earlier researchers proposed to use suboptimal routing paths with low energy consumption [59], or select a routing path based on the residual energy of the motes on the path [60]. The authors in [61] study the tradeoff between traffic overhead and reliability using multipath routing in WSN. More recently, [58] use multipath routing to increase lifetime of WSN and reduce collisions for transmission. Furthermore, researchers are utilizing ant colonization optimization, a swarm intelligence optimization technique, due to their proven success [62] in routing protocols for WSN. In [63], ant colonization is used to discover optimal routes to increase network lifetime and reliability, whereas [62] uses multipath routing to reduce congestion at the mote and link level. Authors in [64] survey swarm intelligence based routing protocols in WSN.

Routing protocols that are built on the pull approach, where queries can be propagated, are known as query based routing protocols. Routing algorithms such as Directed Diffusion and Rumor routing [57] are specifically implemented for query-driven data delivery models, where motes that meet the query requirements setup interest gradients or paths along which the data is routed to the sink. Negotiation based routing algorithms suppress redundant data and use negotiation messages to ensure non-redundant data before transmission, for e.g. Sensor Protocols for Information via Negotiation (SPIN) [65]. QoS-based routing ([66], [67] [68], [69], etc.) ensure that routing in WSN meets application defined QoS metrics, of delay, energy, bandwidth, reliability, fault-tolerance, data integrity, etc. Data aggregation is also an important aspect of WSN and their applications, and is often accounted for when designing and developing routing protocols, to ensure efficient performance.

Not only is it important to secure the data and the motes in the WSN, but also the network traffic patterns. Since malicious motes can redirect traffic, inject false data or drop packets and cause havoc in the WSN. Typical uses of asymmetric key cryptography or complicated symmetric key cryptography are infeasible for resource constraint WSN [70]. While there are techniques for implementing security on the data link layer in WSN, researchers are designing novel techniques for securing routing protocols([71], [72], [73]), to increase reliability of WSN.

Table III classifies some WSN routing protocols based on the routing protocol characteristics illustrated in Figure 7([52], [57], [74], [75], [76], [71], [77], [78], [79]).

Table III. Classification of some WSN routing protocols

WSN Routing Protocol	Network Structure	MANET technique	Operation-based attribute	Communication entities	Data sampling and retrieving
SPIN	Flat	Proactive	Data aggregation, negotiation-based, query-based, multipath	Data-centric	Pull
Directed Diffusion	Flat	Reactive	Data aggregation, negotiation-based,	Data-centric	Pull

WSN Routing Protocol	Network Structure	MANET technique	Operation-based attribute	Communication entities	Data sampling and retrieving
			query-based, multipath, localization, coherent		
Rumor Routing	Flat	Reactive	Query-based	Data-centric	Hybrid
TEEN and APTEEN	Hierarchical	Reactive	Data aggregation, localization, energy efficient	Data-centric	Push
LEACH	Hierarchical	Proactive	Data aggregation, localization	Mote address-centric	Push
ACQUIRE	Flat	Reactive	Data aggregation, query-based	Data-centric	-
COUGAR	Flat	Reactive	Data aggregation, query-based	Data-centric	Hybrid
PEGASIS	Hierarchical	Proactive	Data aggregation, localization, energy efficient	Mote address-centric	Push
GAF	Geographical	Proactive	Energy efficient	Mote address-centric	-
SEAD	Geographical	Proactive	Mobility, energy efficient	Mote address-centric	-
SPEED (SNFG)	Geographical	Proactive	QoS, energy efficient	Mote address-centric	-
CHR	Geographical	Proactive	Data aggregation, heterogeneous	Mote address-centric	-
APR	Geographical	Proactive	Security	Mote address-centric	-

2.2.3 Medium Access Control (MAC) protocols at Data Link Layer

The medium access control (MAC) protocols at the data link layer manage how motes access the shared wireless medium and the backoff approach they employ in the event of a collision. In typical Wireless Networks, the radio channel is decomposed into multiple channels using techniques like time division multiple access (TDMA), frequency division multiple access (FDMA) or code division multiple access (CDMA) to allow collision free transmission of multiple message, simultaneously. Traditional WSN motes used a simple single channel radio to conserve energy, which is required to maintain the complex multiple channel radios [80]. However, recent advances in research have given rise to multi-channel radios for WSN motes.

WSN MAC protocols can be broadly classified into schedule based or contention based medium access protocols. Schedule or reservation based MAC protocols devise an assignment that motes can follow for accessing the physical medium to avoid collision. Contention based MAC protocols, do not require an assignment, they define a backoff algorithm for motes to follow, when there is a collision in accessing the physical medium. Schedule based protocols can be further decomposed into synchronous, asynchronous and hybrid protocols.

Since the MAC protocol directly controls the radio on a mote, an important technique used by motes is called duty cycling, where active motes periodically turn their radio off and go to sleep [81]. Synchronous schedule based MAC protocols require time synchronization and topology information to ensure that neighboring motes are active at the same time to communicate. However, asynchronous

protocols do not require such synchronization or topology information and instigate communication between nodes in different active cycles.

Contention based MAC protocols, eliminate the overhead of synchronization or network topology from schedule based MAC protocols. Nodes can transmit packets immediately and retransmit in the event of a collision. A refined approach is to sense the physical medium for transmission, and when the medium is idle then access the medium and transmit the packet. This is the general concept in Carrier Sense Multiple Access (CSMA), which is one of the canonical contention based MAC protocols.

Ye *et al* [82] have identified four significant aspects of wasteful energy consumption, namely, collision, overhearing, control packet overhead and idle listening at the data link layer. When two or more nodes transmit at the same time, their respective packets are corrupted and require retransmissions increasing energy consumption and latency. Energy consumed in overhearing, is energy spent listening for packets destined for other nodes. Sending and receiving control packets without useful data is also wasteful with respect to energy consumed. Listening to the channel even when there are no radio transmissions is considered idle listening, which is an energy consumption overhead [81].

Contention based protocols include techniques like Carrier Sense Multiple Access (CSMA), and offer multiple benefits like low implementation complexity, flexibility in face of mobility and varying traffic patterns [80]. However, schedule based MAC protocols like TDMA have intrinsic energy conserving features since there are no collisions, overhearing and idle listening. Due to the energy conservation necessary in resource constraint WSN, it is important for MAC protocols for WSN to incorporate primitives for conserving energy by reducing collisions, overhearing, reduced control packet overhead and idle listening, without the need for expensive time synchronization.

These MAC protocols require different degrees of collaboration and cooperation amongst the nodes to achieve medium control access. Nodes in the network could work completely independently in a random manner like those in contention-based MAC protocols, to slightly organized slotted MAC protocols and then to the very organized synchronous schedule based protocols like TDMA [80].

Various MAC protocols have been designed and implemented including synchronous duty cycling MAC protocols such as S-MAC [82], T-MAC, TRAMA, SCP, DW-MAC, etc and asynchronous duty cycling MAC protocols such as X-MAC [83], B-MAC, WiseMAC, PW-MAC [81], ASCEMAC [84], etc. An exhaustive MAC protocols survey is presented in [85]. MAC protocols are also being designed to include important features like security [86], power management [87] and QoS [88]. Table IV delineates some of these protocols and their classification ([80], [89]).

Table IV. Classification of some WSN MAC protocols

WSN MAC Protocol	Organization	Channel
SMACS	Frames	FDMA
T-MAC	Slots	Single

WSN MAC Protocol	Organization	Channel
TRAMA	Frames	Single
BuzzBuzz	Random	Single
DW-MAC	Slots	Single
X-MAC	Random	Single
B-MAC	Random	Single
WiseMAC	Random	Single
PW-MAC	Hybrid	Single
S-MAC	Slots	Single
PicoRadio	Random	CDMA

3.0 Wireless Sensor Network Applications and Problem Space

Numerous intrinsic WSN characteristics distinguish it from its peers in other infrastructure-less communication systems [3] and distributed computing environments [90], where their nodes are more powerful (w.r.t. processor and radio) and reliable than WSN motes. Figure 8 [90], illustrates the difference in the problem space of WSN with other computing paradigms, with respect to, functionality of the network nodes, size of the network and spatial-temporal awareness of the nodes in the network.

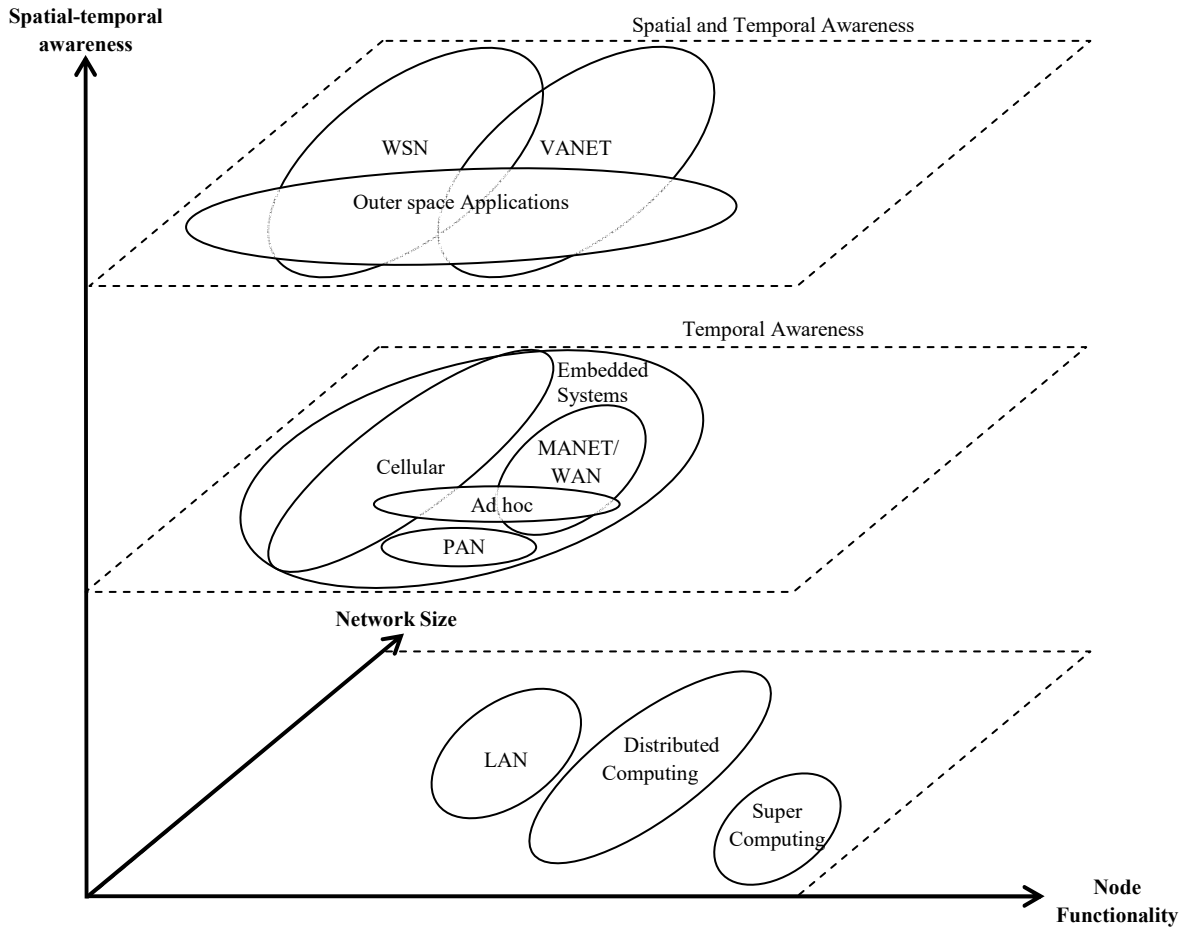


Figure 8. Comparison of problem space of WSN with other computing paradigms

WSN are distinguishable from real-time embedded systems primarily due to the lack of spatial awareness in embedded system nodes. Real-time embedded systems can include various other computing paradigms, especially, infrastructureless mobile environments like Bluetooth personal area networks (PAN), MANET, cellular networks, and ad hoc networks.

WSN are significantly different from traditional distributed and supercomputing paradigms. Distributed computing includes paradigms such as, peer-to-peer (P2P) networks, grid and high-performance computing. Apart from the lack of spatial and temporal awareness of the network nodes, another fundamental difference lies in the functionality of the network nodes. Typically, distributed and supercomputing paradigms consist of high performance powerful nodes, as illustrated in Figure 8.

WSN applications are sensitive to spatial and temporal parameters and can be classified into spatially and temporally related paradigms as follows. Firstly, in the spatial domain, WSN applications can be broadly classified into local and global WSN applications based on the size and coverage of the region of interest of the application. Similarly, WSN applications can also be temporally categorized into continuous, event-driven, query-driven or hybrid applications (cf. Section 2.1.1, Figure 5). This imposes a constraint on when the motes transmit data, either motes transmit data at periodic intervals, or transmit only the data that exceeds application defined triggers or threshold levels, or only those motes transmit data that meets a query criterion or a hybrid approach of these techniques. Figure 9 [49] broadly classifies some WSN applications into these domains.

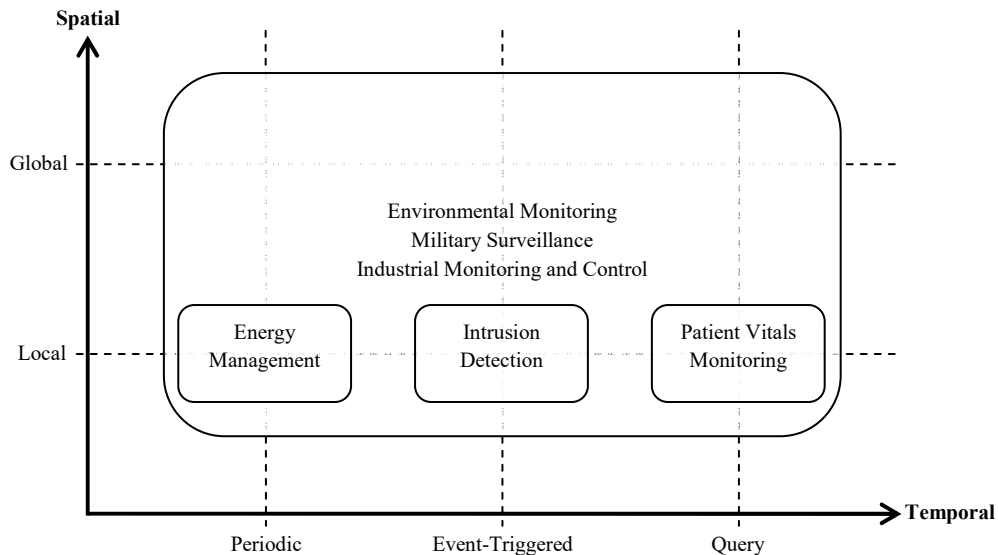


Figure 9. Classification of WSN applications

WSN application operations cannot be discretely categorized into the different temporal domains, since these networks are fundamentally data-driven networks. This implies, events could be triggered in

an otherwise periodic sampling WSN application like environmental monitoring. Furthermore, periodic sampling WSN applications like patient vitals health monitoring could quickly become query-based, if the WSN for patient vitals monitoring is polled for specific vital reading. Therefore, WSN applications are usually designed to operate in multiple temporal domains. Figure 9, illustrates these characteristics of WSN applications and delineates some WSN application examples that accurately fit the domains.

Figure 10 [49] delineates the various design characteristics that form the intrinsic nature and behavior of WSN applications and their motives. This directly influences the design for the components of the network architecture of the WSN. The figure also summarizes the various WSN characteristics we have discussed in this chapter.

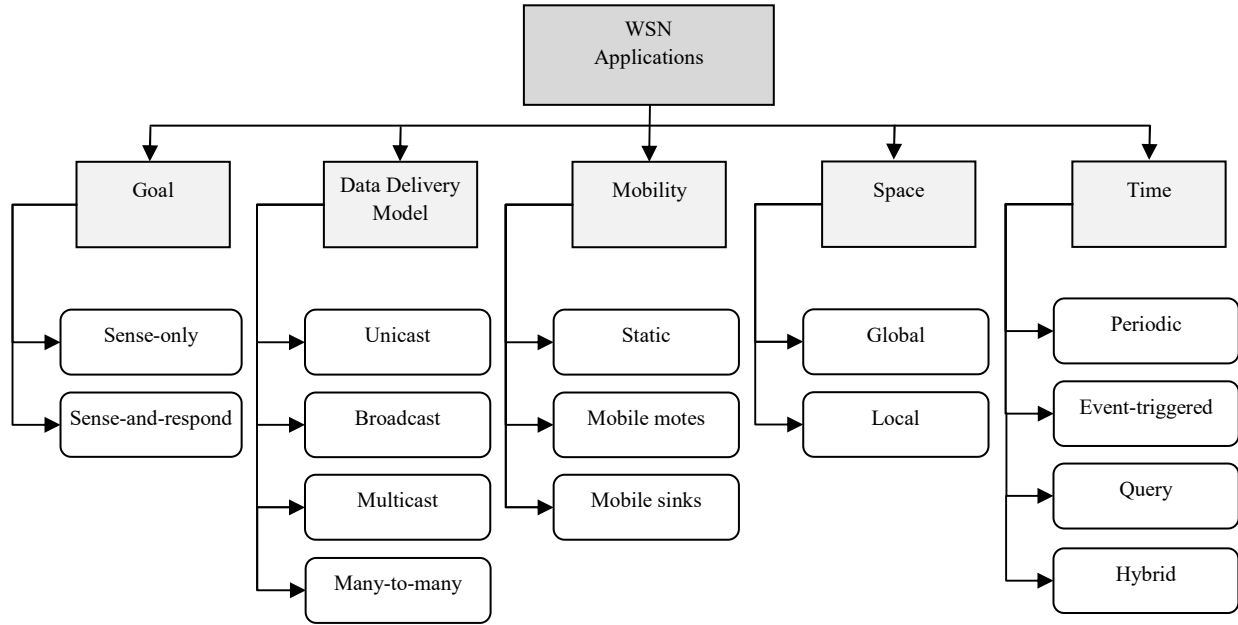


Figure 10. WSN Application Design Criteria

4.0 Programming WSNs

WSNs can only be fully utilized, if WSN programmers have adequate software platforms [49] for efficient and reliable programming of large number of motes. However, due to the lack of programming, debugging, and development environment platforms, WSN programmers are forced to implement application-logic intertwined with low-level WSN implementation issues ([49], [90]). This makes the application code complex, harder to debug and non-scalable for complex application development due to the interdependencies between performance and function [90]. Furthermore, this sort of programming falls out of the expertise of WSN application domain experts [49]. Mottola and Picco [49] discuss various other design criteria and components of WSN programming languages and their affect on the design and development of WSN applications.

Myriad WSN motes exist that differ in functionality as delineated in Table I. Manufacturers often ship an operating system with the motes to ease in the handling of hardware and software components. Most WSN operating

systems (OS) are either event-driven or multithreaded [91]. Multithreaded OS are similar to traditional OS, therefore easier for programmers to manage, however, infeasible for resource constrained WSN. Therefore, lightweight multithreaded OS have been designed and implemented for WSN motes. Event-driven OS do not tax the resources of the WSN motes but pose a learning curve for programmers of WSN applications. Farooq and Kunz [92] survey and classify operating systems such as TinyOS, Contiki, and MANTIS, in WSN.

In the following section, we will discuss TinyOS, an operating system for WSN, and consider a component-based approach to programming applications for WSNs using TinyOS.

4.1 TinyOS: An Operating System for WSN Motes

Operating systems (OS) are programs that coordinate all other programs, such as process scheduling and memory access, in a computer. The typical size of operating systems for embedded devices ranges in the order of megabytes; however, TinyOS is a much smaller operating system for WSN motes, approximately 400 bytes [93]. It is implemented in nesC [92], a variant of C. TinyOS is different from traditional OS, since it provides a framework and a set of component for programming motes to *build* application-specific operating system for a WSN application. Therefore, there is an operating system for each application. Typical TinyOS applications can range from approximately 15 Kbytes to more complex applications in the range of 64 Kbytes, of these the core operating system is about 400 bytes [93]. The networking architecture is encapsulated in Active Message interfaces, which are also implemented in nesC and offer fundamental communication primitives to TinyOS applications ([49], [93]). Surge is the transport layer protocol implemented in TinyOS, which is unreliable and offers no primitives for congestion control [54]. Berkley MAC (B-MAC) is the asynchronous schedule based MAC protocol implemented in TinyOS [94], with interfaces available to change or adapt the MAC protocol.

TinyOS is a component-based programming model, where components are software abstractions of hardware components [93]. Therefore, a TinyOS application, which is really an application-specific operating system, is built on top of these reusable components, [93], wiring them together to provide or use services. Components use interfaces to encapsulate a set of services [93]. For example, turning an LED on is a service that can be modeled by using the LED interface. Components use commands and events for inter-component communication. Commands to a component are requests to start a service, whereas events are signaled to mark the completion of the service at the component [93].

Programming in nesC consists of a configuration and module file. The configuration file connects components together, whereas the module file contains the implementation, by *providing* or *using* interfaces. Figures 11 and 12 illustrate a “hello world” TinyOS application, called Blink [95], where the red LEDs on a mote are toggled every 1,000 ms.


```
configuration Blink {}

implementation {
    components Main, BlinkM, SingleTimer, LedsC;
    Main.StdControl -> SingleTimer.StdControl;
    Main.StdControl -> BlinkM.StdControl;
    BlinkM.Timer -> SingleTimer.Timer;
    BlinkM.Leds -> LedsC;
}
```

Figure 11. Blink.nc configuration file

The `Blink.nc` configuration file shows how the components are wired together. A TinyOS application starts by executing the `init` command of the `StdControl` interface of the `Main` component. `Blink` configuration file wires the `Main` component's `StdControl` to `SingleTimer` and `Blink` component `StdControl`. This binds the implementation of `Main.StdControl` to `SingleTimer.StdControl` and `BlinkM.StdControl`. The `BlinkM.StdControl` interface is implemented in the module file, `BlinkM.nc`. Similarly, the `Timer` and `Leds` interfaces used in `Blink` actually invoke the `Timer` interface provided by component `SingleTimer`, and `Leds` interface of component `LedsC`, respectively. Note, `SingleTimer` and `Leds` provide software abstraction of the hardware components for the clock and LEDS.

The configuration file delineates how control of execution is passed amongst components. The module file (`BlinkM.nc`) provides the application-specific implementation by providing and using interfaces. `BlinkM.StdControl.init()` initializes the application and `BlinkM.StdControl.start()` starts the component. When `Blink` starts, it sets the timer to repeat every 1,000 ms. In the event-driven operating system, since, there is no other code to execute, `Blink` waits until event `Timer.fired()` is triggered. When the event is triggered, `Blink` toggles the red LEDS.

The application is compiled for the target platform, any TinyOS compatible device, e.g. micaz motes, and flushed on the mote hardware using a programming board, which interfaces a mote with the development environment on a computer or laptop.

```
module BlinkM {
  provides {
    interface StdControl;
  }
  uses {
    interface Timer;
    interface Leds;
  }
}
implementation {

  command result_t StdControl.init() {
    call Leds.init();
    return SUCCESS;
  }

  command result_t StdControl.start() {
    return call Timer.start(TIMER_REPEAT, 1000);
  }

  command result_t StdControl.stop() {
    return call Timer.stop();
  }

  event result_t Timer.fired()
  {
    call Leds.redToggle();
    return SUCCESS;
  }
}
```

Figure 12. BlinkM.nc module file

5.0 Summary

Wireless Sensor Networks typically consist of large number of heterogeneous motes organized in a hierarchical manner, to monitor and, or control a region of interest, based on ambient environment factors. Generally, the number of motes deployed in the region increases and the resources decrease as you move down the hierarchy. The mobile, infrastructure-less WSN are able to truly achieve untethered communications in dangerous or mundane regions for autonomous data collection, analysis and response via actuators. However, the greatest resource limitation of WSN is the power source, which must be optimally used or alternatively harnessed to ensure network lifetime.

There is a lack of standards for traditional network concepts, like architecture, topology, routing and security, since WSN are application specific in nature. Due to the different types of motes and wireless sensor networks, there are interoperability issues amongst the heterogeneous hardware units [96]. Further limitations of WSN are attributed to the lack of software for programming and debugging motes and WSN ([49] and [90]).

Energy conservation and power management in the radio and network communications are the underlying design criteria for protocols and primitives in WSN. However, Bose and Helal [48], show that advances in technology have allowed low-power radios, like the Atmel Zlink RCB [48] for effective use in WSN and now attention needs to be on the sensor sampling technique, which consumes significantly more energy than network and application costs. Anastasi *et al* [94] present a survey of energy conservation techniques in WSN.

Furthermore, apart from energy conservation, security challenges in WSN are a make it or break it criteria in the wide-spread use of WSN in all envisioned domains. Boyle and Newe [97] compare some security protocols in use today in WSN, like SPIN [98], consisting of SNEP (Secure Network Encryption Protocol) and μ TESLA (micro version of Timed Efficient Stream Loss-tolerant Authentication), LEAP [99] and TinySec [100], which aim at building security features into network protocols.

As recent routing protocols have gained from swarm intelligence, Kulkarni *et al* [101] discuss the use of other computational intelligence paradigms like neural networks, fuzzy logic, evolutionary algorithms, reinforcement learning and artificial immune system in designing and implementing effective and efficient protocols and primitives for WSN operations like routing, deployment, localization, security, scheduling, data aggregation and QoS management.

To summarize, in this chapter we considered the characteristics and features of Wireless Sensor Networks and their motes. Furthermore, we reviewed the fundamental concepts in wireless sensor networks pertaining to architecture, topology, data delivery models, transport, routing and data link layer protocols. We also presented a classification of some of the protocols for transport, routing and data link layers. We concluded with a discussion of TinyOS, an operating system for WSN motes, and exemplified programming WSN motes in nesC.

References and Bibliography

- [1] "History of Innovation," 1995-2012. [Online]. Available: <http://www.ti.com/corp/docs/company/history/sensortimelinelowbandwidth.shtml>. [Accessed 6 January 2012].
- [2] H. Weinberg, "How They Work: MEMS (Micro Electro-Mechanical Systems) Technology," Sensorland, [Online]. Available: <http://www.sensorland.com/HowPage023.html>. [Accessed 6 January 2012].
- [3] I. F. Akyildiz, W. Su, Y. Sankarasubramainiam and E. Cayirci, "A Survey on Sensor Networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102-114, 2002.
- [4] I. F. Akyildiz and I. H. Kasimoglu, "Wireless sensor and actor networks: research challenges," *Ad Hoc Networks*, vol. 2, no. 4, pp. 351-367, 2004.
- [5] F. Xia, "QoS Challenges and Opportunities in Wireless Sensor/Actuator Networks," *Sensors*, vol. 8, no. 2, pp. 1099-1110, 2008.
- [6] I. F. Akyildiz, T. Melodia and K. Chowdury, "A survey on wireless multimedia sensor networks," *Computer Networks*, vol. 51, no. 4, pp. 921-960, 2007.
- [7] L. Lamont, M. Toulgoat, M. Deziel and G. Patterson, "Tiered Wireless Sensor Network Architecture for Military Surveillance Applications," in *International Conference on Sensor Technologies and Applications (SENSORCOMM)*, Nice/Saint Laurent du Var, France, 2011.
- [8] M. A. Antepi, S. Z. Gurbuz and E. Uysal-Biyikoglu, "Ferromagnetic target detection and localization with a wireless sensor network," in *Military Communications Conference (MILCOM)*, San Jose, CA, 2010.
- [9] J. Teng, H. Snoussi and C. Richard, "Decentralized Variational Filtering for Target Tracking in Binary Sensor Networks," *IEEE Transactions on Mobile Computing*, vol. 9, no. 10, pp. 1465-1477, 2010.
- [10] P. Medagliani, G. Ferrari, V. Gay and J. Leguay, "Cross-layer design and analysis of WSN-based mobile target detection systems," *Ad Hoc Networks, In Press*, vol. July, 2011.
- [11] L. Krishnamurty, R. Adler, P. Buonadonna, J. Chhabra, M. Flanigan, N. Kushalnagar, L. Nachman and M. Yarvis, "Design and deployment of industrial sensor networks: experiences from a semiconductor plant and the north sea," in *3rd International Conference on Embedded Networked Sensor Systems (SenSys 05)*, San Diego, CA, 2005.
- [12] P. Sikka, P. Corke, P. Valencia, C. Crossman, D. Swain and G. Bishop-Hurley, "Wireless adhoc sensor and actuator networks on the farm," in *International Conference on Information Processing in Sensor Networks*, Nashville, TN, 2006.
- [13] W.-Z. Song, R. Huang, M. Xu, A. Ma, B. Shirazi and R. LaHusen, "Air-dropped sensor network for real-time high-fidelity volcano monitoring," in *International Conference on Mobile System, Applications and Services (MobiSys 09)*, Krakow, Poland, 2009.
- [14] M. Ceriotti, L. Mottola, G. P. Picco, A. L. Murphy, M. Corra, M. Pozzi, D. Zonta and P. Zanon, "Monitoring heritage buildings with wireless sensor networks: The Torre Aquila deployment," in *International Conference on Information Processing in Sensor Networks (IPSN)*, San Francisco, CA, 2009.
- [15] J. Ko, J. H. Lim, R. Muvaloiu-E, A. Terzis, G. M. Masson, T. Gao, W. Destler, L. Selavo and R. P. Dutton, "MEDiSN: Medical emergency detection in sensor networks," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 10, no. 1, 2010.
- [16] Y. Chen, W. Shen, H. Huo and Y. Xu, "A Smart Gateway for Health Care System Using Wireless Sensor Network," in *International Conference on Sensor Technologies and Applications (SENSORCOMM 10)*,

Venice, Italy, 2010.

- [17] N. Ota, S. Ahrens, A. Redfern, P. Wright and X. Yang, "An Application-Driven Architecture for Residential Energy Management with Wireless Sensor Networks," in *IEEE International Conference on Mobile Adhoc and Sensor Systems*, Vancouver, Canada, 2006.
- [18] H. Song, S. Zhu and G. Cao, "SVATS: A Sensor-Network-Based Vehicle Anti-Theft System," in *IEEE Conference on Computer Communications*, Phoenix, AZ, 2008.
- [19] M. Tubaishat, P. Zhuang, Q. Qi and Y. Shang, "Wireless sensor networks in intelligent transportation systems," *Wireless Communications and Mobile Computing, Special Issue: Distributed Systems of Sensors and Actuators*, vol. 9, no. 3, pp. 287-302, 2009.
- [20] T. Vladimirova, C. P. Bridges, J. R. Paul, S. A. Malik and M. N. Sweeting, "Space-based wireless sensor networks: Design issues," in *IEEE Aerospace Conference*, Big Sky, MT, 2010.
- [21] D. Culler, J. Hill, M. Horton, K. Pister, R. Szewczyk and A. Woo, "MICA: The Commercialization of Microsensor Motes," *Sensors Magazine*, 1 April 2002. [Online]. Available: <http://www.sensorsmag.com/networking-communications/mica-the-commercialization-microsensor-motes-1070>. [Accessed 1 February 2012].
- [22] W. Merrill and W. Kaiser, "Wireless Integrated Network Sensors (WINS) Next Generation," Airforce Research Laboratory, Rome, New York, 2004.
- [23] "Wireless Applications - Moteiv launches wireless mote with interface to mobile devices," *Sensors*, 11 May 2007. [Online]. Available: <http://www.sensorsmag.com/wireless-applications/news/moteiv-launches-wireless-mote-with-interface-mobile-devices-2523>. [Accessed 7 January 2012].
- [24] J. Hill, M. Horton, R. Kling and L. Krishnamurthy, "The platforms enabling wireless sensor networks," *Communications of the ACM*, vol. 47, no. 6, pp. 41-46, 2004.
- [25] M. Healy, T. Newe and E. Lewis, "Wireless Sensor Node hardware: A review," in *IEEE Sensors*, Lecce, Italy, 2008.
- [26] J. Yick, B. Mukherjee and D. Ghosal, "Wireless sensor network survey," *Computer Networks(Elsevier)*, vol. 52, no. 12, pp. 2292-2330, 2008.
- [27] Z. Sun and I. F. Akyildiz, "Magnetic Induction Communications for Wireless Underground Sensor Networks," *IEEE Transactions on Antennas and Propagation*, vol. 58, no. 7, pp. 2426-2435, 2010.
- [28] P. Bhuvaneswari, R. Balakumar, V. Vaidehi and P. Balamuralidhar, "Solar Energy Harvesting for Wireless Sensor Networks," in *International Conference on Computational Intelligence, Communication Systems and Networks (CICSYN)*, Indore, India, 2009.
- [29] W. K. Seah, Z. A. Eu and H.-P. Tan, "Wireless sensor networks powered by ambient energy harvesting (WSN-HEAP) - Survey and challenges," in *International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology (Wireless VITAE)*, Aalborg, Denmark, 2009.
- [30] T. Bokareva, "Mini Hardware Survey," [Online]. Available: http://www.cse.unsw.edu.au/~sensor/hardware/hardware_survey.html. [Accessed 7 January 2012].
- [31] T. Uchiyama, Y. Uehara, M. Mori, Saito, Hiroki and Y. Tobe, "A System for Analyzing Life Rhythm Using Wireless Sensors on Mules," in *International Conference on Mobile Data Management (MDM)*, Nara, Japan, 2006.
- [32] K. Romer and F. Mattern, "The Design Space of Wireless Sensor Networks," *IEEE Wireless*

Communications, vol. 11, no. 6, pp. 54-61, 2004.

- [33] T. Naumowicz, R. Freeman, H. Kirk, B. Dean, M. Calsyn, A. Liers, A. Braendle, T. Guilford and J. Schiller, "Wireless Sensor Network for habitat monitoring on Skomer Island," in *IEEE Conference on Local Computer Networks (LCN)*, Denver, CO, 2010.
- [34] C.-L. Fok, G.-C. Roman and C. Lu, "Towards a flexible global sensing infrastructure," *Special issue on the workshop on wireless sensor network architecture*, vol. 4, no. 3, pp. 1-6, 2007.
- [35] J. Polastre, R. Szewczyk, A. Mainwaring, D. Culler and J. Anderson, "Analysis of wireless sensor networks for habitat monitoring," in *Wireless Sensor Networks*, Norwell, MA, Kluwer Academic Publishers, 2004, pp. 399-423.
- [36] P. Schoonmaker and W. Luscombe, "Habitat Monitoring: An Approach for Reporting Status and Trends For State Comprehensive Wildlife Conservation Strategies," 22 March 2005. [Online]. Available: http://www.defenders.org/resources/publications/programs_and_policy/biodiversity_partners/habitat_monitoring.pdf. [Accessed 8 January 2012].
- [37] R. Agarwal, R. Martinez-Catala, S. Harte, C. Segard and B. O'Flynn, "Modeling Power in Multi-functionality Sensor Network Applications," in *International Conference on Sensor Technologies and Applications (SENSORCOMM)*, Cap Esterel, France, 2008.
- [38] W. Dargie, "Dynamic Power Management in Wireless Sensor Networks: State-of-the-Art," *IEEE Journal on Sensors*, vol. PP, no. 99, 2011.
- [39] K. Sohrabi, J. Gao, V. Ailawadhi and G. J. Pottie, "Protocols for self-organization of a wireless sensor network," *IEEE Personal Communications*, vol. 7, no. 5, pp. 16-27, 2000.
- [40] O. Saglam and M. E. Dalkili, "A Self Organizing Multihop Clustering Protocol for Wireless Sensor Networks," in *International Conference on Mobile Ad-hoc and Sensor Networks (MSN)*, Fujian, China, 2009.
- [41] R. Kacimi, R. Dhaou and A. Beylot, "Energy-Aware Self-Organization Algorithms for Wireless Sensor Networks," in *IEEE Global Telecommunications Conference (GLOBECOM)*, New Orleans, Louisiana, 2008.
- [42] K. Berman, F. Annexstein and A. Ranganathan, "Dominating connectivity and reliability of heterogeneous sensor networks," in *IEEE International Conference on Pervasive Computing and Communications Workshop (PerCom)*, Pisa, Italy, 2006.
- [43] A. Abbasi, M. Younis and U. Baroudi, "Restoring Connectivity in Wireless Sensor-Actor Networks with Minimal Topology Changes," in *International Conference on Communications (ICC)*, Cape Town, South Africa, 2010.
- [44] A. Ghosh and S. K. Das, "Coverage and connectivity issues in wireless sensor networks: A survey," *Pervasive and Mobile Computing*, vol. 4, no. 3, pp. 303-334, 2008.
- [45] P. Wang, Z. Sun, M. Vuran, M. A. Al-Rodhaan, A. Al-Dhelaan and I. F. Akyildiz, "On network connectivity of wireless sensor networks for sandstorm monitoring," *Computer Networks*, vol. 55, no. 5, pp. 1150-1157, 2011.
- [46] M. Lu, J. Wu, M. Cardei and M. Li, "Energy-efficient connected coverage of discrete targets in wireless sensor networks," *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 4, no. 3-4, pp. 137-147, 2009.
- [47] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless and C. Gill, "Integrated coverage and connectivity configuration in wireless sensor networks," in *International Conference on Embedded Networked Sensor Systems (SenSys)*, Los Angeles, CA, 2003.

- [48] R. Bose and A. Helal, "Sensor-Aware Adaptive Push-Pull Query Processing in Wireless Sensor Networks," in *International Conference on Intelligent Environments (IE)*, Kuala Lumpur, Malaysia, 2010.
- [49] L. Mottola and G. P. Picco, "Programming wireless sensor networks: Fundamental concepts and state of the art," *ACM Computing Surveys (CSUR)*, vol. 43, no. 3, 2011.
- [50] S. Tilak, N. B. Abu-Ghazaleh and W. Heinzelman, "A taxonomy of wireless micro-sensor network models," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 6, no. 2, pp. 28-36, 2002.
- [51] L. Villalba, O. A. A. Cabrera and C. Abbas, "Routing Protocols in Wireless Sensor Networks," *Sensors*, vol. 9, no. 11, pp. 8399-8421, 2009.
- [52] K. Akkaya and M. Younis, "A survey on routing protocols for wireless sensor networks," *Elsevier Ad Hoc Networks*, vol. 3, no. 3, pp. 325-349, 2005.
- [53] D. Niculescu, "Communication paradigms for sensor networks," *Communications Magazine*, vol. 43, no. 3, pp. 116-122, 2005.
- [54] J. Paek and R. Govindan, "RCRT: rate-controlled reliable transport for wireless sensor networks," in *Proceedings of the 5th international conference on Embedded networked sensor systems (SenSys '07)*, Sydney, Australia, 2007.
- [55] A. J. D. Rathnayaka and V. M. Potdar, "Wireless Sensor Network transport protocol: A critical review," *Journal of Network and Computer Applications*, 2011.
- [56] M. Vuran and I. Akyildiz, "XLP: A Cross-Layer Protocol for Efficient Communication in Wireless Sensor Networks," *IEEE Transactions on Mobile Computing*, vol. 9, no. 11, pp. 1578-1591, November 2010.
- [57] J. N. Al-Karaki and A. E. Kamal, "Routing techniques in wireless sensor networks: a survey," *IEEE Wireless Communications*, vol. 11, no. 6, pp. 6-28, 2004.
- [58] Z. Wang, E. Bulut and B. K. Szymanski, "Energy Efficient Collision Aware Multipath Routing for Wireless Sensor Networks," in *IEEE International Conference on Communications*, Dresden, Germany, 2009.
- [59] R. C. Shah and J. M. Rabaey, "Energy aware routing for low energy ad hoc sensor networks," in *IEEE Wireless Communications and Networking Conference*, Orlando, Florida, 2002.
- [60] J. H. Chang and L. Tassiulas, "Maximum lifetime routing in wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 12, no. 4, pp. 609-619, 2004.
- [61] S. Dulman, T. Nieberg, J. Wu and P. Havinga, "Trade-off between traffic overhead and reliability in multipath routing for wireless sensor networks," in *IEEE Wireless Communications and Networking*, New Orleans, LA, 2003.
- [62] K. Saleem, N. Fisal, S. Hafizah, S. Kamilah and R. Rashid, "A Self-Optimized Multipath Routing Protocol for Wireless Sensor Networks," *International Journal of Recent Trends in Engineering (IJRTE)*, vol. 2, no. 1, pp. 93-97, 2009.
- [63] S. Okdem and D. Karaboga, "Routing in Wireless Sensor Networks Using an Ant Colony Optimization (ACO) Router Chip," *Sensors*, vol. 9, no. 2, pp. 909-921, 2009.
- [64] M. Saleem, G. A. Di Caro and M. Farooq, "Swarm intelligence based routing protocol for wireless sensor networks: Survey and future directions," *Elsevier Information Sciences*, vol. 181, no. 20, pp. 4597-4624, 2011.
- [65] J. Kulik, W. Heinzelman and H. Balakrishnan, "Negotiation-based protocols for disseminating information in wireless sensor networks," *Wireless Networks*, vol. 8, no. 2/3, pp. 169-185, 2002.

- [66] X. Huang and Y. Fang, "Multiconstrained QoS multipath routing in wireless sensor networks," *Wireless Networks*, vol. 14, no. 4, pp. 465-478, 2008.
- [67] B. Kusy, H. J. Lee, M. Wicke, N. Milosavljevic and L. Guibas, "Predictive QoS routing to mobile sinks in wireless sensor networks," in *International Conference on Information Processing in Sensor Networks (IPSN)*, San Francisco, CA, 2009.
- [68] J. Ben-Othman and B. Yahya, "Energy efficient and QoS based routing protocol for wireless sensor networks," *Elsevier Journal of Parallel and Distributed Computing*, vol. 70, no. 8, pp. 849-857, 2010.
- [69] Y. Chen, N. Nasser, T. El Salti and H. Zhang, "A multipath QoS routing protocol in wireless sensor networks," *International Journal of Sensor Networks*, vol. 7, no. 4, pp. 207-216, 2010.
- [70] Y. Zhou, Y. Fang and Y. Zhang, "Securing wireless sensor networks: a survey," *IEEE Communications Surveys & Tutorials*, vol. 10, no. 3, pp. 6-28, 2008.
- [71] J.-P. Sheu, J.-R. Jiang and C. Tu, "Anonymous Path Routing in Wireless Sensor Networks," in *IEEE International Conference on Communications (ICC)*, Beijing, China, 2008.
- [72] Z. Zhang, C. Jiang and J. Deng, "A secure anonymous path routing protocol for wireless sensor networks," in *IEEE International Conference on Wireless Communications, Networking and Information Security (WCNIS)*, Beijing, China, 2010.
- [73] T. Zahariadis, H. Leligou, S. Voliotis, S. Maniatis, P. Trakadas and P. Karkazis, "Energy-aware secure routing for large wireless sensor networks," *World Scientific and Engineering Academy and Society (WSEAS) Transactions on Communications*, vol. 8, no. 9, pp. 981-991, 2009.
- [74] S. Singh, M. Singh and D. Singh, "Routing Protocols in Wireless Sensor Networks-A Survey," *International Journal of Computer Science and Engineering Survey (IJCES)*, vol. 1, no. 2, 2010.
- [75] A. Koliouisis and J. Sventek, "Proactive vs reactive routing for wireless sensor networks," Department of Computing Science, University of Glasgow, Glasgow, Scotland, 2007.
- [76] X. Du and F. Lin, "Improving routing in sensor networks with heterogeneous sensor nodes," in *Vehicular Technology Conference*, Stockholm, Sweden, 2005.
- [77] H. Kim, T. Abdelzaher and W. Kwon, "Minimum-energy asynchronous dissemination to mobile sinks in wireless sensor networks," in *1st international conference on Embedded networked sensor systems (SenSys '03)*, Los Angeles, CA, 2003.
- [78] Y. Yao and J. Gehrke, "The cougar approach to in-network query processing in sensor networks," *ACM SIGMOD Records*, vol. 31, no. 3, pp. 9-18, 2002.
- [79] S. Lindsey and C. Raghavendra, "PEGASIS: Power-efficient gathering in sensor information systems," in *IEEE Aerospace Conference*, Big Sky, MT, 2002.
- [80] K. Langendoen and G. Halkes, "Energy Efficient Medium Access Control," in *Embedded Systems Handbook*, CRC Press, 2005, pp. 34.1-34.29.
- [81] L. Tang, S. Yanjun, O. Gurewitz and D. B. Johnson, "PW-MAC: An energy-efficient predictive-wakeup MAC protocol for wireless sensor networks," in *IEEE INFOCOM*, Shanghai, China, 2011.
- [82] W. Ye, J. Heidemann and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," in *INFOCOM 2002, 21st Annual Joint Conference of IEEE Computer and Communications Societies*, New York, 2002.
- [83] M. Buettner, G. V. Yee, E. Anderson and R. Han, "X-MAC: a short preamble MAC protocol for duty-cycled wireless sensor networks," in *Proceedings of 4th International Conference on Embedded Networked Sensor*

Systems (SenSys), Boulder, CO, 2006.

- [84] Q. Ren and Q. Liang, "An Energy-Efficient MAC Protocol for Wireless Sensor Networks," in *IEEE Globecom*, St. Louis, MO, 2005.
- [85] A. Bachir, M. Dohler, T. Watteyne and K. Leung, "MAC Essentials for Wireless Sensor Networks," *IEEE Communications Surveys & Tutorials*, vol. 12, no. 2, pp. 222-248, 2010.
- [86] P. Rao, R. Rani, S. Sankar, K. Pradeep, N. Kumar and V. Kumar, "Secure MAC for Wireless Sensor Networks through RBFNN," *International Journal of Engineering Science and Technology*, vol. 2, no. 8, pp. 3510-3514, 2010.
- [87] Q. Hu and Z. Tang, "ATPM: An Energy Efficient MAC Protocol with Adaptive Transmit Power Scheme for Wireless Sensor Networks," *Journal of Multimedia*, vol. 6, no. 2, pp. 122-128, 2011.
- [88] M. Yigitel, O. Incel and C. Ersoy, "QoS-aware MAC protocols for wireless sensor networks: A survey," *Elsevier Computer Networks*, vol. 8, no. 1, pp. 1982-2004, 2011.
- [89] K. Langendoen, "The MAC Alphabet Soup served in Wireless Sensor Networks - Taxonomy," Delft University of Technology, 16 March 2011. [Online]. Available: <http://www.st.ewi.tudelft.nl/~koen/MACsoup/taxonomy.php>. [Accessed 12 July 2012].
- [90] R. Sugihara and R. K. Gupta, "Programming models for sensor networks: A survey," *ACM Transactions on Sensor Networks (TOSN)*, vol. 4, no. 2, 2008.
- [91] M. Farooq and T. Kunz, "Operating Systems for Wireless Sensor Networks: A survey," *Sensors*, vol. 11, no. 6, pp. 5900-5930, 2011.
- [92] D. Gay, P. Levis, R. von Behren, M. Welsh, E. Brewer and D. Culler, "The nesC language: A holistic approach to networked embedded systems," in *In Proceedings of the ACM SIGPLAN 2003 conference on Programming language design and implementation (PLDI '03)*, New York, 2003.
- [93] P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer and D. Culler, "TinyOS: An Operating System for Sensor Networks," in *Ambient Intelligence*, New York, Springer, 2005, pp. 115-148.
- [94] G. Anastasi, M. Conti, M. Di Francesco and A. Passarella, "Energy conservation in wireless sensor networks: A survey," *Elsevier Ad Hoc Networks*, vol. 7, no. 3, pp. 537-568, 2009.
- [95] TinyOS, "TinyOS Tutorial Lesson 1: Getting Started with TinyOS and nesC," 9 September 2003. [Online]. Available: <http://www.tinyos.net/tinyos-1.x/doc/tutorial/lesson1.html>. [Accessed 8 Jan 2012].
- [96] R. Thusu, "Wireless Sensor Use Is Expanding in Industrial Applications," *Sensors*, 1 June 2010. [Online]. Available: <http://www.sensormag.com/networking-communications/wireless-sensor/wireless-sensor-use-is-expanding-industrial-applications-7212>. [Accessed 8 January 2012].
- [97] D. Boyle and T. Newe, "Security Protocols for Use with Wireless Sensor Networks: A Survey of Security Architectures," in *International Conference on Wireless and Mobile Communications (ICWMC)*, Guadeloupe, French Caribbean, 2007.
- [98] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen and D. Culler, "SPINS: security protocols for sensor networks," *Wireless Networks*, vol. 8, no. 5, pp. 521-534, 2002.
- [99] S. Zhu, S. Setia and S. Jajodia, "LEAP: efficient security mechanisms for large-scale distributed sensor networks," in *ACM Conference on Computer and Communications Security*, Washington D.C., 2003.
- [100] C. Karlof, N. Sastry and D. Wagner, "TinySec: a link layer security architecture for wireless sensor networks," in *International Conference on Embedded Networked Sensor Systems (SenSys)*, Baltimore, MD,

2004.

- [101] "Computational Intelligence in Wireless Sensor Networks: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 13, no. 1, pp. 68-96, 2011.