PEWSN:

Power Equilibrium Wireless Sensor Network

A dissertation

submitted by

Almir Davis

In partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

in

Electrical Engineering

TUFTS UNIVERSITY

February 2013

Advisor: Professor Hwa Chang

ABSTRACT

In this work we present the Power Equilibrium Wireless Sensor Network (PEWSN), a novel wireless sensor network architecture designed to extend the lifetime of wireless sensor networks (WSNs) and at the same time ensures the predictable and bounded payload latency. To achieve this objective, PEWSN strives to reach power equilibrium using highly efficient PEWSN protocol features such as PEWSN-specific TDMA schedule, smart cluster head selection, adjustable communication bandwidth and contention-free reliable communication. Our architecture also takes advantage of energy harvesting capabilities of each sensor. PEWSN architecture supports network metamorphism which means the network is capable of changing its primary sensing objectives and methodologies while running. PEWSN supports seamless network configuring and re-configuring as well seamless repair and restructure. PEWSN comes with custom-made OMNET++-based network simulator and MATLAB-based script.

The predictable latency along with the extended network lifetime makes PEWSN architecture suitable for a wide range of commercial and industrial applications such as health/environmental/habitat monitoring, inventory tracking, and process supervision. The architecture is also suitable for military indoor and outdoor applications such as intrusion detection, critical infrastructure monitoring, ground/port surveillance, and guidance in case of unexpected events.

ACKNOWLEDGEMENTS

I would first like to thank my adviser, Professor Hwa Chang, for all of the support that he has given me during my time at Tufts University. I am especially grateful for the freedom he has allowed me in defining and pursuing research which interests me. It has been a great experience working with such a great person and great advisor.

I would also like to thank the members of my dissertation defense committee, Professor Sameer Sonkusale, Professor Douglas Preis, Dr. Fred Zarinetchi, and Professor Peter Miraglia for their interest, time, and feedback.

I have enjoyed working with all the members of the Tufts Wireless Laboratory. I have deep personal feelings for the Lab that I help co-found and hope that my involvement will continue beyond graduation.

I would also like to thank my coworkers at The Charles Stark Draper Laboratory and EMC Corporation, who were always there for me when I needed consultation and advice.

Special gratitude goes to my wonderful family. I cannot find the words to thank them enough for all the support, love, and commitment I receive from them every day, every month, every year. *My family is my strength, my family is my support, my family is my motivator, my family exist I exist*.

Finally, my greatest thanks go to A.E., my inspiration, my guidance.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

WSN	-	Wireless Sensor Network
PEWSN	-	Power Equilibrium Wireless Sensor Network
FSM	-	Finite State Machine
TWSN	-	Terrestrial Wireless Sensor Network
UWSN	-	Underwater Wireless Sensor Network
TWPSG	-	Truly Wireless Polysomnogram
PSG	-	Polysomnogram
COTS	-	Commercial Off-The-Shelf
QoS	-	Quality of Service
MEMS	-	Micro-electro-mechanical Systems
MTE	-	Minimum Transmission Energy
TDMA	-	Time Division Multiple Access
CDMA	-	Code Division Multiple Access
RTS	-	Request To Send
CTS	-	Clear To Send
NFL	-	Neighborhood Feedback Loop
DoD	-	Department of Defense

CHAPTER 1. INTRODUCTION

1.1 Thesis Motivation

A wireless sensor network (WSN) consists of spatially distributed autonomous sensors that cooperatively monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion, or pollutants, at different locations. Fig. 1 shows examples of wireless sensor nodes. Recent advances in low-power highly-integrated electronics, advances in micro-electro-

Fig. 1. Wireless sensor nodes

mechanical systems (MEMS), rapid growth in the type and quality of available sensors, and progress in communication have allowed WSNs to achieve an unprecedented growth in commercial, industrial and military applications. In order to better understand WSNs, we look at their major characteristics, applications, and performance parameters.

WSNs are characterized by the following:

- Limited Power
- Unreliable Communication
- Need for Self-Configuration
- Need for Scalability (order of 1000s)
- Mostly Immobile
- Harsh Environmental Conditions
- Small Size
- Cooperative network behavior
- Data-centric rather than address-centric (data expected to be aggregated, compressed, prioritized, dropped)
- Very short packets (overhead important)
- Many-to-one traffic common topology (hot-spot problem)
- Unattended operation
- Random deployment

Given these characteristics, the most common WSN applications are:

- Environmental monitoring
- Health monitoring
- Terror threat detection
- Terrestrial habitat monitoring
- Underwater tracking of fishes and micro-organisms
- Military surveillance

- Seismic oil and gas explorations
- Inventory tracking
- Process monitoring
- Acoustic detections
- Objects localization
- Homeland Security protection
- Industrial and commercial components aging detection and failure prevention
- Disaster prevention and disaster recovery
- Structures and pipelines corrosion detection

WSN performance is difficult to evaluate. Various industrial and commercial applications consider WSN scalability, ease of deployment and integration, as well as unit cost as the main performance discriminators. On the other hand, military and homeland security applications tend to be sensitive to durability, security, reliability and latency of their fielded WSNs.

The most common WSN performance parameters are:

- Network lifetime
- Power consumption and efficiency
- Network latency
- Sensor node size
- Network scalability
- Network and sensor node modularity

- Network security (eavesdropping, spoofing, message integrity, denial of service, geolocation, physical compromise)
- Network reliability
- Component durability
- General applicability
- Quality of Service (QoS)
- Fidelity
- Cost
- Ease of deployment
- Privacy

All applications are interested in long WSN lifetime. The challenge and the motivation of this thesis are to present a novel WSN architecture that will increase the network lifetime while ensuring a deterministic latency. By increasing the network lifetime and providing a means to calculate the worst case sensed data latency, this dissertation widens the number and type of applications that can benefit from using WSNs.

1.2 Thesis Contribution

This thesis presents the Power Wireless Sensor Network (PEWSN) architecture. PEWSN is a novel architecture specifically designed to significantly increase the WSN lifetime. The network lifetime, measured as the round in which the first node dies, is doubled when compared with the Low-Energy Adaptive

Clustering Hierarchy (LEACH [1]), increased by 23 times compared to static clustering protocols, increased by 230 times compared to minimum-transmission-energy (MTE) routing protocols, and increased by 17 times compared to direct routing protocols. Moreover, measured as the round in which the last node dies, PEWSN more than doubles the lifetime of a LEACH-based network, increases the lifetime by 25 times compared to static clustering protocols, increases the lifetime by 6 times compared to minimum-transmission-energy (MTE), and 12 times compared to direct routing protocols. The comparison results summary can be found in Table 3 (Section 4.1).

The increased network lifetime is the most important performance parameter for applications with sensors placed in difficult-to-reach places such as underwater, underground, military hostile territories, and other environments where it is difficult to replace individual sensors. The increased network lifetime also helps the applications with a vast number of nodes where the replacement procedure might be an overwhelming task in terms of time and cost. Finally, long-lived WSNs based on PEWSN decrease associated maintenance and replacement costs, making them more attractive for a wide range of applications that are currently not based on WSN technology.

PEWSN is an architecture that ensures a deterministic, bounded latency. In other words, no matter how large the network is or whether the data are coming from sensors close or far from the final data gathering point (base), the latency of the data propagating packets can be determined a priori. The deterministic latency is required for a number of control and cyber physical system applications that are

based on timely interactions between sensors, actuators, processing units, and various robotic components. Furthermore, PEWSN supports a seamless acquisition of nodes geolocation as well as embedded network security features such as symmetric cryptography.

This research develops a MATLAB-based PEWSN protocol simulation. Using the simulation, we can develop and optimize any PEWSN-based network. The simulation also allows for changing a number of PEWSN performance-critical parameters such as the number of clusters and sensors, the cluster head switching threshold, the threshold back-off factor, and the network field size. The MATLAB-based PEWSN simulation represents a behavioral model of the network.

We also develop an PEWSN OMNET++-based discrete event network simulator that includes all PEWSN features. The message traffic among individual nodes and clusters is supported. The simulator also has a configurable number of sensors and clusters. In order to better simulate the real environment, the network can consider real commercially available hardware parts and their parameters. The PEWSN simulator uses the OMNET++ framework and is capable of detailed monitoring and parallel processing of any or all nodes within the defined network. The simulator can also define and execute various trigger events. The entire simulation can be graphically displayed and dynamically updated. The simulation results can be stored as scalars and vectors in a format convenient for further processing with third-party tools such as MATLAB and MS Excel.

1.3 Thesis Organization

This thesis is organized as follows.

Chapter 2 classifies existing WSN architectures into specific groups based on WSN behavior and data flow characteristics. Existing architectures are described and presented along with their advantages and disadvantages. The existing architectures are evaluated in terms of their lifetime and latency, which are highly relevant to PEWSN performance.

Chapter 3 gives a full overview of the PEWSN architecture, including PEWSN network building blocks, TDMA schedule, network phases and messages. Critical, performance-differentiating features are elaborated in detail. The chapter concludes with an example of PEWSN.

Chapter 4 evaluates PEWSN performance. It starts by describing the performance simulation environment and then compares it with direct routing, minimum transmission energy (MTE) routing, static cluster routing, and LEACH. The chapter concludes with a presentation and an analysis of obtained results.

Chapter 5 describes the custom-designed PEWSN network simulator. The simulator's design block diagram along with main structural blocks is presented. The main features of the simulator are presented and described in detail.

Chapter 6 presents four distinct real-life applications where PEWSN architecture was applied. We describe the applications and evaluate their performance in the context of their lifetime and latency.

Chapter 7 concludes the work.

CHAPTER 2. PRIOR ART REVIEW

Advancements in wireless communications, electronics, and battery technology have enabled the development of low-cost wireless sensor networks. Propelled by more efficient electronics, battery, and energy harvesters, a number of new WSN architectures have been developed. Akyildiz et al. [2] provide a detailed overview of various constraints that drive new WSN designs and present the WSN protocol stack, consisting of application layer, transport layer, network layer, data-link layer, and physical layer. While all of the WSN layers developed over time, considerable research attention has been given to network layer of the stack. In fact, most architecture acronyms in the literature are associated with the design and development of the WSN network layer.

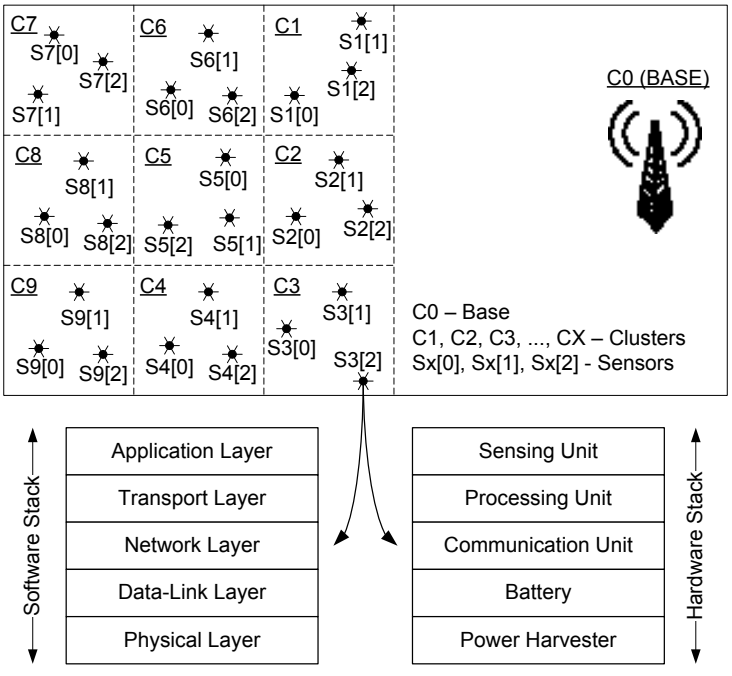


Fig. 2. WSN Stack

In this chapter, we describe various network layer architectures along with their advantages and disadvantages. Akkaya and Younis [3] surveyed the field of WSN architectures and grouped them into data-centric, hierarchical, location-based, and network and QoS flow. Yang and Mohammed [4] define the same architectural groups as Akkaya and Younis but add additional architectures to each group. Singh et al. [5] add three architectural groups: mobility-based architectures, multi-path-based architectures, and heterogeneity-based architectures. Finally, Yick et al. [6] add geographical routing and anchor location service (ALS) to location-based architectures and security routing (SecRout) and secure cell relay (SCR) to the hierarchical group.

We group all architectures/protocols as follows [7]:

- Data-centric architectures
- Hierarchical architectures
- Location-based architectures
- Mobility-based architectures
- Quality of Service (QoS) architectures
- Other Architectures
    - Network flow architectures
    - Multipath-based architectures
    - Heterogeneity-based architectures

Table 1 groups all major architectures.

Group	Architectures/Protocols	
Data-centric	1. Flooding 2. Gossiping 3. SPIN 4. Directed Diffusion 5. Rumor Routing 6. Energy-aware routing for low-energy ad-hoc WSN	7. Gradient-based 8. COUGAR 9. ACQUIRE 10. Information dissemination by negotiation 11. EAD 12. Information-directed
Hierarchical	1. LEACH 2. PACT 3. HEED 4. PEGASIS 5. Hierarchical-PEGASIS 6. TEEN	7. APTEEN 8. Energy-Aware Routing for Cluster-based WSN 9. SecRout 10. SCR
Location-based	1. GAF 2. GEAR 3. SPAN 4. TBF 5. GeRaF	6. ALS 7. BVGF 8. MECN 9. SMECN 10. Geographic Routing in Lossy WSNs
Mobility-based	1. SEAD 2. TTDD 3. Joint Mobility and Routing	4. Data Mules 5. Dynamic Proxy Tree-based dissemination
QoS	1. SAR 2. SPEED 3. N-to-1 Multipath Disc.	4. RL-MAC 5. MMSPEED 6. DARP
Network flow	1. Max Lifetime Energy 2. Max Lifetime Data Gathering and Aggreg.	3. Min Cost Forwarding
Multipath-based	1. Node-disjoint 2. Braided Path 3. N-to-1 Multipath Discovery	4. SEEM 5. REER 6. HMRP
Heterogeneity-based	1. CADR 2. IDSQ 3. CHR	4. HDMRP 5. SEP 6. EEHC

Table 1. WSN Architectures Grouping

2.1 Data-Centric Architectures

Data-centric architectures are characterized by a vast number of randomly deployed sensors that only communicate node-to-node without any global network identification. In these types of architectures, a sink node sends a request query through the network of nodes, and the source node responds to the query; alternatively, the source node sends an event query, and the sink node routes to the event. The goal in these architectures is to send the data through the most efficient route between the sink node and the source node. Data-centric networks tend to be power inefficient, because the entire network is involved in data transfer. Many data-centric protocols try to improve the power efficiency by creating dedicated source-to-sink routes so that the rest of the network can save power. Data aggregation is another popular approach, because it reduces the number of packets traversing through the network.

Flooding and gossiping [8] are the most prominent representatives of data-centric architectures. In *flooding*, each sensor node sends data to all of its neighbors. The send packet is propagated throughout the entire network until it either reaches the destination or the number of maximum hops is reached. Flooding was one of the first data-centric protocols. The main flooding disadvantages are implosion and overlap. Fig. 3 explains both problems using four nodes (A, B, C, and D). Implosion is where a packet originating from the same source (Node A) travels through different paths (Nodes B and C) but arrives as a duplicate at some other node (Node D) within the network. This creates power inefficiencies within the network. A similar case is the overlap problem,

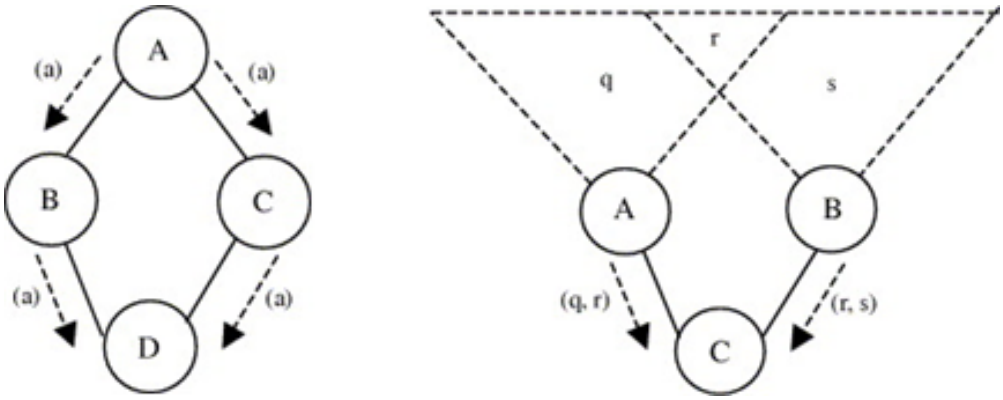


Fig. 3. Flooding implosion and overlap problem.

where data indeed originates from two different sources (Nodes A and B), but both sources cover the same overlapping area r, resulting in a data duplicate at the node neighboring both source nodes (Node C). Another disadvantage of the flooding protocol is the fact that all nodes must be on all the time to avoid missing packets. This is very power inefficient, leading to extremely short network lifetime.

The *gossiping* protocol is a more efficient version of flooding, because it uses a single, randomly selected neighbor to transfer each packet. Therefore, gossiping avoids the implosion problem by creating a single random path from the source to the sink. The overlap problem still exists in gossiping. Another disadvantage is that gossiping creates long propagation delays, because the selected random path might be suboptimal in terms of propagation latency. In fact, the propagation delay is not bounded within some limits, because the selected data path is random. On the other hand, flooding and gossiping protocol offer simple implementation, and there is no need for network level synchronization among nodes.

Sensor Protocol for Information via Negotiation (SPIN) [9] starts with a source (Node A) advertising the availability of its data to neighboring nodes (Fig. 4). The ADV message contains meta-data necessary for neighboring nodes to decide if they would like to acquire the data. The nodes interested in data submit their requests for data (REQ message); in return, they receive the data. The negotiations between nodes continue until the data reach its final destination—the sink node. The SPIN advantage is seen in the relative localization of topological changes. In other words, changes in the location of nodes only affect local negotiation but also the overall source to sink delivery. However, SPIN suffers

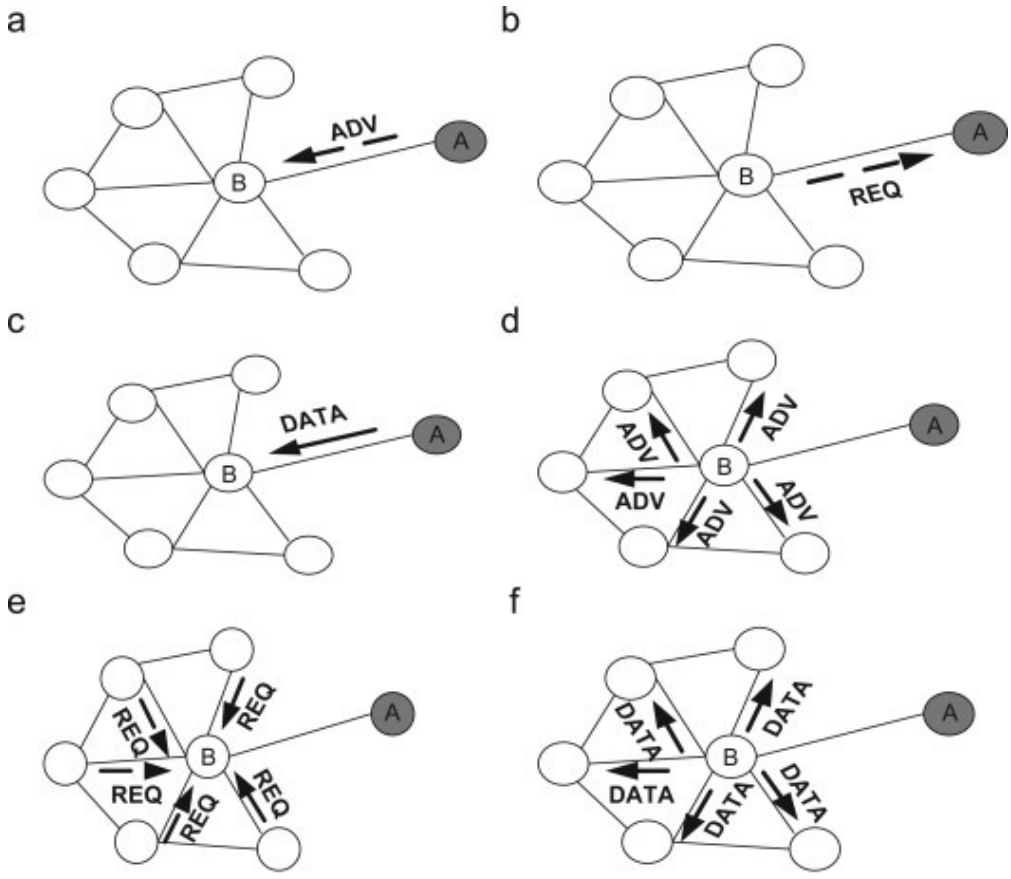


Fig. 4. SPIN Protocol

from lack of quality of service (QoS) and cannot ensure that negotiation among nodes along the source-sink path will guarantee the final delivery of data (i.e., intermediate nodes might decide not to request data upon receiving the data advertising message).

The *directed diffusion* architecture [10] starts with sink advertising or requesting data and nodes responding to the request (Fig. 5). The transfer starts with a source flooding the network with messages containing attribute-value request pairs. For example, a pair might be (location, 50^{o}C) signifying the request for location of the place where the measured temperature exceeds 50^{o}C. Once the flooded message arrives at the source node, the node issues the acknowledgment of data existence. While the acknowledgment message travels back to sink, the hop gradient, defined as the measure of hop distance between two nodes, is

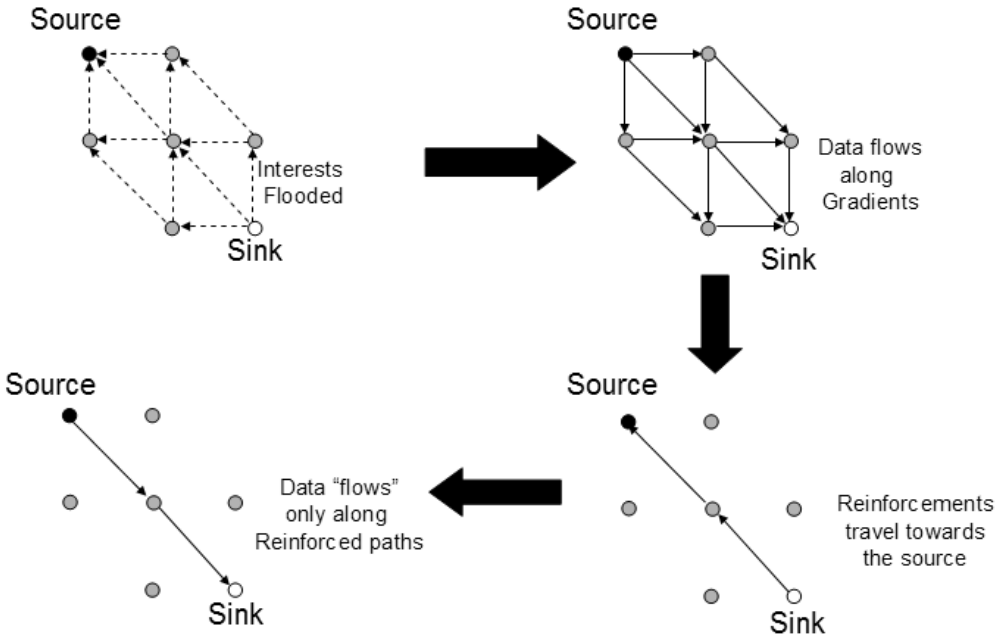


Fig. 5. Directed Diffusion

recorded and propagated along with the acknowledgment. Next, the sink nodes send the request for data through the most optimal path, effectively reinforcing the path through which it would like data to go. Finally, data are sent through the designated path. A major advantage of directed diffusion is the node-to-node communication without the need for a global network addressing mechanism. In addition, nodes can cache and aggregate data, which in turn saves overall network power consumption, and the data traverse the network only when requested, avoiding unnecessary power consumption when the data are not needed. On the other hand, directed diffusion is not suitable for applications that require immediate reporting to a trigger event (such as military and homeland security applications). It also has poor QoS, because the latency between source and sink can vary greatly.

In *rumor routing* [11], flooding the network with queries or events is prevented through the use of packets called agents (Fig. 6). In the case of an event, the event triggering node generates a packet called an event agent. The agent is then sent through several random paths advertising the event existence and the source of the event. All nodes that are not already familiar with the particular node log the agent into their routing tables. Eventually, the particular event's agent builds one or more event agent paths. The node that is interested in the event sends its own agent, the query agent, through multiple nodes until it reaches the node that knows how to route the particular query to the source node. Therefore, the query agent path and event agent path cross, and the path from the source to the sink is established. Rumor routing is energy efficient since it

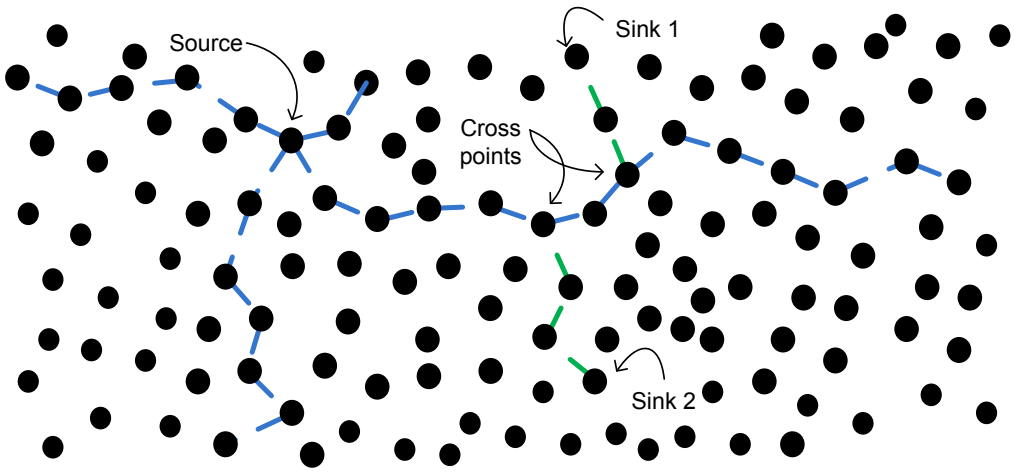


Fig. 6. Rumor Routing

prevents sources and sinks from flooding the network with events and queries. On the other hand, it establishes only a single path between the source and the sink, creating a network reliability problem. In addition, rumor routing does not guarantee the discovery of a successful route between the source and the sink nodes, because it cannot ensure that the query agent's and the event agent's paths will cross in the discovery phase.

Gradient-Based Routing (GBR) [12] defines the distance between the event source and individual nodes in hop counts. The minimum number of hops from the source to the user is called the node height. The difference between a node's height and that of its neighbor is called the link gradient. A packet is always sent via path with the largest gradient. For example, if the packet originating Node A has a hop count of 15 and its neighbors Node B and Node C have hop counts of 14 and 7, respectively, then grad(Node A, Node B) is 1, and grad(Node A, Node C) is 8. The higher gradient is grad(NodeA, NodeC); therefore, the packet will be sent from Node A to Node C. Since Node C's height

is 7, the arriving packet's maximum number of additional hops will be 7 as opposed to 14 if the packet were sent via Node B. GBR is an improved version of the directed diffusion protocol, with an increased network lifetime of up to 90%. Like many other data-centric architectures, GBR also takes advantage of data compression and data fusion.

Energy Aware Routing for Low Energy Ad Hoc Sensor Networks [13] extends network lifetime by occasionally using power sub-optimal routes. Shah et al. argue that permanently using the most optimal minimum energy path will actually decrease the network lifetime, because the same nodes will experience a disproportionally high traffic compared to other less utilized nodes.

COUGAR architecture [14] is a software approach to solving the WSN power efficiency issue. In the COUGAR, the number of power costly data transmissions from individual nodes to the base (central data gathering place) is replaced with cheap local computation. In other words, the so-called declarative query generates an efficient optimized query plan that interrogates only necessary nodes and reduces their data load by using local processing. The result is minimum data communication with the base. However, the power savings from the reduced communication among nodes come at the expense of the more sophisticated nodes' communication stack. Recently, COUGAR introduced a new communication layer called query layer. In addition, leader nodes that generate declarative queries are disproportionally utilized, creating a power imbalance among nodes.

Another interesting data-centric architecture is *ACQUIRE - ACtive QUery forwarding in sensoR nEtworks* [15]. In ACQUIRE, the sink node sends a data request query that propagates throughout the network either randomly or through some directed means. Each time that an active query reaches an intermediate node, the node will try to resolve the query using either event cached information or information from its neighbors *d* hops away from the node. Once the query is fully resolved, the response to sink is sent via the most optimal intermediate nodes, and the sink-source path is established. Therefore, ACQUIRE relies on selecting the most power efficient path by each node on the path looking *d* hops ahead for the most optimal solution. For *d=1* case, ACQUIRE behaves as flooding architecture.

Information dissemination by negotiation [16] is an energy efficient data-centric approach that has three major advantages: a fully distributed network, high success rate for data retrieval, and capability to deal with mobile sensors in addition to static sensors. Energy efficiency is measured by the number of message transmissions required for the source node to advertise its data to all possible data consumers and by the number of hops of the path between the source node and the querying node for data transmission.

Additional data-centric power efficient architectures are Energy-Aware Data-Centric Routing (EAD) [17] and information-directed routing [18].

2.2 Hierarchical Architectures

Data-centric network topologies are not suitable for large-scale sensor networks. Covering a large area without performance degradation is not possible with data-centric architecture. Moreover, in data-centric architectures, the reporting latency increases with the size of the network. The data-centric approach also causes significant power inefficiencies as the network grows.

The network scalability issue is addressed in hierarchical routing. The hierarchical routing's main goal is to efficiently maintain network power consumption even in large-scale networks. In other words, hierarchical routing allows the network to scale in a number of sensor nodes. Most hierarchical architectures consist of sensor nodes grouped into cluster heads. Cluster heads build intra-cluster communication with other nodes within the same cluster, but they also build inter-cluster communicate with other cluster heads. Cluster heads aggregate data obtained individual sensors and then transfer the same information mostly in a multi-hop approach to the base.

Low-Energy Adaptive Clustering Hierarchy (LEACH) [1] is one of the most popular hierarchical architectures. LEACH utilizes a randomized rotation of local cluster heads to evenly distribute the energy load among sensors in the network. It also minimizes the overall energy consumption by allowing each sensor node to determine which cluster it wants to join by choosing the cluster head that requires the minimum communication energy (typically, the cluster head closest to the sensor). However, LEACH architecture determines the percentage of cluster heads in the network and cluster switching frequency a priori. This approach may

lead to a less than optimal number of cluster heads in the network at any point of time. It also leads to unnecessary overuse of cluster head switching and a waste of network power capacity associated with the switching overhead.

Power Aware Clustered TDMA (PACT) [19] uses a more efficient cluster head switching algorithm. The PACT architecture takes individual nodes (sensors) energy levels into account when selecting cluster heads. PACT also uses passive clustering [20] that limits the number of exchange control messages and therefore reduces the power overhead associated with cluster head switching. However, PACT cluster switching is still probabilistic and does not always lead to optimal network lifetime. Additionally, neither PACT nor LEACH uses ambient power harvesting methods to extend the network lifetime.

The *Hybrid Energy-Efficient Distributed (HEED)* hierarchical architecture [21] extends the network lifetime by taking into account the residual energy of each node (primary parameter) and considers intra-cluster communication cost (secondary parameter). The second parameter is based on AMRP (average minimum reachability power) and is a good measure of communication energy consumption if the node becomes a cluster head. Like LEACH, HEED selects a percentage of cluster heads a priori that does not always lead to an optimal number of cluster heads. Furthermore, HEED cluster head selection is probabilistic. The selection heavily relies on an a priori selected percentage of cluster heads allowed in the network and the a priori selected minimum ratio between the sensor residual and maximum energy.

Power-Efficient Gathering in Sensor Information Systems (PEGASIS) [22] builds sensor node chains rather than clusters. Fig. 7 explains the concept; each node (s1 through s4) sends its data to only one neighboring sensor. As sensors

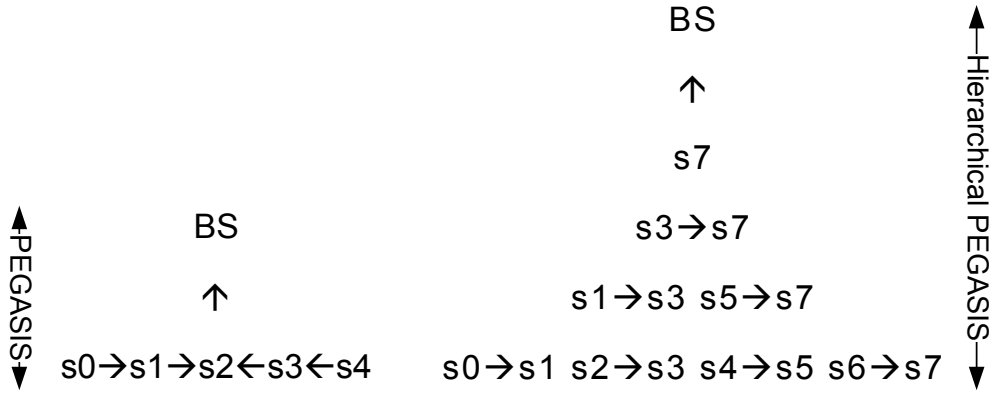


Fig. 7. PEGASIS and Hierarchical PEGASIS routing

send data to only one of their neighbors, they build a chain of sensors, and only one node transmits data to the base or to another cluster. PEGASIS is very efficient, because it allows each node to have its transmitter turned on for only one TDMA slot and its receiver also for only one TDMA slot. In all other TDMA slots, the node can sleep and therefore conserve energy. PEGASIS is also power efficient, because the cluster setup phase is minimal. On the other hand, the biggest disadvantage is the excessive delay for distant nodes.

Hierarchical PEGASIS [23] improves traditional PEGASIS by decreasing the propagation delay (Fig. 8). In hierarchical PEGASIS, each node sends data to only one neighbor; instead of building a chain of nodes, it builds a binary scheme of nodes. The binary scheme of nodes precludes the protocol with embedded CDMA coding, because multiple transmissions occur in the same TDMA slot.

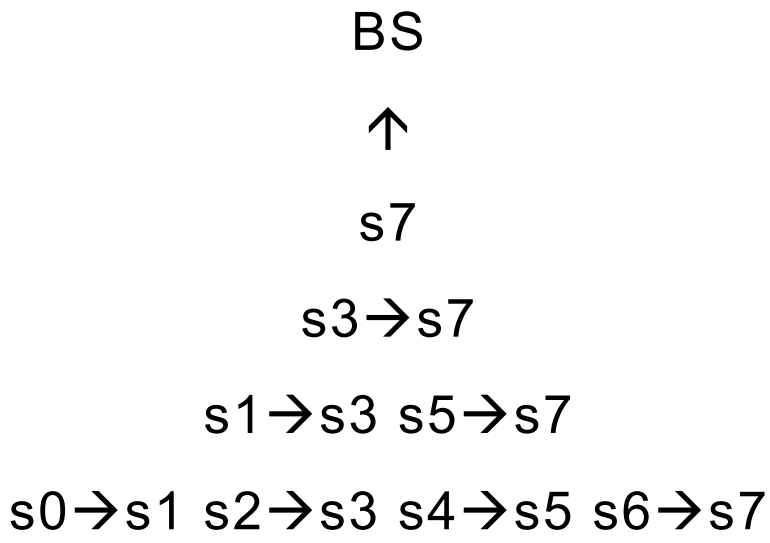


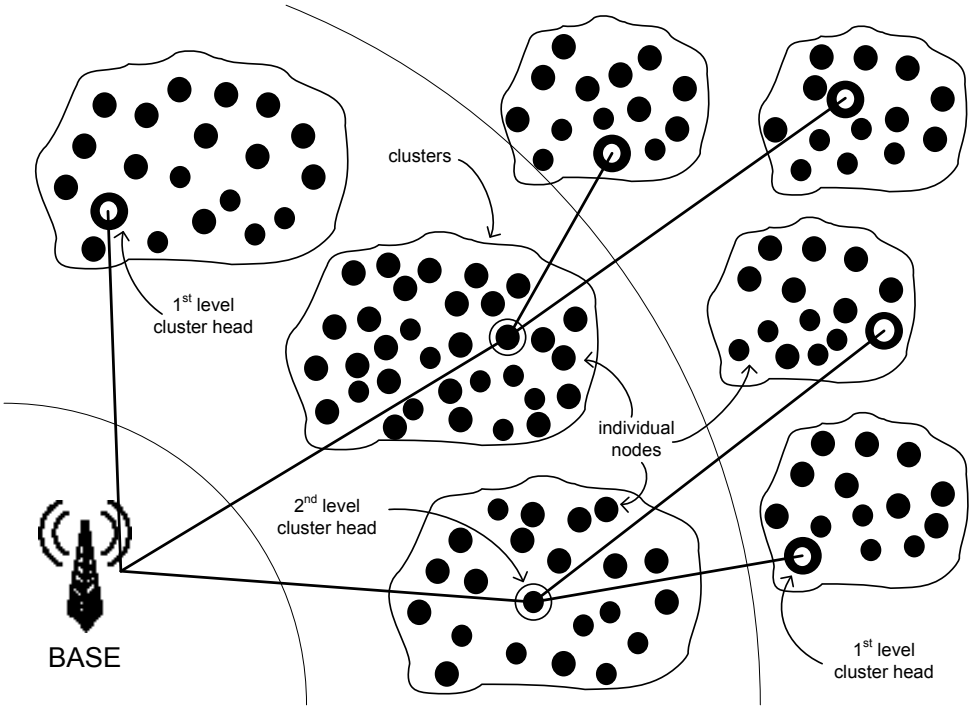
Fig. 8. Hierarchical-PEGASIS routing

The use of CDMA code might significantly decrease the channel bandwidth and therefore cause network power inefficiencies. In addition, the binary scheme requires more carefully synchronized nodes.

The *threshold sensitive energy efficient sensor network (TEEN)* [24] architecture relies on building two levels of cluster head nodes (Fig. 9). Individual nodes are grouped into clusters, and clusters are represented via cluster heads. Cluster heads aggregate the data of all individual nodes and transfer them further toward the base. If a cluster head has an uplink path to at least one cluster head farther away from the base, then it is a second-level cluster head. Unlike first-level cluster heads, which only transfer cluster data down the link to the base, second-level cluster heads aggregate and transfer all data originating from directly linked cluster heads.

The feature that differentiates TEEN from any other hierarchical architecture is the use of hard and soft thresholds. The hard threshold defines when an individual node is allowed to send its data to the cluster head. In other words, if a particular node's attribute (e.g. temperature) reaches the threshold value, the node reports the event to cluster head once. The software threshold defines the incremental delta value that needs to be reached in order for node to report the event again. For example, if the hard threshold is 70^{o}C and the soft threshold is 5^{o}C, the node will report the temperature rise at 70^{o}C, 75^{o}C, 80^{o}C and so on. Hard and soft thresholds help TEEN limit the number of intra-cluster messages by filtering small changes in measure attributes. TEEN does not perform well in applications where periodic reports are needed, since the user may not receive any data if the thresholds are not reached. If the collected data does not exceed hard threshold, the node does not transmit any sensed data. And if it does not exceed soft threshold, we cannot know about data changes after the default value is passed, especially if the data change is under the threshold value. Moreover, due to those thresholds it is hard to judge whether the nodes are alive or not.

The *Adaptive Threshold Sensitive Energy Efficient Sensor Network (APTEEN)* [25] architecture is an extension of the TEEN architecture. APTEEN addresses TEEN's shortcomings by capturing periodic data collections and reacting to time-critical events. In addition to all of TEEN's features, APTEEN also supports three different query types: historical, to analyze past data values; persistent, to deliver data on a regular basis; and one-time, to take a network



snapshot. While APTEEN extends the number of applications in which it can be used, it is still considered one of the setup overhead heaviest architectures.

Energy-aware Routing for Cluster-based Sensor Networks [26] is an interesting novel architecture since it takes into account the latest hardware features to save transmission power. This architecture relies on individual nodes being capable of adjusting their transmission power to account for the distance range between nodes. The main architectural structures are gateways (cluster heads) that program individual nodes with the exact TDMA schedule as well as precise functionality (sensing nodes, sensing-relaying nodes, relaying nodes and inactive nodes). An extended version of the same architecture is proposed in [27] where the algorithm constrains the minimum transmission range in order to limit the delay.

Security Routing (SecRout) [28] is cluster-based approach that emphasizes the secure delivery of packets from the source to the sink. This architecture employs standard hierarchical routing network elements such as individual nodes, clusters, cluster heads and sink. Packet transfers among sensor nodes are secured via symmetric cryptography. Each sensor is given a unique ID and a unique pre-loaded key. All individual sensors with a cluster use the cluster head's KEY to encrypt data. The cluster head decrypts and aggregates all data from all sensors within the same cluster. Then, it encrypts the aggregated packet and sends it back to the sink node. The sink node (base) is assumed to be trusted and power-rich. The sink node also contains all ID/KEY pairs from all sensor nodes and therefore can easily detect an attack.

Secure Cell Relay (SCR) [29] is an even more secure architecture providing security against the following attacks: Sybil, wormhole, sinkhole, selective forwarding, and hello flood. In SCR, the sink node distributes a global key that is used for initial neighborhood discovery and handshake communication. In the discovery phase, nodes use a three-way handshake protocol to establish a shared secret key between neighboring nodes. Therefore, each pair of neighboring nodes shares one unique secret key. Once the discovery phase is complete, the global shared key is no longer needed, so it is destroyed. Another SCR feature is that routing paths from the source to the sink are formed through a series of cells and cannot be altered via wormhole or sinkhole attacks.

Additional architectures such as [30], [31] and [32] provide low-energy wireless sensor network communication and routing.

2.3 Location-based Architectures

Location-based architectures along with their underlining routing algorithms rely on knowledge of nodes' positions to route packets. Nodes might obtain their positions using low-power, embedded GPS receivers, through triangulation techniques, or simply by being placed at the known location. In this section, we concentrate on location-based architectures with the primary goal of energy efficient data routing.

Geographic Adaptive Fidelity (GAF) [33] is an energy-aware location-based architecture that conserves energy by identifying routing equivalent nodes and then turning off the unnecessary nodes, keeping a constant level of routing fidelity Fig. 10 depicts an example. Here, assume that each node from region 1 can talk to

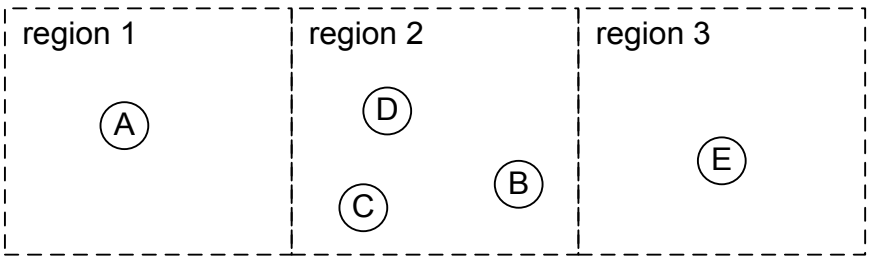


Fig. 10. GAF architecture

each node from region 2. Likewise, each node from region 2 can talk to each node in region 3. None of the nodes from region 1 can talk to any node from region 3. Since Nodes B, C, and D effectively cover the same region 2, two of them can go to sleep without affecting the overall routing scheme. Therefore, GAF saves energy by turning off two of the three nodes in region 2. Once the predetermined active time expires, a new node wakes up and takes over the responsibilities of the currently active node, which then goes to sleep. By activating and deactivating

different nodes that cover the same region, GAF significantly extends the network lifetime. GAF can be implemented for non-mobility and mobility of nodes. The disadvantage is that the leading nodes do not aggregate, filter, or compress data.

Geographic and Energy Aware Routing (GEAR) [34] is an energy-aware architecture that uses nodes' geographical information to route packets to the target area. In a sense, it is similar to directed diffusion, except the interest is sent to the specific target region. Fig. 11 depicts the architecture. There are two phases in the algorithm. In the first phase, each packet is routed through the set of nodes that uses the nearest-neighbor-to-target-region approach to select the next hop. In

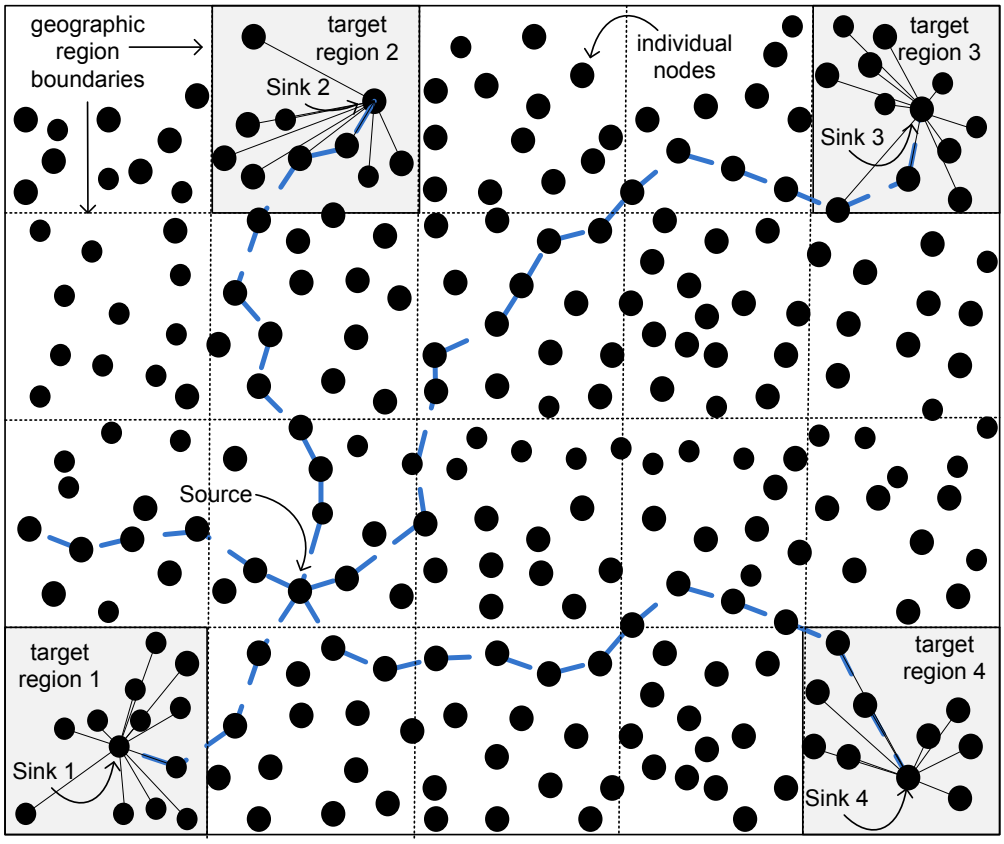


Fig. 11. GEAR architecture

the second phase, the packet that arrives in the target region is diffused using either recursive geographic forwarding or restricted flooding. Unlike other architectures, GEAR ensures that all nodes within the target region receive the packet. Like all other location-based architectures, GEAR requires the knowledge of nodes' positions to properly and energy-efficiently route a packet from the source to the target region (sink).

SPAN architecture [35] [36] uses its routing algorithm to efficiently select a set of backbone nodes (coordinators) whose goal is to efficiently transfer packets between the sink node and the source node. All other nodes not currently used as coordinators can retain inactive (sleep) status and therefore save energy. SPAN aims to satisfy the following requirements:

- As many nodes as possible should be able to sleep most of the time, with forward packets experiencing minimally more delay than if all nodes are awake.
- The backbone capacity (throughput) should be as high as the capacity of the original network.
- There shall be seamless coordination, interoperability, and sleep mode support by all link-layer and physical layer protocols.

SPAN performance increases with increased node density. A disadvantage is the relatively high overhead associated with coordinator selection and backbone routing algorithms. This is magnified in larger and denser WSNs.

Trajectory-Based Forwarding (TBF) [37] uses node geolocation to forward packets via a predetermined, source-specified trajectory. The advantage of the approach is that the trajectory specifies the general direction or nature of the path, but it does not specify the exact nodes that need to participate in the packet forwarding. This feature allows the network to route or even reroute packets using the best available resources that are close to the projected path. Fig. 12 depicts various trajectories descried in [37]. The TBF protocol is very flexible and can implement path redundancy by simply sending the same packet through two or more separate trajectories.

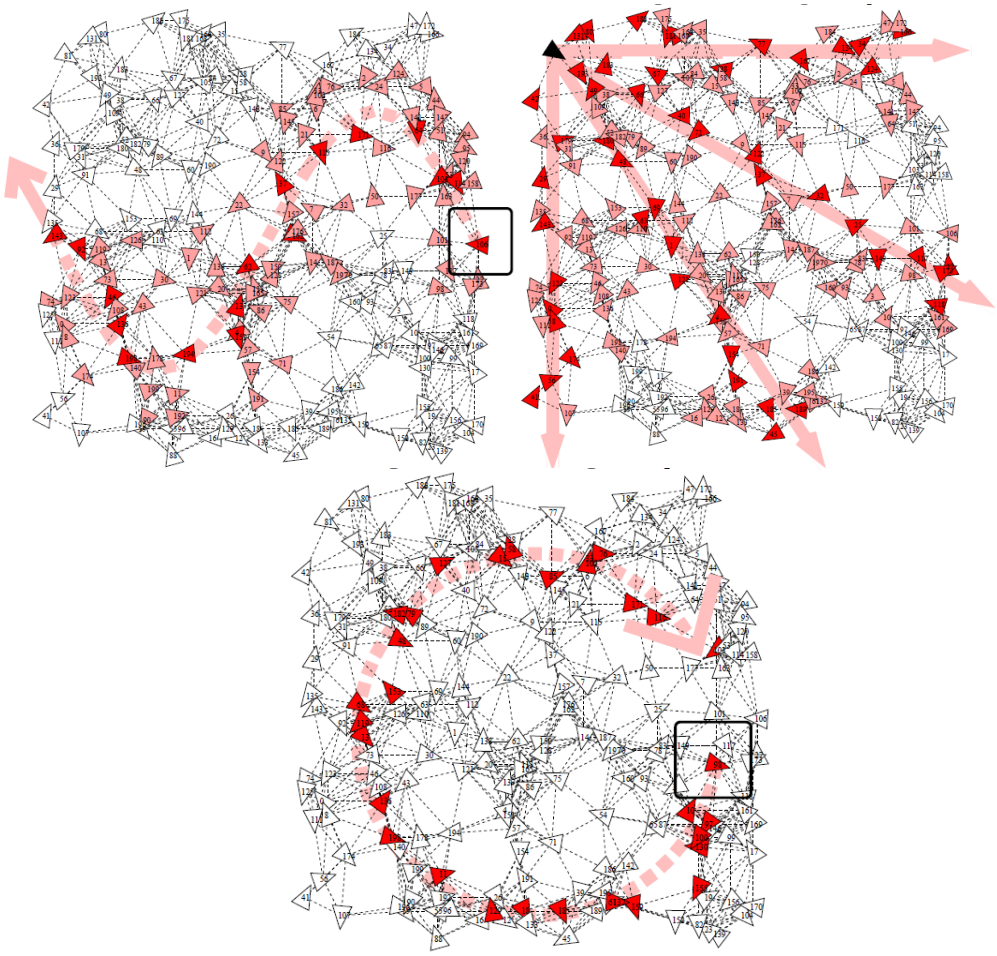


Fig. 12. TBF architecture

Geographic Random Forwarding (GeRaf) [38] is a novel greedy-based forwarding architecture that uses a randomly selected relay node from the region closest to the sink to route its data packets. The process starts with the source node sending a request-for-send (RTS) message to all nodes within a priority region. The priority region is known, because the source node uses location-based architecture that requires the location knowledge of neighboring nodes. If there is no clear-to-send (CTS) reply from the priority region, the source node sends an RTS to the region with the second most priority. If there is still no answer, the source node continues traversing the priority region until the CTS message is received, or the source node simply gives up and declares the packet as undeliverable (best-effort forwarding). However, if a CTS is received, the source node simply forwards the data packet to the CTS message originating node. The transaction is complete once the source node receives the data acknowledgment packet. If multiple relay nodes send CTS messages, a contention resolution algorithm is used to resolve the contention and to allow a single randomly chosen node to send a CTS back to the source node. The same concept is applied to further advance the packet through relay nodes to the final destination (i.e. sink node). GeRaf is considered a greedy, best-effort forwarding architecture that does not guarantee packet delivery. This is a substantial disadvantage, especially when used in applications requiring a certain level of QoS. On the other hand, GeRaf is adaptable to network topology changes created by nodes changing their status from active to sleep and vice versa.

Anchor Location Service (ALS) [39] is a power-efficient, location-based architecture that supports routing among multiple moving sources and destinations. Fig. 13 depicts the basic ALS routing mechanism. ALS constructs a

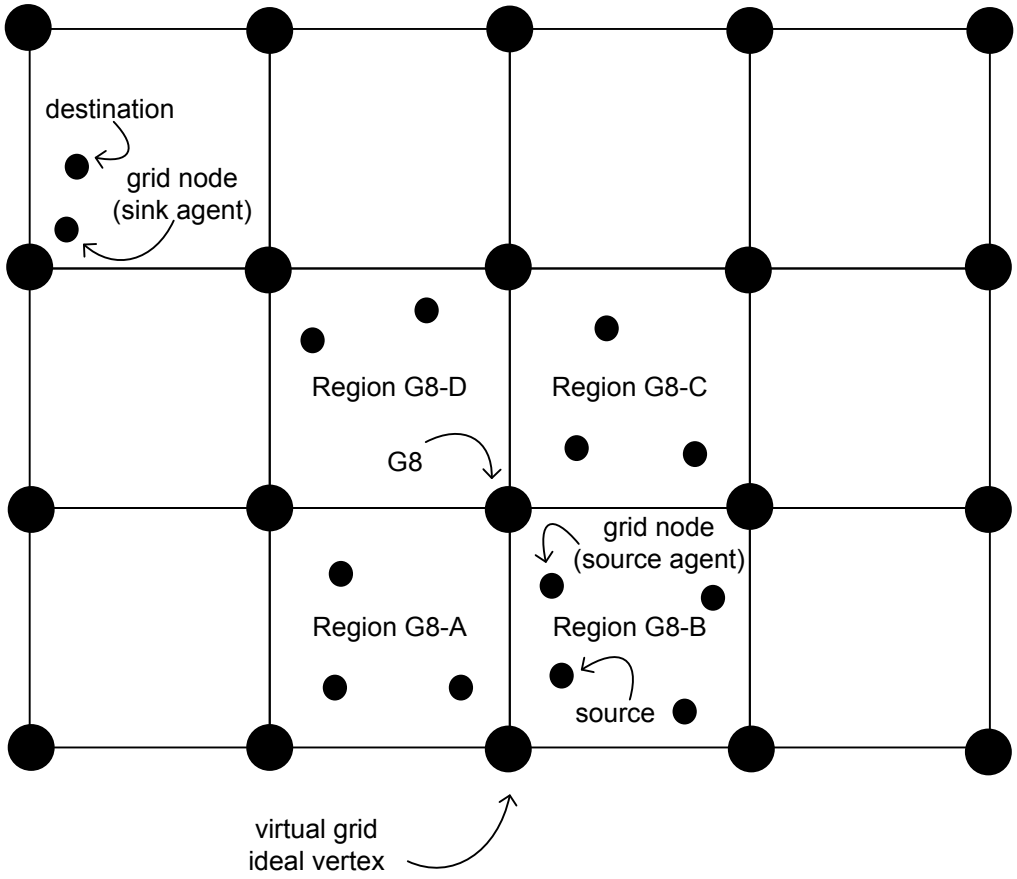


Fig. 13. ALS geographical grid

virtual Cartesian grid with a scalable number of ideal vertexes. Ideal vertexes are not real nodes but rather virtual positions of squares' vertexes that are used as the reference points for further location-based ALS routing steps. Each square is α*α. Therefore, all nodes belong to one of the squares that is (a*α, b*α) away from some reference geographic point in the coordinate system. In the initialization phase, the sensor node closest to the virtual grid ideal vertex is selected as the so-

called grid node for that particular square. In Fig. 13, virtual grid vertex G8 is used as the location reference for selecting Region-G8B's grid node. Therefore, each square in the coordinate system has one designated grid node. The same grid node also becomes a source or a sink agent node if the source or the sink node is found in the particular square.

The sink agent is responsible for distributing the information about the sink location via the anchor system of grid nodes. Each sink node builds its own anchor system. Having separate anchor systems, sink nodes are allowed to move within the network without losing the node's synchronization with the overall network's node structure. The source agent is responsible to find and attach to the sink's anchor system. Once the source agent discovers the sink's anchor system, ALS uses the location-based routing algorithm to find the most power-efficient path from the source to the sink. Implementation of ALS tends to be straightforward and less cumbersome than the average location-based routing scheme. Its advantage is the fact that it supports multiple moving sources and sinks with modest storage and communication power requirements. It is also scalable in terms of covered geographic space as well as the network density.

Other power efficient location-based architectures are Bounded Voronoi Greedy Forwarding (BVGF) [40], Minimum Energy Communication Network (MECN) [41], Small Minimum Energy Communication Network (SMECN) [42], and Geographic Routing in Lossy WSNs [43].

2.4 Mobility-based Architectures

Mobility-based architectures assume that a source, a sink, or intermediate nodes change their positions over time. Some architectures also assume that there are multiple sources and multiple sinks in the WSN field. Routing through a constantly moving set of nodes is a difficult problem that requires a lot of energy to keep the network well connected. Consequently, architectures presented below limit the problem to mobile sources and sinks moving within the stationary network of intermediate nodes. This approach is reasonable, because most applications have a stationary network structure in the field and expect only a source node or a sink node to move.

Scalable Energy-efficient Asynchronous Dissemination (SEAD) [44] supports moving data from a stationary source to a moving sink via a network of stationary nodes. The SEAD architecture starts with a source node that builds its own dissemination tree. In the case of multiple source nodes, there are multiple dissemination trees. The sink node is not a part of the tree; rather, it creates a relationship with the closest node belonging to the tree. The closest node to the sink becomes the sink's access node, which seeks to transfer source data via the dissemination tree. Once the data are available at the access node, the node simply transfers the data to the sink. SEAD is very flexible, because it allows the sink to move and change its access node. The access node changes once the distance threshold between the sink and its access node is reached. The value of this threshold allows trade-offs to be made between path delay and energy spent on

reconstructing the tree. SEAD also allows limited network traffic reduction by being able to send data to multiple sink nodes.

Two-Tier Data Dissemination (TTDD) [45] is similar to SEAD in that it also relies on stationary source and network nodes while allowing multiple sinks to move. Fig. 14 depicts TTDD data flow. It starts with each source building its

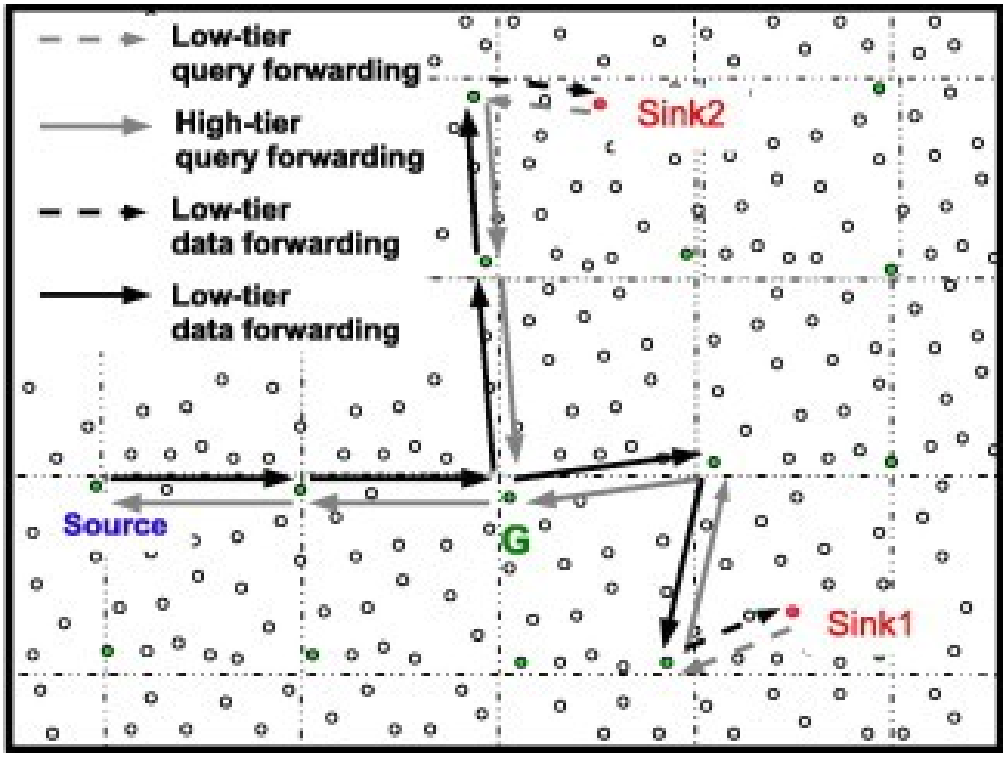


Fig. 14. TTDD architecture

grid structure (one tier). Once the grid structure is established, the data requesting sink node floods the local cell to find the cell's dissemination node (i.e. the node closest to the virtual cell vertex). Once it finds the dissemination node, the path from the sink to the source is established. Then, the sink requests the data, and the source responses with the data delivery. As Fig. 14 depicts, TTDD allows for limited data traffic reduction by requesting only once identical requests originating from different sink nodes (see Node G).

Joint Mobility and Routing [46] defines mobile sink nodes (called base stations) very differently than SEAD and TTDD. In this architecture, the nodes closest to the sink (base) deplete their energy fastest, because all other distant network nodes route their data through those nodes closest to the sink. In order to prolong the network lifetime, a mobile sink node constantly changes its position, which forces the network rotation of the closest nodes. Therefore, power consumption is evenly split among all network nodes. Sink mobility and the network routing protocol are developed jointly, so that the harmonious coexistence ensures even longer network lifetime. The main disadvantage is the fact that a mobile sink must be some kind of moving robot capable of traversing through long and sometimes unevenly distributed networks and terrains. This type of architecture might be suitable for underwater WSN applications.

Data MULEs [47] propose special, mobile nodes called MULEs (Mobile Ubiquitous LAN Extensions) that move through the network and pick up data from nodes found in close proximity. The close-range transfers can use promising communication technologies such as Ultra-Wideband (UWB) radios. By establishing data transfers from the source to the sink via mobile data MULEs, significant power savings can be achieved. On the other hand, there is substantial power loss due to the continuous listening needed to identify a passing MULE. Also, data latency is high due to the fact that sink nodes must first wait for source nodes to offload data and then wait for MULEs to deliver the data. However, a major advantage is the low cost of placing and maintaining the network.

Dynamic Proxy Tree-based Data Dissemination [48] is another mobile-based architecture that relies on the dynamic proxy tree-based framework. In this framework, each source is associated with a source proxy, and each sink node is associated with a sink proxy. Proxies related to the same source build a proxy tree, which is the facilitator of data movement. The source node disseminates data through its source proxy, which further propagates data to multiple sink proxies. The sink can then query its proxy to obtain data. The advantage of this architecture is the efficient reconfiguring of the proxy tree, as the proxies frequently change from one node to another.

2.5 Quality of Service (QoS) Architectures

QoS architectures are characterized by stringent requirements such as packet end-to-end network delay and packet end-to-end energy cost. QoS is usually needed for networks required to deliver real-time data or to deliver data with predefined reliability metrics. Generally, QoS architectures are complex with high network maintenance overhead, because WSNs are generally viewed as non-deterministic, randomly spread set of nodes with limited lifetime.

Sequential Assignment Routing (SAR) [49] creates multiple trees, each rooted from one-hop neighbor of the sink. The trees are created by taking into account the link cost between immediate neighbors. The tree creation algorithm avoids nodes with very low QoS and energy resources. All network nodes belong to multiple trees and can send data through multiple paths. Having multiple paths to the sink node, each sensor uses the SAR algorithm for path selection. The SAR

algorithm takes into account energy resources, the path's QoS metric, and the priority of the packet to select the optimal routing path to the sink. While the SAR architecture includes QoS in terms of latency, robustness, and reliability, the same architecture suffers from high overhead in maintenance of routing tables.

SPEED [50] is a truly unique architecture centered on real-time packet delivery. SPEED differentiates three types of services: unicast (point to point packet delivery), area-multicast (delivery to all nodes within an area), and area-anycast (one node representing the whole area of nodes). The routing itself is a combination of feedback control and non-deterministic geographic forwarding. In other words, SPEED is capable of routing packets through the most optimal path based on prior communication history through various paths. At the same time, SPEED is fully capable of managing the immediate network congestions through the backpressure rerouting scheme. Backpressure rerouting allows SPEED to change the direction of incoming packets if severe congestion is detected. Another QoS feature embedded into SPEED is the Neighborhood Feedback Loop (NFL), which is responsible for maintaining an a priori set single hop relay speed by effectively dropping all backlogged packets if the delivery speed drops below the set point value.

Energy-Aware QoS Routing [51] provides QoS aware routing for video and imaging transmission. This architecture finds the least-cost, delay-constrained path for real-time data in terms of link cost that captures nodes' energy reserve, transmission energy and error rate. A novel feature is the capability to prioritize real-time and non-real-time data at sensor nodes. However, the coexistence of

real-time and non-real-time data makes the routing problem extremely complex. In addition, this architecture also provides QoS meeting preset end-to-end delay requirements.

Additional QoS architectures are *Reinforcement Learning based MAC (RL-MAC)* [52], *Multipath Multi-SPEED (MMSPEED)* [53], and *Distributed Activation based on Predetermined Routes (DAPR)* [54].

2.6 Other Architectures

Network flow, multi-path, and heterogeneity-based architectures are described in this section.

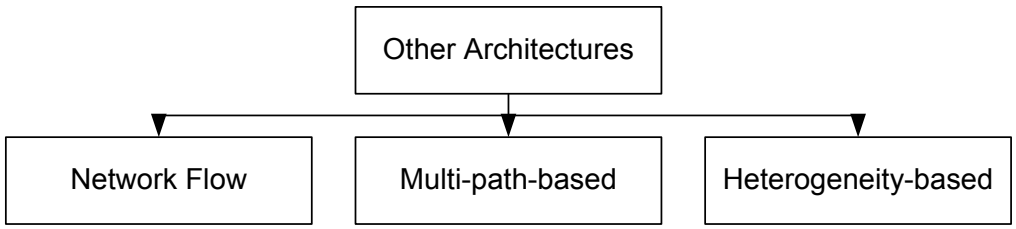


Fig. 15. Summary of other architectures

Network flow architectures are defined by the optimization process across node links. Links are defined as node-to-node communication with certain cost parameters such as the power to transfer a bit of data, the latency to transfer a bit of data, or the communication delay between two nodes. Therefore, the goal is to find the optimal routing path across all links between the source and the sink given the predefined link cost metric. The network flow architectures are: *Maximum Lifetime Routing* [55], Maximum *Lifetime Data Gathering and Aggregation* [56], and *Minimum Cost Forwarding* [57].

Multipath-based architectures connect the source node and the sink node via multiple routes (paths). This approach allows data payload to be evenly distributed across the multiple paths. These types of architectures are also popular for real-time streaming data as well as data requiring a special level of reliability. These architectures tend to be power efficient, because they spread the energy load across multiple paths. The most prominent representatives of this type are *Node-Disjoint* and *Braided Path* architectures [58] and *N-to-1 Multipath Discovery* [59], *Secure and Energy Efficient Multipath (SEEM)*[60] *, Robust and Energy Efficient Multipath Routing* [61], and *Hierarchy-based Multipath Routing Protocol* [62].

Heterogeneity-based architectures imply a network with multiple types of nodes. Nodes within the same network might be split into battery operated nodes vs. power-operated nodes, sensing nodes vs. communication nodes, or nodes with processing power vs. nodes with sensor units. In all cases, the goal is to optimize the network in order to best utilize each node's available resources. *Information-driven Sensor Querying (IDSQ)* and *Constrained Anisotropic Diffusion Routing (CADR)* architectures [63], *Cluster Head Relay (CHR)* [64], *Heterogeneous Disjoint Multipath Routing Protocol (HDMRP)* [65]*, Stable Election Protocol (SEP)* [66]*, and Energy Efficient Heterogeneous Clustered scheme (EEHC)* [67] fall into this category.

2.7 Summary

This section presented an overview of WSN architectures. We grouped all major published work in distinct functional groups. All architectures had in common the ability to extend the network lifetime. The network lifetime is the most relevant performance metric for this dissertation. All architectures, however, differed in other performance parameters such as packet latency, network security, quality of service, geographical awareness, and data and network centricity. This section described each architecture separately, pointing out its advantages and disadvantages. The next section introduces a novel *Power Equilibrium Wireless Sensor Network* (PEWSN) architecture.

CHAPTER 3. PEWSN ARCHITECTURE

3.1 Introduction

The PEWSN is a hierarchical architecture with the following key features:

- *Smart Cluster Head Selection* –to extend the network lifetime, the cluster head selection algorithm takes into account not only battery status (BatteryStatus) [1], [19] [21] but also the sensor's harvesting capability (BatteryChargingHealth) and the type and level of service that it can provide to the network (ServiceLevel).
- *Adjustable communication bandwidth–the* PEWSN network is capable of detecting the overall power health of the network on each hierarchical level (sensor, cluster, and network). Consequently, PEWSN can control and adjust its data bandwidth and reach the power equilibrium necessary for extending the network lifetime. Given that the bandwidth adjustments are usually temporary and depend on power harvesting oscillations (day vs. night, cold vs. warm, static vs. dynamic), PEWSN is capable of extremely fast bandwidth adjustment (within one TDMA round).
- *Contention-free Reliable Communication* – PEWSN TDMA mode ensures contention-free communication among individual sensors and cluster heads. PEWSN guarantees that each sensor's data will arrive at the base within a bounded maximum latency (one TDMA round).
- *Seamless bidirectional network control, monitoring, configuration, and reconfiguration*–special events such as cluster head switching, network tree repairs and restructurings, data compression methodology changes, special

base-initiated sensor applications, and firmware code updates and other network maintenance tasks are seamless. This ensures uninterrupted data flow and minimal power overhead.

- *Embedded network geolocalization*–this further expands PEWSN's potential applicability and adaptability to various commercial, industrial, and military applications by adding geographical awareness of individual nodes to the network.
- *Embedded network security support* is an optional feature that does not add any extra setup cycles. This feature, however, is power costly, because it includes encrypting and decrypting transmitted data packets.

Features such as smart cluster head selection and adjustable communication bandwidth are needed in order to extend the overall network lifetime. Contention-free reliable communication is a highly sought feature in latency-sensitive, trigger-event WSNs such as those in military and homeland security applications. Seamless bidirectional network control reduces the overall network cost by making the network easier to maintain and to monitor. Embedded node geolocalization is needed in applications requiring precise location of an event (e.g. oil pipe leaks, intrusion detection, engine failure, etc.). Finally, network security based on symmetric cryptography protects the network against unwanted Sybil, wormhole, sinkhole, selective forwarding, and hello flood attacks. Both geolocation and network security are optional features and therefore do not have to be implemented in all PEWSN-based applications.

3.2 Building Blocks

The PEWSN architecture consists of the base station (central data gathering place), individual field sensors, sensor clusters (groups of individual sensors), and cluster heads. In PEWSN, the hardware found in cluster heads is identical to that in individual field sensors, but they carry additional tasks and responsibilities. As in [21], we assume that each sensor can adjust the transmit power [68] in order to support intra-cluster communication as well as inter-cluster communication. We also assume that all sensors within the same cluster are within transmission reach of each other.

Fig. 16 shows an example of the PEWSN configuration. Each sensor is denoted as Sx[y] where *x* is cluster_id (cluster address) and *y* is sensor_id (sensor address) within the cluster. Therefore, sensor S1[2] has sensor_id=2 and belongs to cluster 1. The sensor_id and the cluster_id uniquely identify each individual sensor within PEWSN. The base is S0[0]. The number of clusters and sensors is fully scalable.

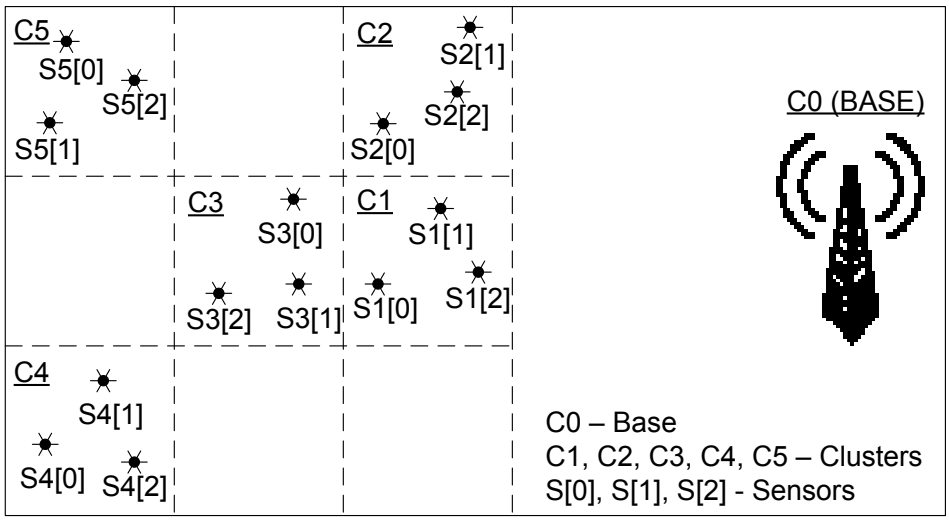


Fig. 16. PEWSN architecture with its major building blocks

Fig. 17 shows hardware components of each individual sensor. Power consumption, the critical network lifetime parameter, varies greatly between individual sensor components (units).

<table>
<tr><td>Sensing Unit</td><td>Processing Unit</td><td>Communication Unit</td></tr>
<tr><td>Battery Unit</td><td colspan="2">Power Harvesting Unit</td></tr>
</table>

Fig. 17. Sensor (node) individual components (units).

A communication unit [68] consumes two orders of magnitude more power than a processing unit [69]. Shrestha and Xing [70] have shown that a sensor requires around 100 to 1000 times more energy to transmit a bit than to execute an instruction. This is partially because sensors are usually positioned close to the ground where wireless communication degradation factors such as ground reflection, signal shadowing and fading, require more transmission power to overcome [71]. Merrill et al. [72] also conclude that the environment can greatly impact propagation loss.

Sensing unit power consumption depends greatly on detector type and precision. Motion sensors consume a few microwatts, while acoustic sensors can consume hundreds of microwatts. An accelerometer with a limited precision and measuring range needs no more than 1 mW, while a high end accelerometer with precision of ~10 ng/rtHz can consume 100 mW. Battery technology has advanced greatly in recent years. Now, a 1”x1” thin-film solid-state battery can deliver 1 mAh [73]. However, the lifetime of the battery is greatly affected by variations in operating temperature and the sensor’s current draw behavior.

3.3 Transmission Channel Access Method

The PEWSN uses time division multiple access (TDMA) mode as its transmission channel access method. In the case of a multi-cluster network, TDMA is supplemented with code division multiple access (CDMA), resulting in an overall time-slotted-CDMA approach. CDMA codes are used to avoid contention between neighboring clusters. The time reservation based method was chosen over contention-based methods, since PEWSN is used in highly reliable, latency-bounded applications.

The PEWSN TDMA structure (Fig. 18) is split into three sections:

- CLUSTERUP
- SENSING
- CLUSTERDOWN

CLUSTERUP and CLUSTERDOWN slots are exclusively used by cluster heads

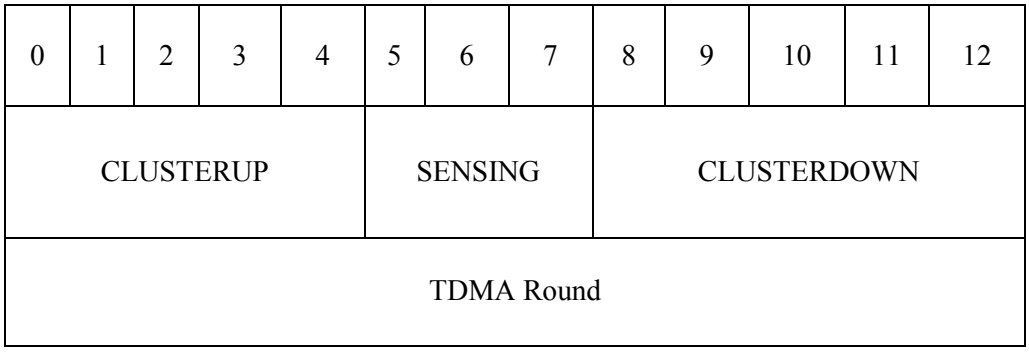


Fig. 18. PEWSN TDMA Schedule with 5 clusters and 3 sensors per cluster

to transfer packets up and down the network tree.

In CLUSTERUP sections, all packets originate in lower clusters and are transferred to higher clusters (in terms of network tree, with base being the lowest cluster/ root cluster). For example, cluster 0 uses slot 0 to transfer packets to

cluster heads directly connected to the base. In slot 1, cluster 1 sends its packet to directly connected upper cluster heads. The number of TDMA slots within a CLUSTERUP section is equal to the number of clusters within PEWSN. The CLUSTERUP TDMA section is absolutely contention-free, because there is always only one cluster head (or base) transmitting in each TDMA slot.

In SENSING sections, all packet transfers occur within the clusters themselves. All sensors within one cluster sense and send their data to their cluster head. Each sensor uses one slot to transfer the data. The number of TDMA slots within the SENSING section is equal to the number of available sensors within the cluster. The optimal PEWSN configuration has an equal number of sensors within each cluster; hence, no TDMA slot is wasted in any PEWSN cluster. Unlike the CLUSTERUP and CLUSTERDOWN sections, a pure

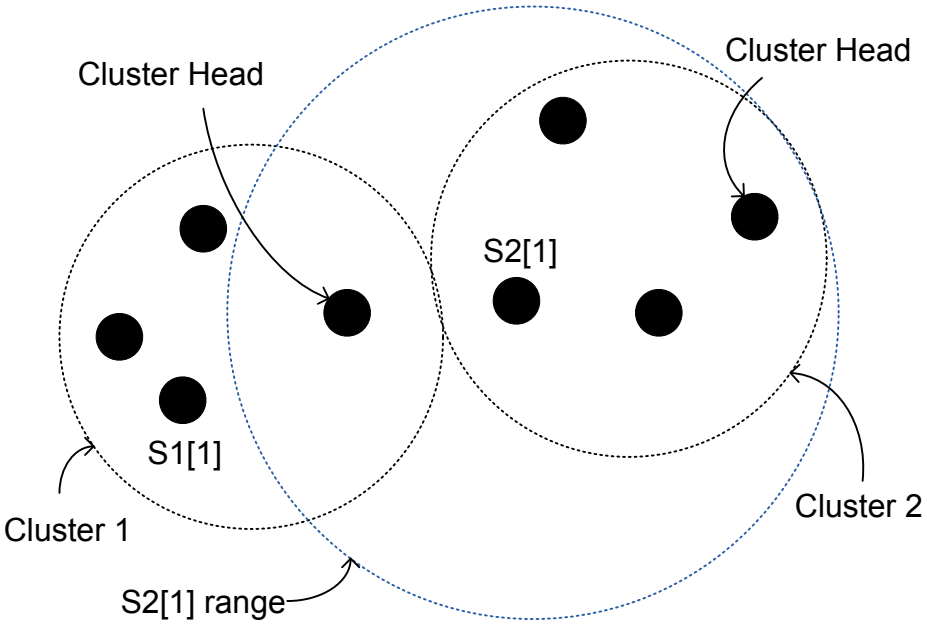


Fig. 19. SENSING phase contention

TDMA does not guarantee a full contention-free communication in the SENSING section.

Fig. 19 shows node S2[1] belonging to cluster 2. Here, S2[1] is scheduled to transmit its data to cluster 2's head. Because of S2[1]'s location, the wireless transmission range is such that S2[1]'s transmission is also heard by cluster 1's head. Yet, the same TDMA slot in cluster 1, slot 1, is occupied by the S1[1] node belonging to cluster 1. Therefore, packet collision between nodes S1[1] and S2[1] will occur, resulting in data corruption at cluster 1's head. To avoid this scenario, PEWSN implements code division multiple access (CDMA) on top of its TDMA schedule, resulting in a time-slotted CDMA approach. Due to the need to use CDMA, the transmission bandwidth is lower than in a pure TDMA approach. The precise CDMA parameters such as code size and chip code, depend on the exact network topology, i.e. the number of neighboring clusters and the transmission range of individual sensors. The transmission range is proportional with $\frac{1}{d^{\alpha}}$ where d is the distance between two nodes and α is the environmental transmission coefficient, ranging from 2 to 4 (in extreme urban, multi-path environments it can reach even 5). The contention is possible only in the SENSING section, because both CLUSTERUP and CLUSTERDOWN are assured to have only one cluster head potentially transmitting at each TDMA slot.

In the CLUSTERDOWN section, packets stream from upper cluster heads to lower cluster heads, eventually reaching the base (cluster C0). The CLUSTERDOWN section involves real data streaming, because the data is sensed using individual sensors and propagated back to base in wireless sensor

networks. The number of TDMA slots within a CLUSTERDOWN section is equal to the number of clusters within the PEWSN. The CLUSTERDOWN section has an equal number of slots to the CLUSTERUP section, but the slot period for those two sections might be different. Therefore, the overall TMDA round latency L is:

$$L = (\overbrace{c \cdot t_{CLUSTERUP}}^{CLUSTERUP\ Section}) + (\overbrace{s \cdot t_{SENSING}}^{SENSING\ Section}) + (\overbrace{c \cdot t_{CLUSTERDOWN}}^{CLUSTERDOWN\ Section})$$
$$L = c \cdot (t_{CLUSTERUP} + t_{CLUSTERDOWN}) + s \cdot t_{SENSING}$$
$$if \quad t_{CLUSTERUP} = t_{CLUSTERDOWN} = t_{SENSING} \quad then$$
$$L = (2 \cdot c + s) \cdot t_{slot}$$

Where

L – latency
$t_{CLUSTERUP}$ – CLUSTERUP slot period
$t_{SENSING}$ – SENSING slot period
$t_{CLUSTERDOWN}$ – CLUSTERDOWN slot period
c - number of clusters
s - number of sensors

The most optimal PEWSN-based network topology is with all clusters having the same number of sensors. In that case, the SENSING section of all clusters does not have any wasted TDMA slots. However, in an imbalanced network where some clusters have only a few sensors and others have a large number of sensors, there will be significant bandwidth waste in less populated clusters. The number of TDMA slots in the SENSING section is defined by the number of sensors in the largest cluster within the network. However, sub-optimal bandwidth use in imbalanced networks might not be relevant for all applications.

3.4 Clock Synchronization

The most challenging issue in TDMA-based wireless communication is global clock synchronization. Kopetz and Ochsenreiter [74] define the issues associated with the synchronization of a number of different time references found in TDMA-based distributed systems. Two levels of synchronization are necessary: internal and external. External synchronization generates the internal time by synchronizing the approximate global time base with the external physical time. Internal synchronization, on the other hand, constructs an approximate global time base among the network's sensors.

Trying to solve the same problem, Elson et al. [75] coined the term "Periodic Global Broadcast Time Synchronization (PGB-TS)." They recommend a periodic broadcast of beacon messages to synchronize the clocks of neighboring sensors in an ad-hoc network. Bekmezci and Alagoz [76] use the concepts described in [74] and [75] to architect a TDMA-based sensor network for military monitoring (MIL-MON). The proposed solution reduces the power associated with global clock synchronization.

The PEWSN uses the concepts described in [74], [75] and [76] to synchronize the clocks of individual sensors. In the CLUSTERUP TDMA section, beacon messages are sent to synchronize clocks of cluster heads. In the SENSING section, cluster heads broadcast the beacon message to synchronize the clocks of all individual sensors with their clusters. Hence, global clock synchronization within PEWSN is achieved.

3.5 PEWSN Phases

The PEWSN contains three phases:

- *Setup (Initialization)*–PEWSN's sensors are organized into clusters, and clusters are organized into a network tree. All cluster heads are selected and linked together. It takes two TDMA rounds to set up a single PEWSN cluster. Therefore,

  $$total\_setup\_rounds = 2 * number\_of\_clusters$$

- *Transition Phase* –all clusters and all sensors are synchronized so that the data phase can begin. A message is sent from the base to all cluster heads and from all cluster heads to all individual sensors announcing the start of the data phase. This phase is important for proper global clock and timing synchronization among all network nodes and clusters.
- *Data Phase* –sensor nodes sense the environment and report the findings back to their corresponding cluster heads, which then process and report the data back to the base. Within each TDMA round, PEWSN guarantees to report data from each sensor and each cluster no matter how far the sensor/cluster is. All network reconfigurations are seamless. For example, in the cluster head switching phase, the particular cluster announces the switch to all sensors within the cluster and to all linked neighboring cluster heads. However, since the switch occurs without data flow interruption, the phase is considered a special case and treated as such.

3.6 Messages

The PEWSN defines 13 distinct messages with which it is able to perform all functionalities:

- *TDMASETUP_PKT (id=1)*— sets up a new uninitialized cluster.
- *TDMASETUPFORWARD_PKT (id=2)*— forwards the setup request to the new uninitialized cluster via other already initialized cluster heads (ad-hoc network).
- *SENSORSTATUS_PKT (id=3)*—sent during initialization in order to select the cluster head. Each individual sensor sends its status in terms of ChargingHealth, BatteryStatus and ServiceLevel to the lower cluster head.
- *SENSORSTATUSFORWARD_PKT (id=4)*— same content as id 3 message. Contains additional routing information needed when the base is not directly reachable but rather the packet must go through the network of cluster heads.
- *HEADSETUP_PKT (id=5)*—advertises the cluster head choice to the sensor. This message originates at the base and is forwarded through cluster heads until it reaches the cluster whose setup is still in progress.
- *HEADSETUPWITHINCLUSTER_PKT (id=6)*— advertises the new cluster head to all sensors within the setup cluster.
- *CLUSTERSETUPDONE_PKT (id=7)*— the new cluster's head acknowledges its selection back to the base. With this message. the new cluster is fully setup and waiting for the data phase transition.

- *NETWORKSETUPDONE_PKT (id=8)*— end of setup phase is advertised to all cluster heads. This message, sent only during the transition phase, re-synchronizes clocks among cluster heads.
- *NETWORKSETUPWITHINCLUSTERDONE_PKT(id=9)* —end of setup phase is advertised to all individual sensors via their respective cluster heads. This message, sent only during the transition phase, is sent by cluster heads to their respective sensors. The message re-synchronizes the clocks of all individual sensors within the cluster.
- *SENSORDATA_PKT (id=10)*— contains sensed data sent from each individual sensor to its cluster head.
- *HEADDATA_PKT (id=11)*— accumulated/aggregated data from all sensors within one cluster and one TDMA round sent down the network tree, ultimately reaching the base. The message content is fully configurable, allowing for different types of data compression and data filtering.
- *INTERNALHEADSWITCH_PKT (id=12)*—current cluster head advertises the selection of the new cluster head to all sensors within the particular cluster. The switch is triggered due to the PEWSN cluster head switch algorithm that takes into account the minimum energy and backoff thresholds.
- *MAINTENANCE_PKT (id=13)*—propagated from lower cluster heads to upper cluster heads advertising all cluster head changes, power level configuration requests, data type collection changes, etc. This message is sent only during the TDMA CLUSTERUP section.

3.7 PEWSN (5,3) Example

Fig. 20, Fig. 21, and Table 2 show an example of the PEWSN-based wireless sensor network. The PEWSN (5,3) example contains five field clusters (C1 through C5), each having three sensors (S[0], S[1] and S[2]). The base C0 can directly communicate only with C1 and C2. Each TDMA round has 13 TDMA slots ($k = 13, c = 5, s = 3$). Special features, such as seamless cluster head switching, geolocalization, and security, are not covered in this example.

Fig. 21 shows the PEWSN (5,3) example's finite state machine (FSM) where C is the number of clusters, S is the number of sensors within each cluster, CH is the cluster head, C_{now} is the current cluster, and T is time.

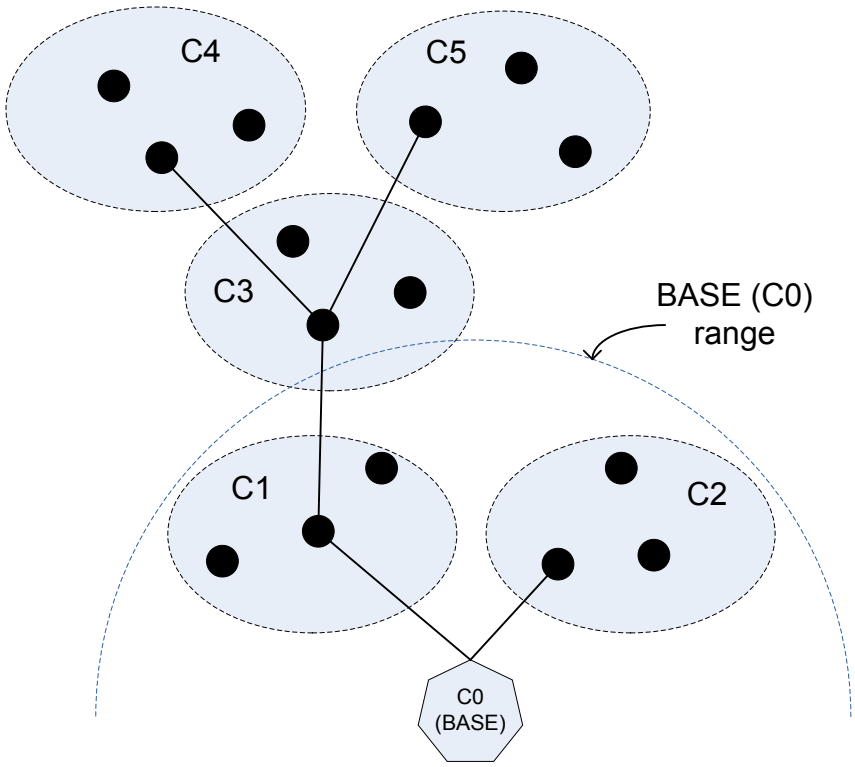


Fig. 20. PEWSN (5,3) configuration example

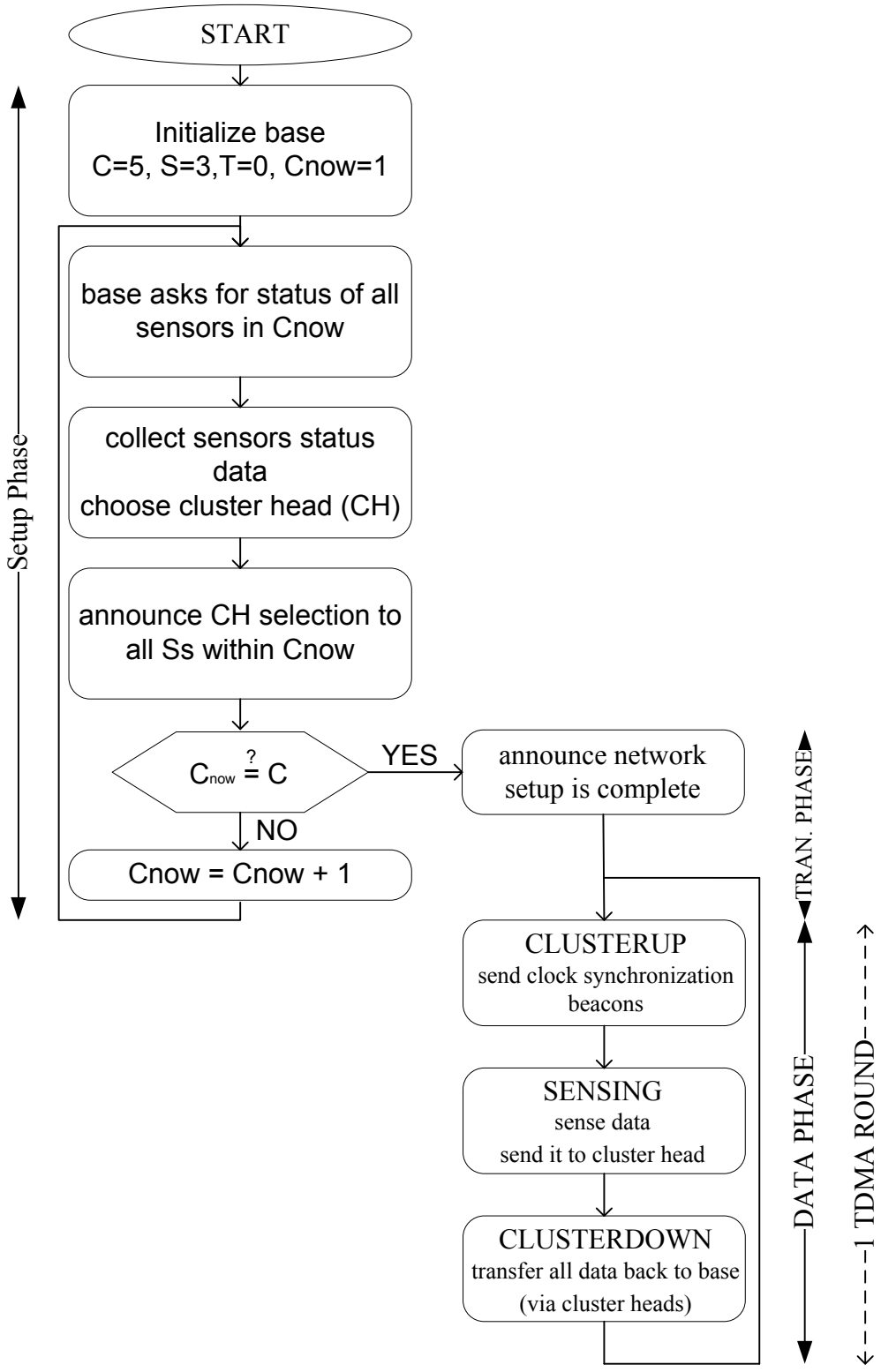


Fig. 21. PEWSN (5, 3) FSM

Table 2 shows a complete TDMA schedule for the PEWSN (5, 3) example. Each TDMA round is made of five CLUSTERUP TDMA slots (0 through 4), three SENSING TDMA slots (5 through 7), and five CLUSTEDOWN TDMA slots (8 through 12). Therefore, there are overall 13 TDMA slots per each round. The table also shows the owner of each slot. In the CLUSTERUP and CLUSTERDOWN sections, the owners are cluster heads, while in SENSING section, the owners are individual sensors.

The CLUSTERUP section belongs to the cluster heads. All other individual sensors sleep during that time. In slot 0, the base C0 can send messages to neighboring clusters C1 and C2. In slot 2, C1 can send messages to C3. As per this particular network configuration, C2 does not have upper links; therefore, TDMA slot 2 is always unpopulated. In TDMA slot 2, no messages are being sent. The same is true for slot 4, where C4 does not have any links. Since C5 is the last cluster in the network and does not contain the upper link, it does not have its own CLUSTERUP slot.

The SENSING section only involves individual sensors. There is no communication across clusters during this time. In slot 5, the sensor with id=0 transfers its data to its cluster head. Since there are multiple cluster heads, five individual sensors (one in each cluster) are active at this time. The same is true for slot 6 (all individual sensors with id=1 are active) and slot 7 (all individual sensors with id=2 are active).

The CLUSTERDOWN section again only involves only cluster heads. This time sensed data is moved downstream toward the base. In slot 8, cluster 5

(C5) transmits data to C3. In slot 4, C4 transmits data to C3. In slot 10, cluster 3 (C3) aggregates, filters, or compresses data from C4 and C5 and sends it down the link to C1. In slot 11, C2 sends data to the base; in slot 12, C1 sends data to C0. With slot 12, one TDMA round is complete. In this example, the setup phase takes 10 rounds; each cluster needs two rounds to be fully setup.

During setup phase round 1, there are only four messages exchanged. The first message with id=1 (TDMASETUP_PKT) is sent in TDMA slot 0 from C0 to all sensors within an unutilized cluster 1 (broadcast message). The message automatically creates TDMA schedules within the receiving nodes, which then send their responses (id=3, SENSORSTATUS_PKT) in slot 5 (sensor id = 0), slot 6 (sensor id = 1), and slot 7 (sensor id = 2). In round 2 (RND=2), slot 0 chooses the cluster head for C1 and communicates the decision via message id=5 (HEADSETUP_PKT). The selected sensor (in this case, the sensor with id=1) advertises the decision to all other sensors within the cluster via message id=6 (HEADSETUPWITHINCLUSTER_PKT). Finally, the same cluster head completes the C1 setup phase using TDMA slot 12 to send message id=7 (CLUSTERSETUPDONE_PKT) back to the base.

C2 is setup in the same way as C1. C3 is also setup in the same way, except there are additional TDMASETUPFORWARD_PKT (id=2) and SENSORSTATUSFORWARD_PKT (id=4) messages needed to account for the fact that the C3 to base link is a multi-hop link traversing through C1. Therefore, C2 is used as the forward cluster for messages between the base and C3.

The same is true for clusters C4 and C5, whose messages must traverse through C3 and C1 to reach the base. Messages id=1 through id=7 are only used in the setup phase.

The transition phase takes only one round (RND 11). At the start of this round, all clusters are assumed to be setup and waiting for the trigger to move to the data phase. This is done via message NETWORKSETUPDONE_PKT (id=8). This message traverses through all cluster heads advertising the start of the next phase. This message is also used to synchronize cluster heads clocks, so that no large time margins between TDMA slots are necessary. While NETWORKSETUPDONE_PKT (id=8) is used for synchronizing all cluster heads, NETWORKSETUPWITHINCLUSTERDONE_PKT(id=9) is used for synchronizing all individual sensors within their respective cluster heads. In other words, all cluster heads send message id=9 to all sensors within their clusters advertising the transition to the next data phase and synchronizing their local clocks so the TDMA mode can function properly.

Finally, the data phase uses messages SENSORDATA_PKT (id=10) and HEADDATA_PKT (id=11) to transfer aggregated, filtered, or compressed data from individual sensors to the base. As Table 2 shows, in the first data round (RND=1) message id=10 is sent from all individual sensors to their respective cluster heads, and id=11 is sent through cluster heads down the network tree back to the base. The network continues to operate in the data phase until its end of life.

	CLUSTERUP					SENSING			CLUSTERDOWN				
slot #	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
slot owner	**C0**	**C1**	**C2**	**C3**	**C4**	**S[0]**	**S[1]**	**S[2]**	**C5**	**C4**	**C3**	**C2**	**C1**
link	**C0>C1 C0>C2**	**C1>C3**		**C3>C4 C3>C5**					**C5>C3**	**C4>C3**	**C3>C1**	**C2>C0**	**C1<C0**
SETUP PHASE													
RND	at this point nothing is setup/initialized - goal: base(C0) is to set up the first cluster(C1)												
1	#1					#3	#3	#3					
2	#5						#6						#7
	at this point C1 cluster is fully setup - next goal: set up cluster C2												
3	#1					#3	#3	#3					
4	#5						#6						#7
	at this point C1,C2 clusters are setup - next goal: set up cluster C3												
5	#2	#1				#3	#3	#3					#4
6	#5	#5						#6			#7		#7
	at this point C1,C2,C3 clusters are setup - next goal: set up cluster C4												
7	#2	#2		#1		#3	#3	#3			#4		#4
8	#5	#5		#5		#6				#7	#7		#7
	at this point C1,C2,C3,C4 clusters are setup - next goal: set up cluster C5												
9	#2	#2		#1		#3	#3	#3			#4		#4
10	#5	#5		#5		#6				#7	#7		#7
TRANSITION PHASE													
	at this point ALL clusters are set up - next goal: transition to Data Phase												
11	#8	#8		#8		#9	#9	#9					
DATA PHASE													
	at this point network is in Data Phase - next goal: data collection												
1						#10	#10	#10	#11	#11	#11	#11	#11

Table 2. PEWSN (5,3) TDMA message schedule

CHAPTER 4. PEWSN PERFORMANCE

4.1 PEWSN Network Lifetime Comparison

In this section, we compare a number of popular WSN protocols:

- Direct Routing
- Minimum Transmission Energy (MTE) Routing
- Static Clustering
- Low-Energy Adaptive Clustering Hierarchy (LEACH)
- Power Equilibrium Wireless Sensor Network (PEWSN)

4.2 Comparison Protocols Overview

In direct routing based protocols, each node sends the data directly to the base. In direct routing, randomly distributed nodes may experience substantial differences in their distance to the base. Nodes found far away from the base suffer from long-distance transmission power loss. In other words, nodes located far away from the base will deplete their energy much faster than the nodes closer to the base. In direct routing, the transmission power consumption dominates the electronics power consumption.

Minimum transition energy (MTE) routing solves the problem imposed by direct routing by sending each packet through multiple nodes (hops) before it reaches the base (final destination). The energy usage is optimized by always choosing the closest hops so that the energy needed for the transmission is reduced. In MTE routing, the nodes closest to the base deplete their energy faster, because they are utilized more often than other nodes. Therefore, the nodes

closest to the base die relatively quickly, while the far distant nodes can survive longer.

Static clustering takes advantage of the clustered architecture by creating clusters for each set of neighboring nodes. The advantage of this protocol is the fact that most of the nodes communicate with their cluster heads over a short distance. The only long distance transmission is from a cluster head to the base. Static clustering suffers from the fact that the cluster head, once selected, never changes. Therefore, the cluster head node that must listen to all other nodes and then transmit all or at least part of their data back to the base depletes its energy much faster than the individual nodes in the cluster. There are many architectural variations and improvements to the static clustering algorithm. Some improvements are based on smart selection of cluster head nodes; i.e. they take into account node energy levels, cluster head proximity to neighbors within the same cluster, cluster head to cluster head distances, and other useful information that might help extend the network lifetime.

LEACH recognizes the shortcoming of static clustering by introducing cluster head switching. In LEACH, the cluster head node is not static; rather, it rotates between all nodes. Therefore, each node usually has multiple chances to become the cluster head. The creation of clusters, cluster heads, and cluster head switching is done using a probabilistic function. LEACH achieves great lifetime improvement over all above-mentioned routing protocols, but it still suffers from the suboptimal not-energy-based cluster and cluster head selections. The protocol

also has a long repetitive setup phase that might substantially degrade the performance if the data rounds are short and low-bandwidth.

4.3 Simulation Input Parameters

In order to have a fair comparison, we use the network parameters and assumptions described in [1]:

- Each node's initial energy = 0.5 J
- transmitter electronics (E_{tx}) = receiver electronics (E_{rx}) = electronics (E_{elec}) = 50nJ/bit
- transmit amplifier (E_{amp}) = 100 pJ/bit/m^2
- data processing = 5nJ/bit
- message size = 2000 bits
- field size = 50 m x 50 m
- Number of sensor nodes = 100
- Base Location = (25,0)
- LEACH specific parameters
    - Percentage of cluster heads = 5% (optimal number as per [1])
- PEWSN specific parameters
    - No. of clusters = 4
    - No. of sensors per cluster = 25
    - Cluster head switching threshold = 0.45 J
    - Cluster head backoff coefficient = 2

4.4 Simulation Field Visualization

In order to better visualize the above-mentioned parameter constraints, Fig. 22 shows the PEWSN-based simulation example during round 1. The field is 50 m x 50 m with the base at (25,0) . The field is split into four quadrants (clusters) each containing 25 sensors randomly distributed throughout the field (blue circles). The first quadrant (bottom, left) is cluster 1; the second quadrant (bottom, right) is cluster 2; the third quadrant (upper, left) is cluster 3; and the fourth quadrant (upper, right) is cluster 4. The initial cluster head (light blue filled circle) as per the PEWSN algorithm is closest to the midpoint of each quadrant (blue cross). Currently, all clusters are blue circles, which means that there are no dead nodes.

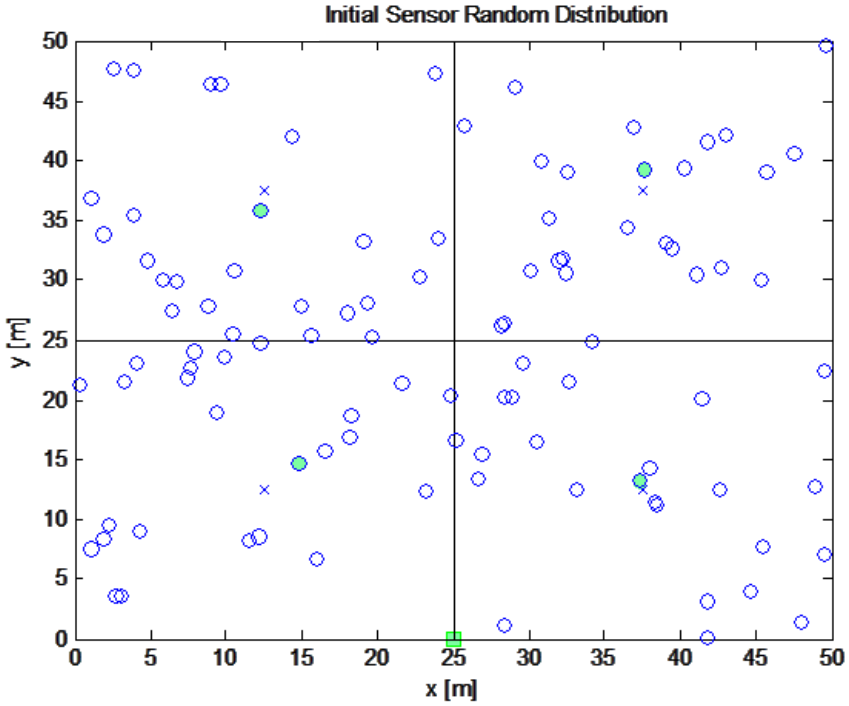


Fig. 22. PEWSN Startup

Fig. 23 shows the state of the simulation at round 1900. The blue circles represent the nodes that are still alive, while the red dots represent the dead nodes. Note that the cluster head has switched from its initial node. The individual cluster fields have not changed since the round 1. This is contrary to the LEACH protocol, which introduces a new setup phase for each data phase. The sensors in LEACH are free to join newly chosen cluster heads based on distance proximity. Unlike in PEWSN, each cluster in LEACH most likely contains a different number of sensors. LEACH cluster borders will also have an irregular shape.

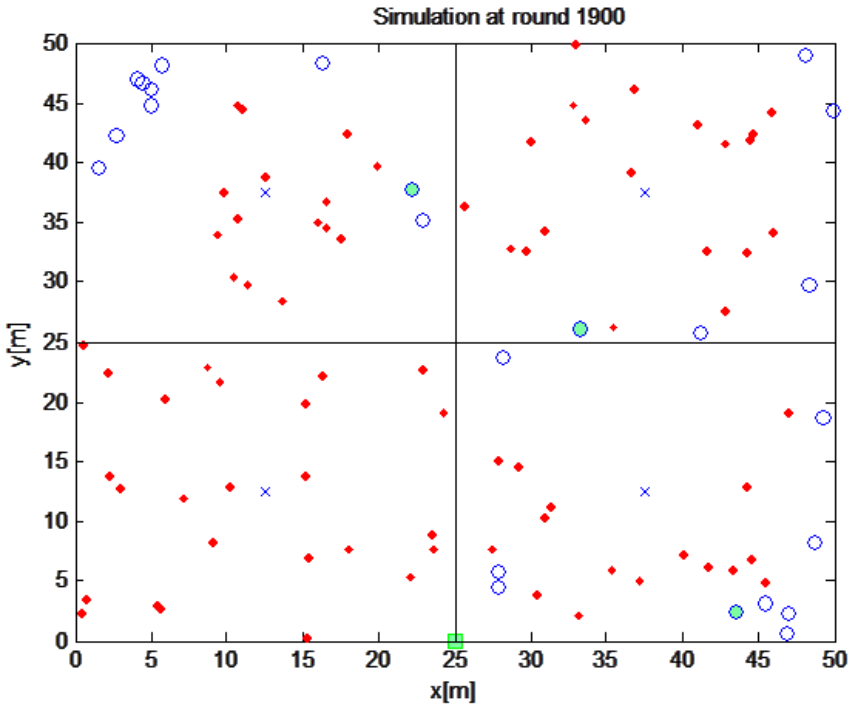


Fig. 23. PEWSN simulation at round 1900

Finally, Fig. 24 shows the state of nodes at the end of simulation (final round). All nodes are depicted as red dots (dead nodes). No further packets are being delivered to the base. All nodes are assumed to have 0 Joules of energy at the end of the simulation.

All other protocols including direct routing, MTE, static clustering, and LEACH use a similar type of simulation model. The individual models also include certain protocol specific features. These models were used as the basis for our network comparisons and final results.

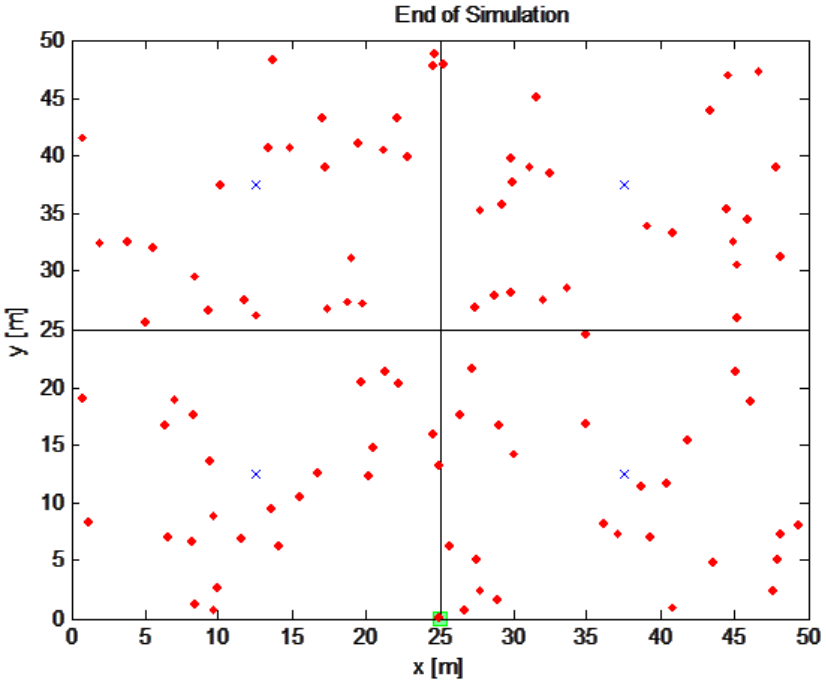


Fig. 24. PEWSN end of simulation

4.5 Results

Table 3 summarizes the comparison results. Heinzelman et al.'s [1] results are partially used for comparison. We can conclude that PEWSN-based network lifetime is significantly longer than the lifetime of any other WSN architecture. This is true whether we use the first dead node or the last dead node as the performance benchmark. The results in Table 3 use the same parameters as Heinzelman et al. simulation. The result, however, would come out even more in favor of PEWSN architecture if adjusted the parameters to represent a more realistic real-world environment (further explored in section 4.6).

From Table 3, we can draw a number of conclusions:

- Direct routing, MTE, and static routing are greatly inferior to LEACH and

Energy (J/node)	Protocol	Round first node dies	Round last node dies
0.5	Direct	109	234
	MTE	8	429
	Static Clustering	80	110
	LEACH	932	1312
	PEWSN	1840	2800

Table 3. WSN Protocols - Lifetime Comparison

PEWSN due to the absence of adequate power usage distribution among nodes via mechanisms such as cluster head switching.

- MTE routing performance is very different for the two benchmarks. When the first dead node benchmark is used, MTE performs very poorly. The fact that traffic in every round goes through the node closest to the base greatly reduces that node's lifetime. The critical node's battery absorbs packet receptions from the upper node and packet transmissions to the base during each round until it finally dies after only eight rounds. On the other hand, as the nodes closest to the base die relatively quickly, nodes far away from the base conserve their energy by only transmitting (but never receiving) packets. Since they never fall within the minimum-routing path, they are only responsible for transmitting packets. Those nodes die slowly, as the last dead node benchmark suggests. The MTE last dead node benchmark outperforms direct routing, because while many nodes are initially still alive, the longest living nodes must transfer their packets only to the closest living nodes within the network. In direct routing, all nodes must transmit directly to the base all of the time. As most of the nodes eventually die, MTE and direct routing have very similar performance, since all remaining MTE nodes must transfer their packets over much longer distances than at the beginning of the network life.
- PEWSN outperforms all other architectures by a significant margin. The only performance-wise comparable network architecture is LEACH. The following are the reasons why PEWSN outperforms LEACH:
    - Number of cluster heads per round

- Position of cluster heads within the network
- No cluster-head to cluster-head hopping
- Setup overhead

The suboptimal number of cluster heads selected for each round can cause LEACH to have too many long-distance transmissions (Fig. 25) or too many of medium-distance transmissions (Fig. 26). LEACH uses the following probabilistic formula to select cluster heads:

$$T(n) = \begin{cases} \dfrac{P}{1 - P * \left(r mod \dfrac{1}{P}\right)} & if\ n \in G \\ 0 & otherwise \end{cases}$$

P = the desired percentage of cluster heads

r = the current round

G = set of nodes that have not been cluster heads in the last $\frac{1}{P}$ rounds

Here, P is selected a priori and might not be the optimal number of cluster heads.

In Fig. 25, we overlaid LEACH topology over the PEWSN topology to depict the case where the above LEACH function T(n) has delivered too many cluster heads. Red circles in Fig. 25 represent extra cluster heads created by LEACH. While the additional cluster heads might reduce some short-distance paths from individual sensors to their corresponding cluster heads, the number of long-distance, power-hungry paths increases. Therefore, the overall network power consumption increases.

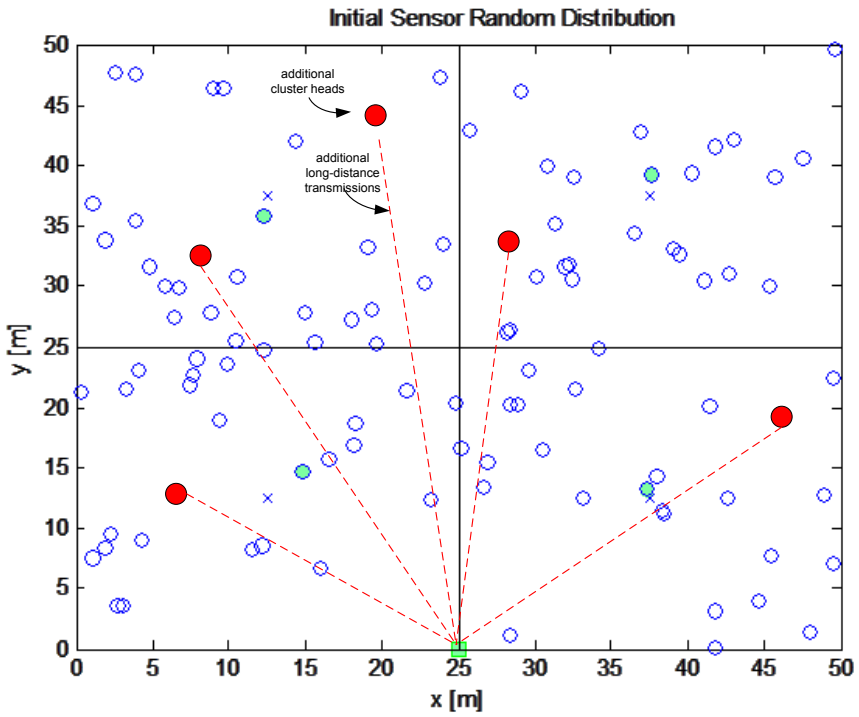


Fig. 25. Too many cluster heads

Fig. 26 shows the case of too few cluster heads. In this case, the number of nodes that must transmit over longer distances to reach their cluster head increases. Therefore, the overall power consumption increases. While Fig. 25 and Fig. 26 represent extreme cases, the likelihood of such cases is high since LEACH restructures its network in each round.

Another reason for the performance difference between PEWSN and LEACH is the fact that LEACH does not have the cluster head to cluster head hoping feature. Therefore, each cluster head must transfer all its data directly to the base no matter how far the base is. When the cluster head does not reduce or compress the amount of data received from all individual sensors

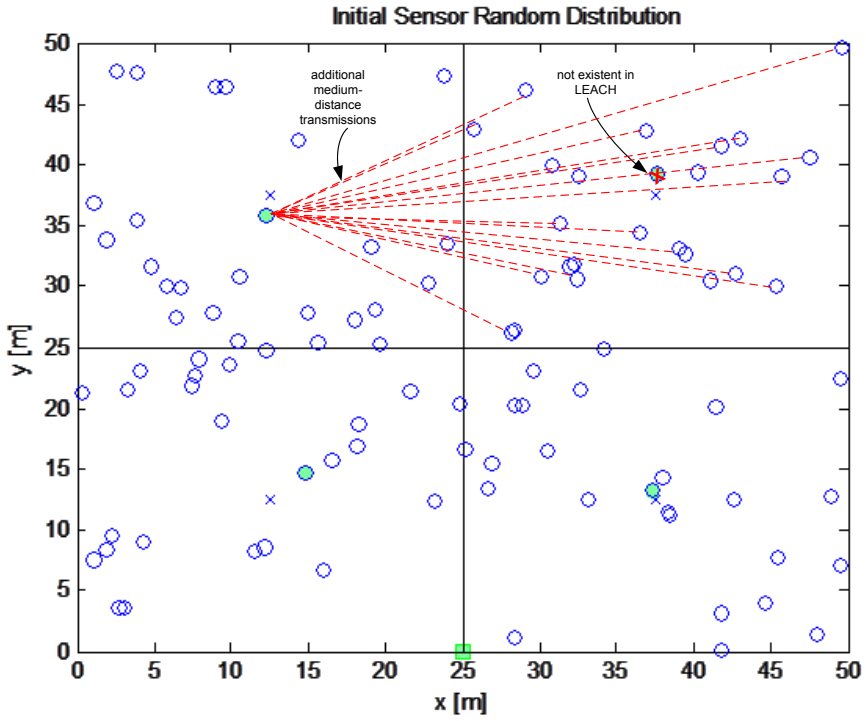


Fig. 26. Too few cluster heads

before shipping the payload to the base, the power consumption due to long-distance, heavy-payload packets is significant.

Finally, the LEACH protocol has an extremely high protocol setup overhead. LEACH reconfigures the entire network in each round. This means that the network must choose all new cluster heads, which cannot be cluster heads from the previous round, and individual sensors must perform join the new cluster heads. This overhead consumes much more power than in the PEWSN protocol, where the network is setup once. and all the cluster head switching is done seamlessly and on-fly during the regular data packet transfer phase.

4.6 Simulation Changes with Performance Impact

While our goal is to keep the simulation parameters in line with the parameters described in [1], we note that the current simulation setup does not favor PEWSN. In fact, the simulation favors the LEACH protocol. Below are the simulation changes that would enhance PEWSN performance and further differentiate it from other WSN protocols:

- Base position (25,0) – having the base positioned close to the sensor field erases some of the benefits that multi-hop routing architectures provide. The current base position allows sensors close to the base to communicate with the base without paying a heavy transmission penalty. LEACH, for example, would spend much more transmission energy if its furthest cluster heads had to communicate directly to a base that is positioned at (25, -25) or even (25, -50).
- The transmission amplifier energy equation used in [1] is:

$$E_{transmit} = e_{amp} * k * d^2$$

$$d - transmission\ distance$$

$$\varepsilon_{amp} - transmit\ amplifier$$

$$k - number\ of\ bits$$

This is only true if the environment does not exhibit ground reflection effects such as signal shadowing and fading. With these effects, the equation above would change to:

$$E_{transmit} = \varepsilon_{amp} * k * d^4$$

Therefore, the energy lost due to signal transmission would substantially increase for inter-node communication. This change would adversely affect direct routing, static cluster routing, and LEACH (uses direct connection from any of its cluster heads to the base; does not use multi-hop cluster head to cluster head transfers). The change would benefit MTE and PEWSN routing.

- Radio model used in this simulation is:

$$transmit\colon\ E_{T_x}(k,d)=E_{elec}*k+\varepsilon_{amp}*k*d^2$$

$$receive\colon\ E_{R_x}(k)=E_{elec}*k$$

$$E_{elec}=50\ {}^{nJ}/_{bit}$$

$$\varepsilon_{amp}=100\ {}^{pJ}/_{{bit}/_{m^2}}$$

Given today's technology, this radio model can be used to favor electronics power consumption over transmission power. Current electronics commercial off-the-shelf (COTS) components use half of the power used in this model. On the other hand, the impact of transmit amplifiers on power budget might be even higher in environments such as the under tick foliage, heavy cement, heavy rain, or concrete floors.

CHAPTER 5. NETWORK SIMULATOR

5.1 Overview

This chapter describes a custom-built PEWSN network simulator that uses the OMNET++ framework and includes all PEWSN features discussed in previous chapters. The simulator is fully scalable in terms of the number of clusters, the number of sensors per cluster, and the data payload size. The simulator is also fully configurable in terms of network topology, TDMA slot duration, hardware components/parts used in each individual sensor, and data transfer type (aggregation, filtering, and compression). In order to simulate the network more realistically, the simulator is fully capable of randomizing parameters such as node positions, position and time of event trigger, and dynamic changes in environment.

Consistent with the OMNET++ framework, the PEWSN simulator tracks each sensor individually and graphically displays the entire network topology along with all communication messages traversing back and forth between nodes. The graphical interface can also accurately depict the environment in which the network exists (terrain, obstacles, rivers, woods, oceans etc.). In addition to the graphical interface, which is dynamically updated, the simulator also supports the recording of chosen data in scalar and vector form. The data file is especially valuable if third-party tools (MATLAB, Excel, PHP) are used for final data post-processing and analysis. All events are logged in a separate file and can be displayed within OMNET++ framework environment.

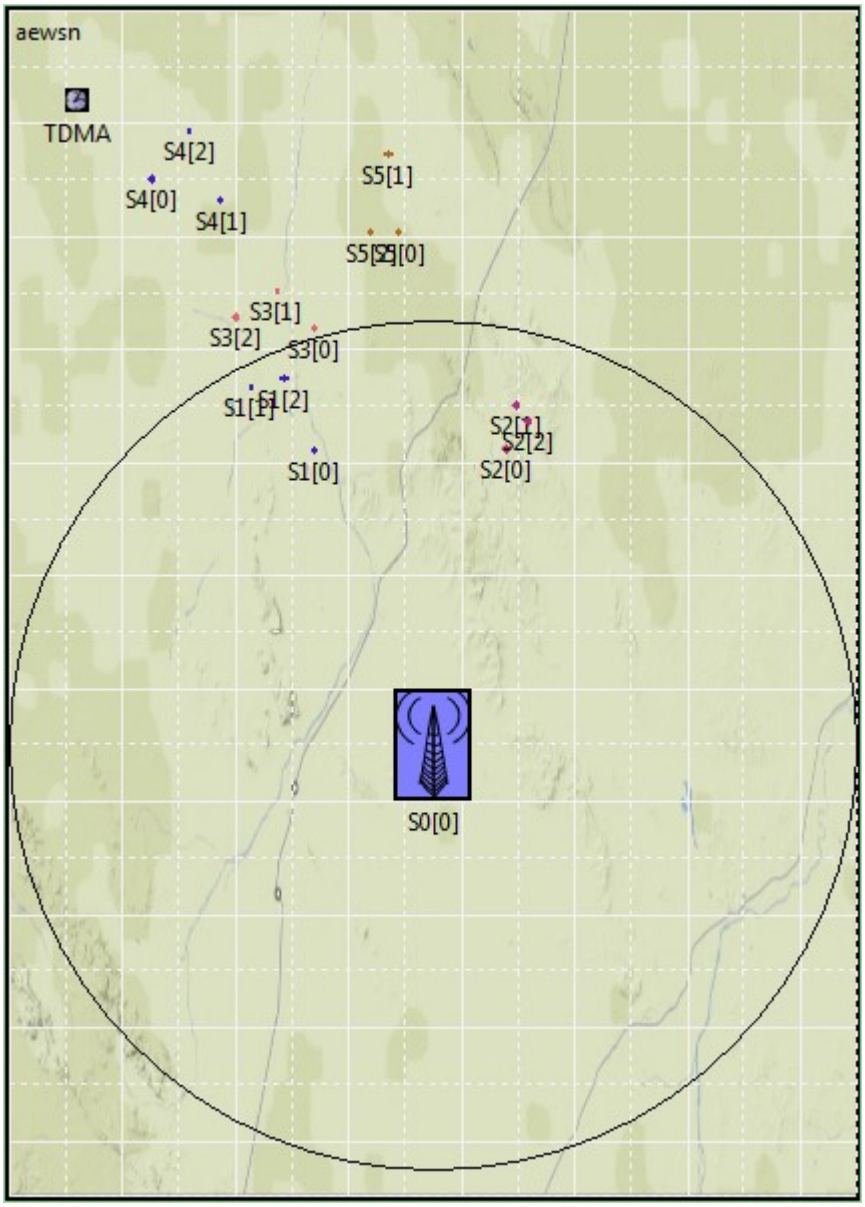


Fig. 27. PEWSN simulator - graphical example

An example of the PEWSN simulator's graphical interface is depicted in Fig. 27. The figure represents the network topology found in the PEWSN (5, 3) example described in section 3.7. It depicts the terrain as well as random-located sensor nodes and clusters. The base is depicted in the center along with the circle signifying the base's transmission range.

5.2 PEWSN Simulator Structure

Fig. 28 shows the PEWSN block diagram with all major design entities.

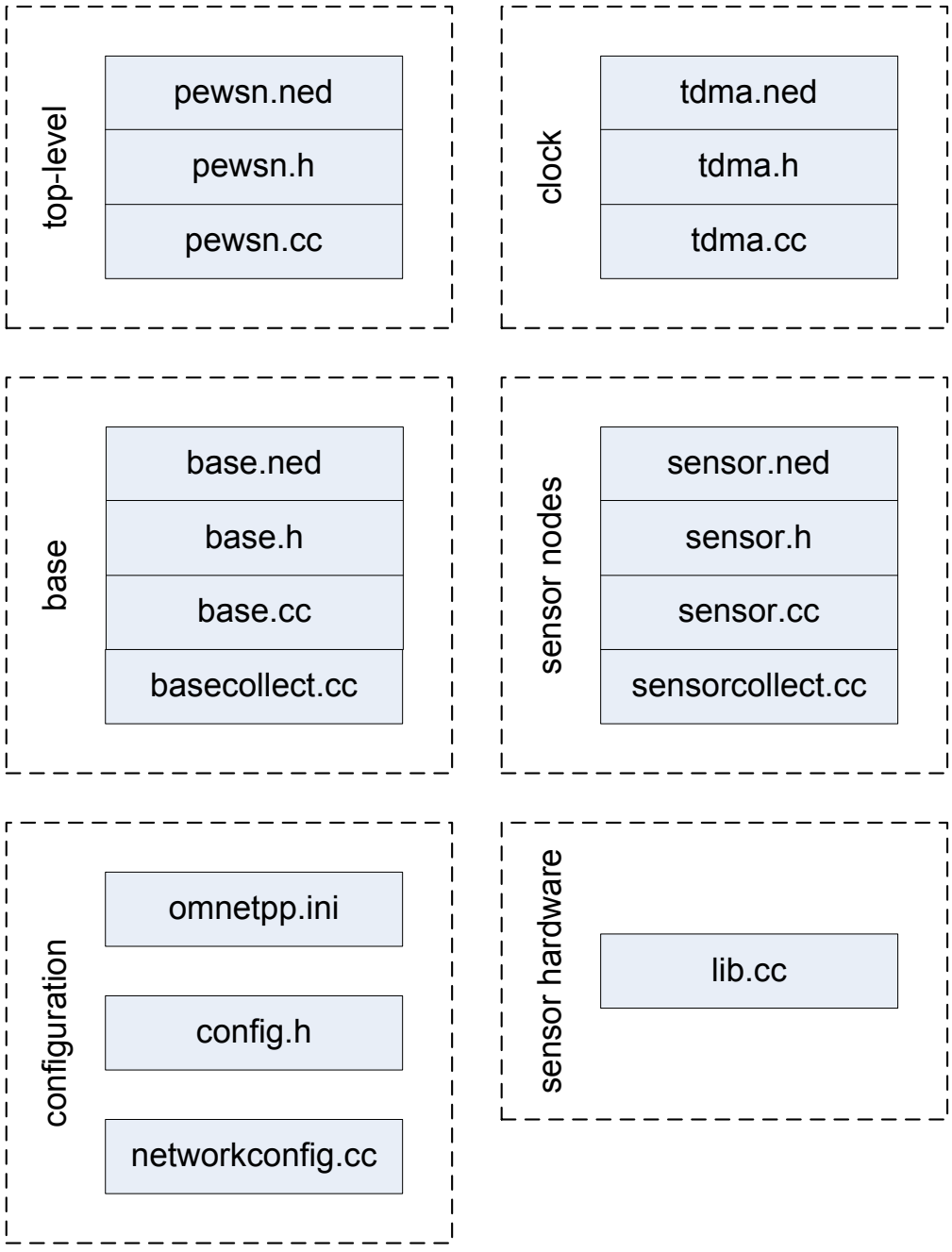


Fig. 28. PEWSN simulator block diagram

The simulator consists of the PEWSN block as the top level entity block. The block captures functions that are common in both base and sensor modules. The

design is fully object-oriented. The modules consist of the pewsn.ned file, which is an OMNET++ language native GUI file, along with pewsn.h and pewsn.cc as C++ code entities.

The next module is the base module, which captures all functionalities of the PEWSN base station. In other words, this module creates a base station that serves as the central data gathering place. Other functionalities include programming all sensors, reprogramming all sensors, gathering and analyzing data coming from all sensors, choosing initial cluster heads, and creating the network topology.

The sensor module is the main PEWSN module, because it contains all necessary sensor functionalities. Therefore, this module acts as a fully independent sensor in the field. The sensor module contains functionalities like sensing data, processing data, handling incoming and outgoing communication packets, handling power harvesting and power depletion, acting as the individual sensor or as the cluster head, and creating connections with other sensors/nodes in the network.

The TDMA module acts as the TDMA clock. Since the OMNET++ framework is event-driven, this module makes sure that TDMA slots are distributed according to the configured slot duration parameter.

The basecollect and the sensorcollect modules are debugging tools which display the data currently found in the base or in any individual sensor. This is particularly useful when concepts such as data aggregation, data filtering, or data compression are used.

The lib module is the hardware parts module. Since the PEWSN simulator uses real COTS parts performance parameters as the input in its simulation, this library helps select the right parts such as the battery, harvester, sensor, processor, and communication unit. This library makes this simulator well tuned with the real world hardware environment.

The config module is the main simulator's configuration file. This module programs the number of clusters, the number of sensors per cluster, the size of sensed data payload, the sensor's hardware parts, and the size of the overall network address space.

The networkconfig module is another configuration file. It is responsible for generating sensors' random positions in the field given field size and density constraints, generating the network cluster tree, and making sure that all appropriate variables are properly set for the graphical interface.

The omnetpp.ini file is the OMNET++ framework configuration file. This file ensures that all necessary simulation parameters are set. Examples of simulation parameters are enabling message tracing, specifying the simulation duration, specifying the data collection type (compression, filtering, aggregation etc.), enabling scalar and vector variables recordings, specifying the TDMA slot duration, and specifying the simulation time precision (s, ms, μs etc.).

The end result of each simulation is the file containing recorded scalar- and vector-based data. The file is in a format easily accessible by OMNET++ data display tools as well as by any third-party data post-processing tool. The simulator also allows for archiving data originating from multiple random runs.

The simulator works as following: At the very beginning, the C++-based code is compiled and the executable is generated. Running executable starts the application and the GUI interface. The simulation runs in event steps, with the predefined step period (in our case the step period is the TDMA slot period). The step period is a pre-compile parameter found in omnetpp.ini. There can be multiple events simultaneously executed at the specific simulation time. As the simulation progresses all variables in all sensor nodes are being updated according to PEWSN sensor node algorithm. Vector variables are also being stored in the memory for post processing manipulations. Each vector entry is accompanied with the timestamp. For example, the battery capacity vector variable in Node 1 might have entries [0 s, 1mW; 1s 0.9mW; 2s 0.8mW]. The simulator also keeps track of scalar variables. Those variables are updated as the simulation progresses, but they are recorded as a single value at the end of the simulation. During the simulation, nodes communicate among each other via messages. Messages are the primary and the only mean of nodes' communication. The simulation ends when an end simulation statement is triggered. For example, if the battery capacity on one of the nodes reaches 0 mW, the simulation will automatically end. All vector and scalar values will be collected and archived for further processing.

CHAPTER 6. PEWSN APPLICATIONS

6.1 Airport Protection using WSNs

6.1.1 Introduction

Protecting critical infrastructure such as ports, nuclear facilities, power grids and public buildings is a cornerstone of the Homeland Security Department strategy to make the country more safe. Airport protection, as a part of that strategy, has generated much research in recent years.

Fig. 29 shows a basic airport layout. We divide the area into three security zones (regions):

- Zone 1 – no trespassing area
- Zone 2 – limited access area
- Zone 3 – public space

Zone 1 is defined as an absolute no trespassing area. Public access is strictly prohibited in this area at all times. Violations could cause serious risk to public safety and security. Airport areas marked as Zone 1 must be rigorously protected. Zone 1 is usually protected via physical security systems such as fences and walls, as well as electro-mechanical systems such as surveillance cameras and intrusion detection sensors.

Zone 2 is a limited access area. This area is accessible only to airport-cleared personal such as suppliers delivering food and gas, surveillance officers working on airport's security installations, computer technicians maintaining

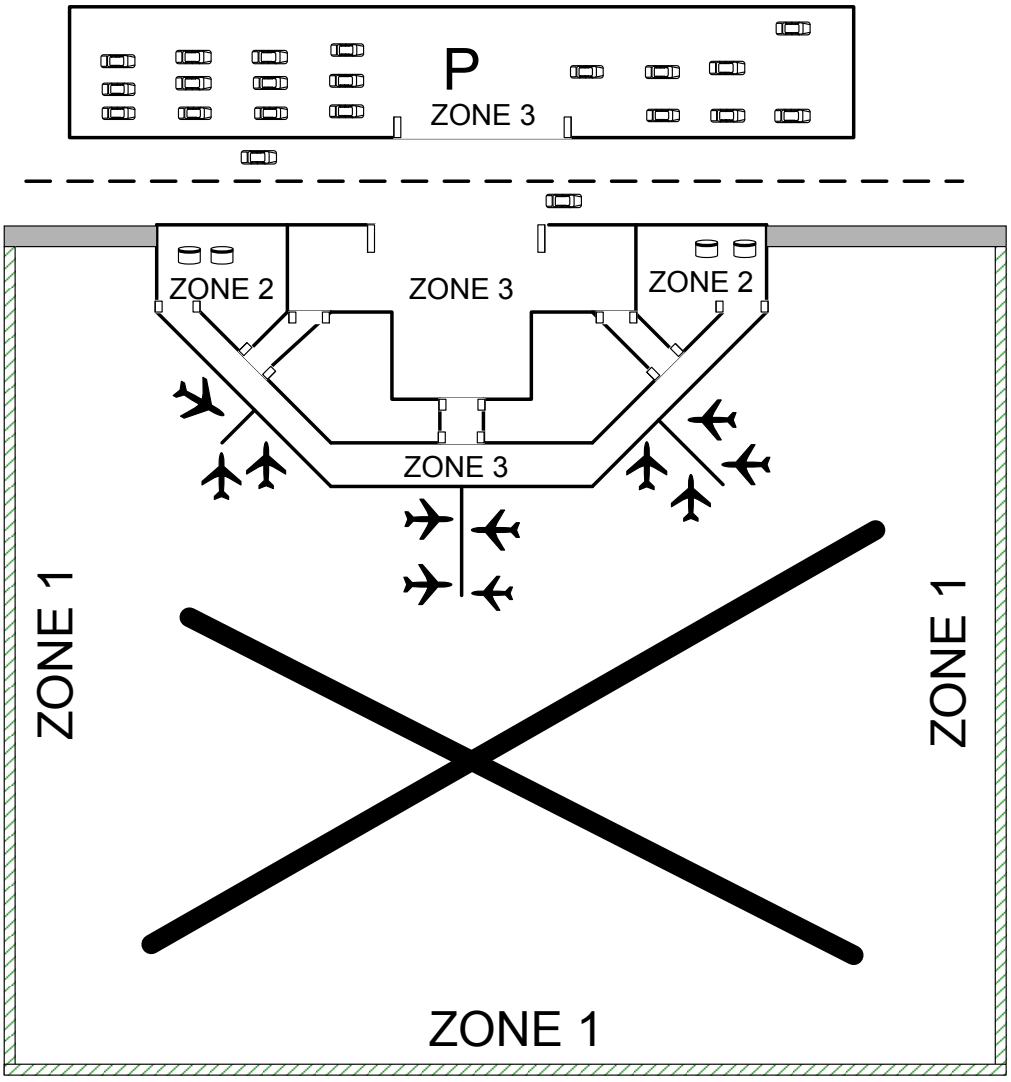


Fig. 29. Airport Layout

airport's computer systems and other airport employees. Public access is strictly prohibited, as it can create serious risk to public safety and security. The traffic in these areas is also expected to be heavily dependent on the time of day, i.e. at night, almost any motion must be carefully examined.

Zone 3 is generally accessible to the public. This area must be monitored against unexpected events that could cause mass panic and human casualties. Zone 3 must be actively monitored against any airport prohibited items such as

weapons, explosives, radioactive materials, drugs, and certain chemical and biologic agents.

In this application, we describe a way how to use PEWSN architecture for Zone 1 surveillance, i.e. open-field intrusion detection. We first present related work in the field of airport surveillance. We describe general airport security technologies and then Perimeter Intrusion Detection Systems (PIDS), focusing on the wireless sensor network (WSN)-based PIDS. Then we propose a PIDS that is based on PEWSN carrying the accelerometer as its main sensing device. At the end, we summarize our conclusions.

6.1.2 Related Work

6.1.2.1 Airport Security Overview

The Transportation Security Laboratory (TSL) of the Department of Homeland Security identifies Commerce Inspection Systems, Passenger Inspection Systems, Infrastructure Protection Systems and Conveyance Protection Systems as four key security challenging areas [77]. Airport security, as part of Infrastructure Protection Systems, requires the demonstration and evaluation of technologies necessary for intrusion detection/control, access control and building protection/control of over 5,000 US public-use airports.

Airport protection is difficult to achieve in a balanced way. Protecting the airport perimeter against intrusion, monitoring passengers, inspecting cargo, checking luggage, controlling access to limited-access (Zone 2) areas and providing guidance in case of unexpected events requires synchronization and

interoperability across different surveillance and detection technologies. Simply building a testbed of that scale presents several major challenges, all constrained by the relevance of the operational environment [78]. Schiefelbein [78] suggests an information-centric approach where all data fit a specific format that is observed, filtered and used as required by the security system.

This approach would most certainly simplify the modeling and testing of a complex airport security system. One of the proven modeling tools for Homeland Security that can be adapted for airport protection modeling is BALANCE [79]. BALANCE models the interactions between personnel, vehicles and sensors and their associated communication needs to determine if the desired level of protection is achieved. Another useful airport security modeling tool that was successfully deployed at the Los Angeles International Airport (LAX) is ARMOR [80]. ARMOR designers realized that limited security resources prevent full security coverage at all times, which allows adversaries to observe and exploit patterns in selective patrolling or monitoring. In order to avoid the vulnerability that comes with predictability, the ARMOR software randomizes schedules for patrolling, checking, or monitoring and therefore maximizes protection with available resources.

Since airports present a dynamic environment where passengers, cargo, and service providers constantly move, location-based services such as active tracking and surveillance are becoming essential airport security tools. LocOn [81] is one such tool that was successfully deployed at Faro Airport, Portugal. The tool provides airport localization, resource identification, surveillance, path

analysis, multi-tracking, safety incursions and safety infringements detection, environment actuation and collision avoidance services.

While there are various tools aiding in homeland security efforts to protect critical infrastructure such as airports, there is no unified way of providing services. Various organizations sponsored by the Homeland Security Department are trying to develop open standards for homeland security sensor networks. Lee and Reichardt [82] state that sensors and networks are the key components in building a distributed nationwide security infrastructure; they provide certain guidance for consolidating various systems and architectures. Saputra et al. [83] propose a way to combine IP-based and non- IP-based security and safety networks under the same underlying framework.

While [82] and [83] properly identify key security infrastructure elements in need of standardization and consolidation, and provide solid recommendations on how to standardize those elements, these recommendations are yet to be embraced by the industry.

6.1.2.2 Parimeter Intrusion Detection System (PIDS)

PIDS are a subset of available airport protection technologies. In [84], the author outlines the major technologies used in creating PIDS. Microphonic sensors measuring acoustic signals are one example. Since those sensors are usually mounted on fences to detect intruder attacks, characteristic frequency spectra were found for different forms of attack to the fence [85]. Properly filtering the obtained signals increases the chance of recognizing the type and the position of the attack.

Another interesting PIDS uses FAA's Airport Surface Detection System (ASDE-3) radar as its core intrusion detection technology. Barry and Czechanski [86] investigate conditions under which ASDE-3 can detect human targets. A system based on that concept is deployed at John F. Kennedy (JFK) International Airport. The system integrates ASDE-3 radar and on-site cameras to protect the airport's perimeter [87]. The deployed system first monitors the objects of interest via ASDE-3 radar. In the event of a perimeter breach, the system immediately alerts personnel and automatically moves the cameras toward the intruder's position. A similar system called Mobile RAVIN [88] was deployed at Seattle-Tacoma International Airport (SeaTac). Mobile RAVIN also uses the existing ASDE-3 radar and cameras to detect intrusion. However, both systems are also error prone, because they partially rely on human interaction with the system. The camera performance is also diminished at night.

Dibazar et al. [89] describe a "smart fence" using geophones and microphones. Properly filtered seismic/acoustic signals can detect different kinds of perimeter intrusions (climbing, kicking) because a distinct intrusion signature is recognizable in the received seismic/acoustic data.

Another very interesting solution uses optical fiber vibration sensors to detect events such as fence climbing and shaking. The system described in [90] uses the light's phase shift to calculate the strength of the event. The clockwise (CW) and counterclockwise (CCW) light beams are sent from the storage box to the terminal box. In the case of no event, the CW and CCW light return at the same time. The photo detector detects no phase shift. In an intrusion event, the

two light waves arrive at the photo detector at slightly different times, causing a phase shift that suggests movement along the covered path. The proposed system is very attractive because it not only recognizes an event, but it can also accurately detect its strength. In addition, the detection bandwidth is very wide. However, this system suffers from a few major setbacks. The need for a long fiber optical cable along the fence makes the system detectable and vulnerable (cutting a fiber cable would cause fatal system breakdown). The system is also further exposed due to the fact that it must use ground- buried power and communication cables to feed its power-hungry storage boxes. Fiber optics is also an expensive, temperature sensitive medium.

As is the case with security technologies in general, there is little standardization within PIDS. As Dewar [91] points out, the standardization of PIDS would reduce the cost of system integration, both up-front during initial installation and downstream over the life-cycle of the system. This point is further emphasized by Raytheon's architectural concept for an integrated perimeter security system [92]. The concept uses Department of Defense (DoD) Architecture Framework and is therefore compliant with Homeland Security requirements.

In summary, perimeter intrusion detection systems (PIDS) rely on diverse sensor technologies including radars, cameras, optical fiber, acoustic and seismic sensors. The network technology carrying the sensed data mostly uses wired underground communication cables.

6.1.2.3 Wireless Sensor Network (WSN) based PIDS

As mentioned before, wired network technologies are much more established in PIDS. Traditionally, the wired approach was perceived as more reliable and secure. Only some PIDS use wireless sensor networks as the data carrier.

Navin et al. [93] propose a camera-based wireless sensor network system. The promise is that the large amount of low-power distributed camera nodes can efficiently monitor the secure environment by providing different views of the scene under surveillance.

The system proposed in [94] uses GPRA and CDMA dual-mode wireless sensor networks (DM-WSN) to carry the sensed data obtained via a Perimitrax electromagnetic field based buried sensor cable [95].

Our PEWSN-based solution encompasses all of the following:

- Primary purpose is airport protection
- Primary use is perimeter intrusion detection (PIDS)
- Primary data carrier is wireless sensor network (WSN)
- Primary sensing device is a 3-axis accelerometer

6.1.3 Proposed Solution

We propose PEWSN-based network of wireless accelerometer-based sensor nodes to detect airport intrusion events such as fence climbing, kicking and cutting. Our system concentrates on an airport zone where the following assumptions and requirements exist:

- Perimeter Intrusion Detection System (PIDS) is needed (Zone 1

environment)

- The area is fenced

Fig. 30 depicts an example of a Zone 1 area.

Fig. 30. Chain-link fence commonly found in an airport's open field

Fig. 31 shows a fence containing three parallel independent network virtual lines. Sensors in the line are connected via PEWSN protocol. If one line fails, the event can still be detected through the other two network lines. Each PEWSN-based network line consists of a series of clusters. Each cluster contains 3 sensors spaced 3 feet apart. Each cluster is bound by the fence segment (the space between two consecutive posts). There are three clusters within each fence segment. Each cluster contains three sensors equating to nine sensors per fence segment.

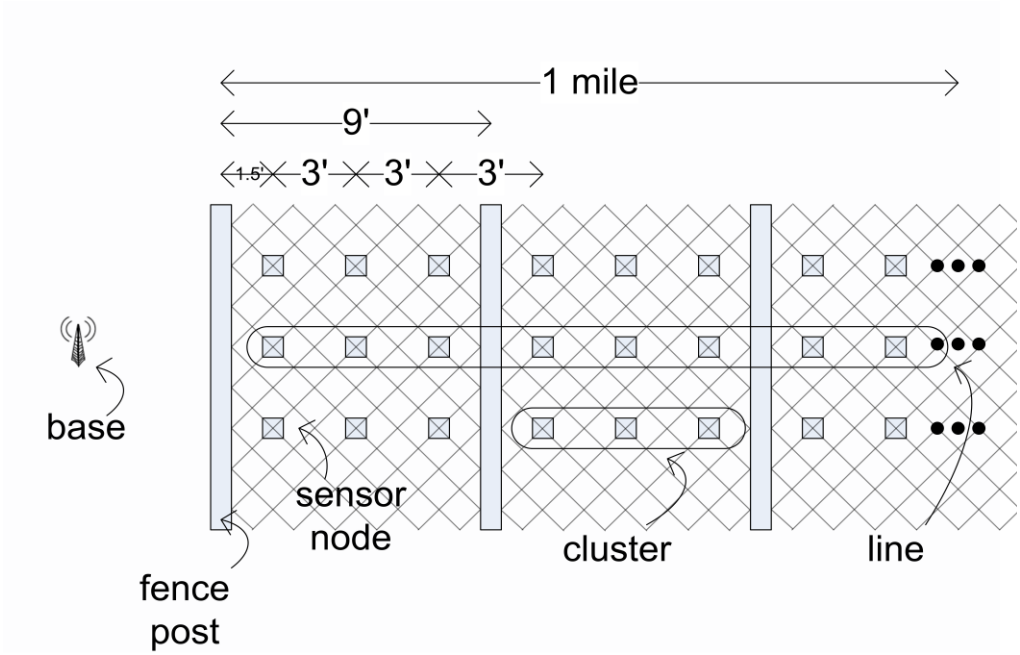


Fig. 31. Wireless Sensor Network Architecture for an Airport PIDS

When an event such as fence shaking, climbing, kicking or cutting occurs, the closest node accelerometers register the event. The data carrying the event signature are then transferred via wireless sensor network to the control base. The base then decides how to react to the event (points cameras, sends security, turns on the light, etc…).

Our security system cannot be easily disabled, unlike the interferometric gyroscope based system [90], which is exposed to the intruder disabling detection mechanism by cutting the optical light cable between the storage box and the terminal box, or by cutting either the network or the power supply cables running from the storage box underneath the surface back to the base. Since the event is most likely detected by multiple accelerometers/multiple sensor nodes, bypassing detection would require quiet destruction of a significant portion of the fence/network.

Our system also addresses reliability concerns by integrating three independent network virtual-communication lines. Since all three lines are independent (frequencies separated by 50MHz around nominal frequency of 2.4GHz), a disruption in one line will not affect communication in the other two lines. Cutting the Perimitrax electromagnetic field based buried sensor cable would cause critical network security failure in the system described in [94] and [95].

Solving the perimeter intrusion detection via a large number of low-power, low-resolution cameras, as proposed in [93], is also inferior to the accelerometer-based wireless sensor network as it partially relies on human monitoring. As [84] points out, humans are very poor monitors and detectors due to difficulty remaining alert for long periods of time.

6.1.3.1 Simulation

Before we tested the system outdoors, we ran PEWSN-based network simulation with the following goals:

- Verify the overall functionality of the wireless sensor network.
- Estimate the network lifetime assuming a 1-mile-long fence and random intrusion events.
- Estimate the amount of energy that each sensor node would need to harvest in order to extend the network lifetime infinitely (limited only by its hardware's lifetime)

In our PEWSN-based network simulator (OMNET++ framework), we also used parameters that accurately represent the parts found on a STMicroelectronics MB851 board (sensor board used in our outdoor field test and described in [96]). Our simulation environment was identical to the one depicted in Fig. 31:

- 1-mile-long fence with 587 fence segments containing 1761 clusters
- 3 individual sensors per cluster (one of them is the cluster head)
- Distance between two consecutive nodes was 3 feet
- 3 independent network lines (one above the other)
- Nodes positioned high enough above the ground to avoid significant signal fading and shadowing effects

Fig. 32 shows the results obtained from the three virtual network lines. The first line is the truth line, showing exactly when the event happened. The next three lines are the wireless network communication lines (A, B, and C). When the first event happened, all three lines (A, B, and C) registered the event with a delay. There was no disruption in the operation of any of the three lines. In the second case, line C was disabled (simulates physical attack or a network failure) and had not registered the event. The event was still recorded by the remaining two lines. In the third case, both B and C failed, but the event still came through via line A. In the final case, all three lines were disabled and the overall wireless sensor network experienced catastrophic failure. Therefore, an intruder would need to disable all three lines (this means multiple clusters (fence segments) within each line) in order to penetrate the security zone without being recognized.

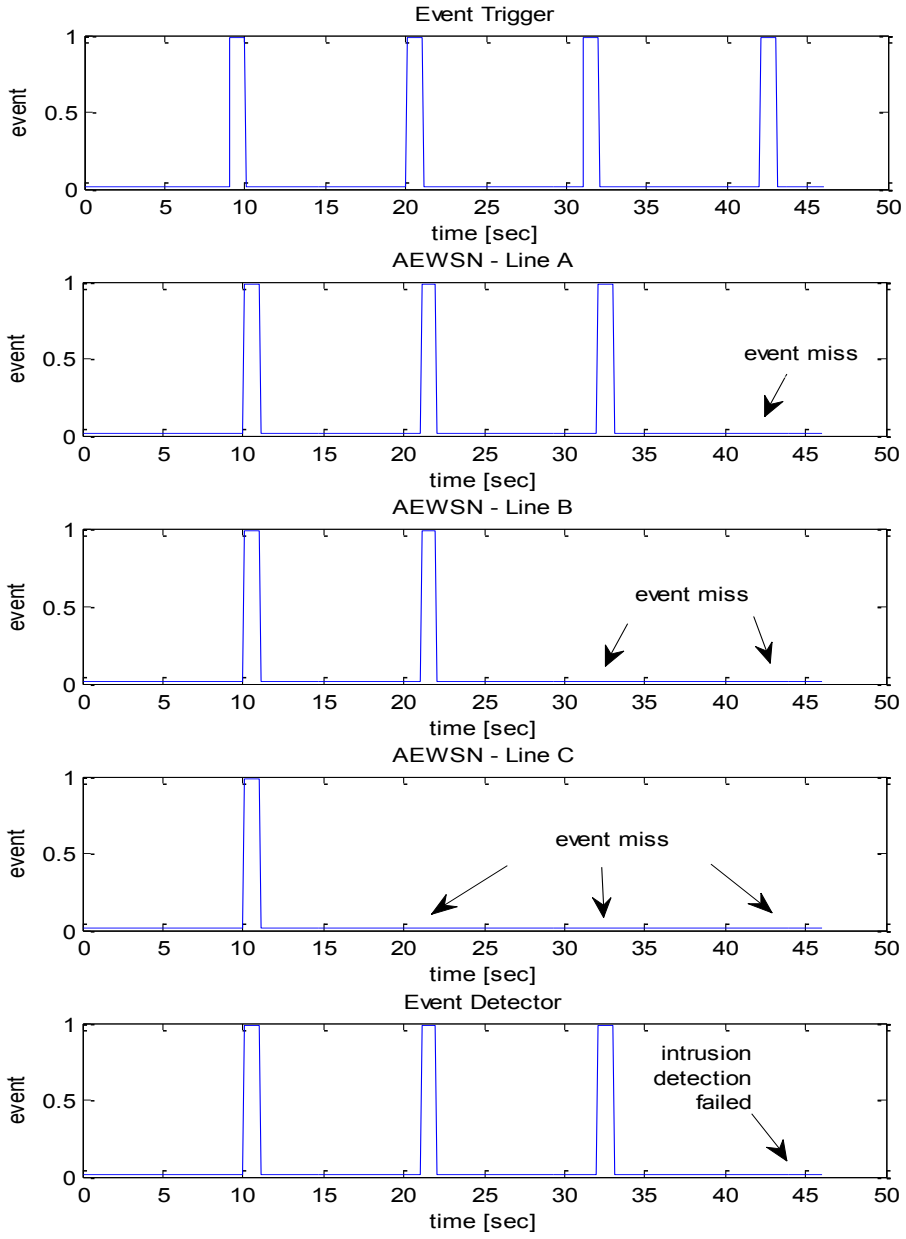


Fig. 32. Three network lines - functional simulation.

While reliability is certainly the first concern for this type of application, network lifetime is also a very important performance parameter. Network lifetime must be long enough to justify the installation and maintenance costs. Fig. 33 shows the network lifetime comparison between PEWSN and two other popular network architectures, LEACH [97] and PACT [98]. The comparison also includes a case in which there is no cluster head switching among nodes within the same cluster. It is clear that PEWSN is certainly the most enduring wireless sensor network architecture. Even though PEWSN lifetime (~33hours) is longer than the lifetime of any other WSN architecture, it is still too short for the target application. Therefore, power harvesting capability is necessary.

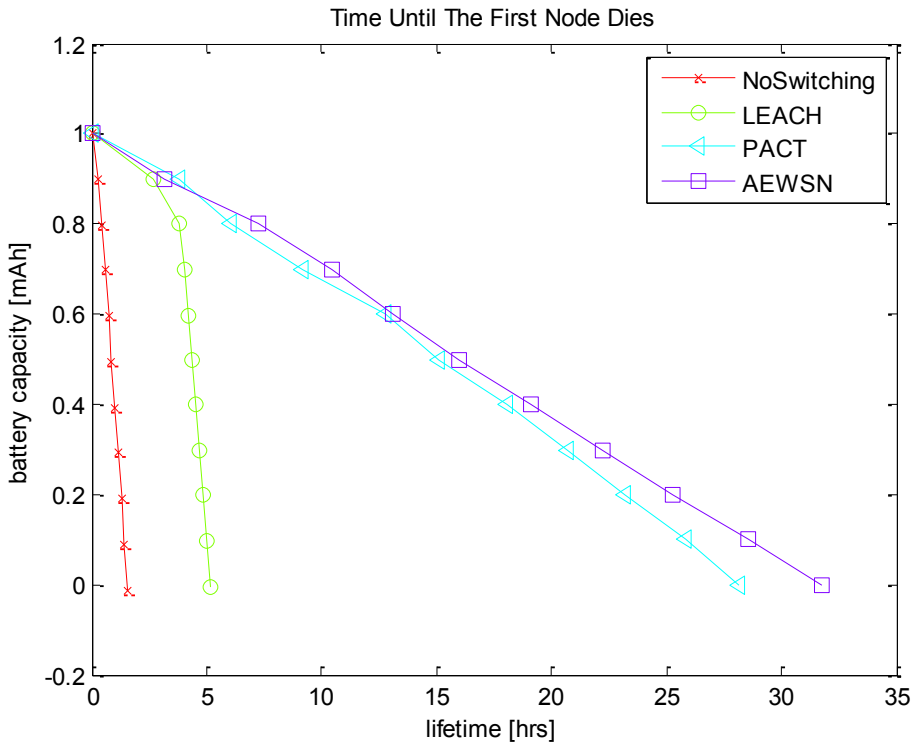


Fig. 33. Network lifetime comparison.

Fig. 34 shows the PEWSN network with all nodes equipped with a power harvesting units. The first case, the 0uA average current harvesting case, matches the results from Fig. 33. However, the network lifetime gradually increased with the stronger nodes' harvesting capabilities until it reached its power equilibrium at 50uA. Therefore, our simulation estimates that the network lifetime will not be limited by its power consumption if the sensor nodes can harvest 50uA of average current during their lifetime. Harvesting 50uA of average current is achievable with 5 to 10mm^2 commercially available solar cells.

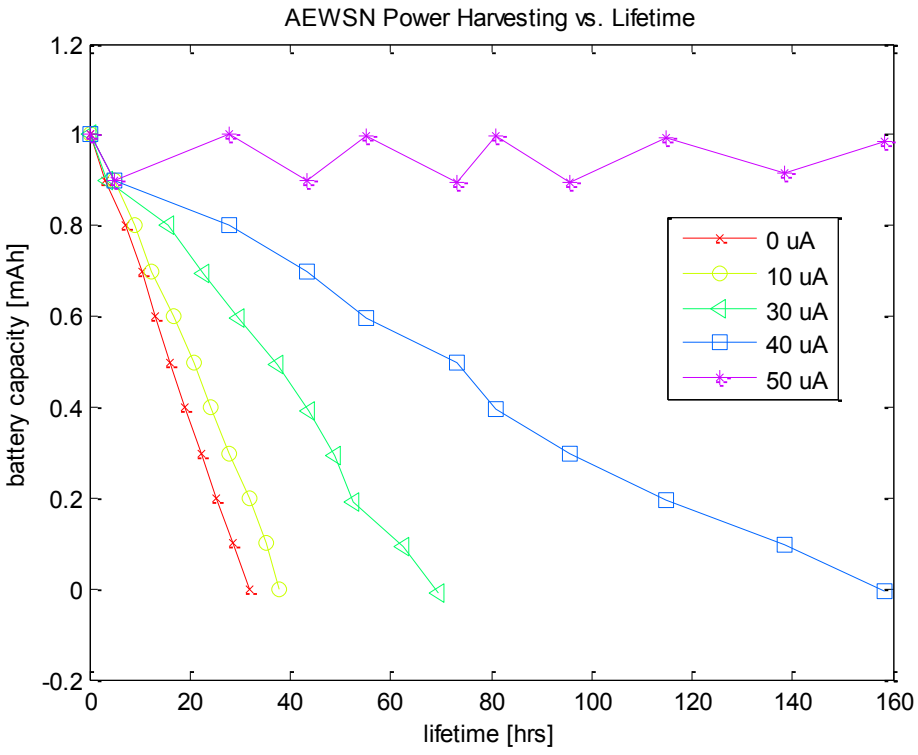


Fig. 34. PEWSN Power Harvesting vs. Lifetime.

In summary, our simulation proves that our PEWSN-based wireless sensor network system is robust, reliable and long-lasting. The next step is to verify certain aspects of this system in an open field environment where noise due to environmental factors, such as wind or ground signal fading and shadowing, might impact the overall soundness of the system.

6.1.3.2 Open Field Test

The open field testing goals were:

- Functional verification of our hardware, software and the network protocol
- Functional verification of the system under events such as fence shaking, fence up and down climbing, and fence kicking
- Outdoor environment noise levels assessment (i.e. wind)

Fig. 35 shows the open field test setup. For the test, we installed two network virtual lines (A and B), each consisting of two clusters. Each cluster consisted of three sensors. The sensors were mounted closer to the top of the fence because we experienced the effects of ground reflection (signal fading and shadowing). As per Merrill et al.[72] and Kvaksrud [71], the wireless signal range dramatically attenuates when the antennas are closer to the ground. Given that chain-link fences protecting an airport is usually about 10 feet high, moving our lowest network line 5 feet high completely eliminated the ground effect issue.

We set up the network so that only the nodes belonging to the cluster closest to the base could talk to the base. The cluster further down the fence (i.e. the next fence segment) could talk only to the next cluster. This behavior resembles our PEWSN network architecture, which we find optimal for protecting an airport.

Therefore, to achieve the desired effect in the test environment, we lowered the transmit power of each node to about -3dBm (normal mode is 0dBm or 1mW) and the RX sensitivity of each node to -90dBm. Both STM32W108 chipset [99] settings are fully programmable.

As Fig. 35 shows, shaking the fence, climbing up and down the fence, and kicking the fence produced significant spikes in the accelerometer data (both A and B lines). We also tested the overall system reliability by disrupting line B (synchronously pulled out batteries from all the nodes). The noise (non-zero accelerometer data when there is no event) shown in the test was mostly due to wind and was not observed inside when we were programming the boards. The boards were not mounted perfectly but just to support the experiment. Custom sensors, however, could easily be designed to include weatherproof housing, smaller MEMS-based sensors, improved power consumption within each node and more discrete camouflaging.

6.1.4 Conclusion

In this section, we presented our published work [100] on PEWSN-based airport Perimeter Intrusion Detection System (PIDS). The sensing part of the network is based on accelerometers capable of detecting events such as fence climbing, fence shaking and fence kicking. The proposed security system was comprehensively tested in the simulation environment and in the open field. The open field testing proved the major concepts of the system's architecture on a small scale.

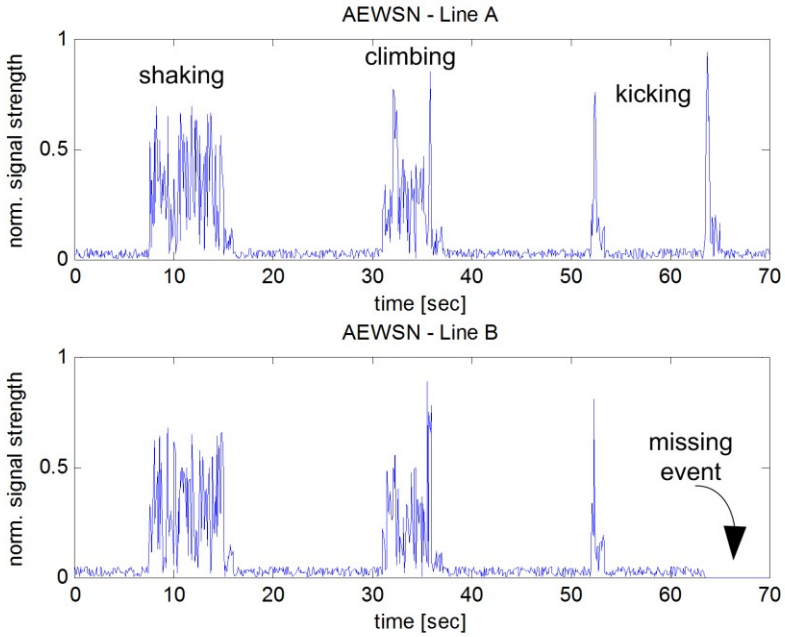


Fig. 35. Open Field Test

6.2 Underwater WSNs

6.2.1 Introduction

Terrestrial wireless sensor networks (TWSNs) such as PEWSN are an active area of research and development. TWSN fundamental properties, such as low-cost, low-power, a vast number of miniaturized nodes that collaboratively are capable of monitoring, detecting, and tracking various events and objects within a specified environment, makes them very attractive for a number of commercial, industrial, and military applications. TWSNs are capable of rapid deployment, large area coverage, low maintenance, ease of operation, and effective scalability. The number and the variety of TWSN applications are vast and are rapidly increasing. Environmental monitoring, health monitoring, habitat monitoring, seismic detection, acoustic detection, industrial process monitoring, military surveillance, terror threat detection, protection of critical infrastructure, intrusion detection, monitoring of large crowds, and guidance in case of unexpected events, are just some of the possible applications.

Fig. 36 shows a typical PEWSN architecture with clusters (C1 through C9), individual sensors (Sx[0] through Sx[2]), and the base (C0).

Underwater wireless sensor networks (UWSNs) is an emerging area of research within the overall wireless sensor network (WSN) area. With water covering 70% of the Earth, there is a need for extensive research in monitoring and exploring various aspects of ocean environment. The natural approach is to adapt currently available, and well proven terrestrial architectures, for underwater use. The architectural mapping approach, however, is very difficult to achieve.

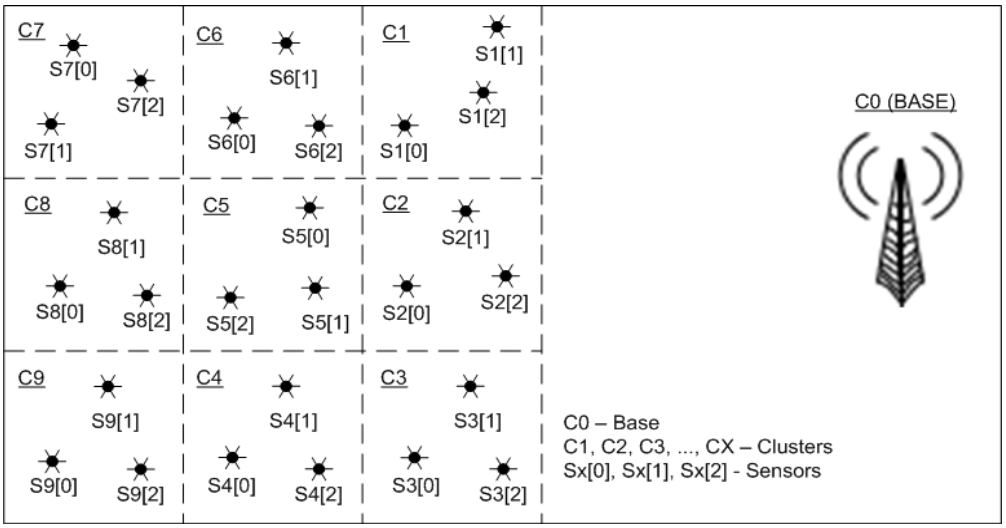


Fig. 36. A typical PEWSN-based terrestrial WSN architecture.

One of the main reasons for TWSNs and UWSNs architectural non-compatibility are extreme environmental differences in which these two networks operate. For example, TWSNs operate in an environment dominated by RF communication. RF communication is a logical choice because the propagation delay through air, as well as the environmental noise is relatively small. Yet, RF communication is not an optimal communication channel for underwater applications because of the extremely limited RF wave's propagation underwater. The acoustic communication is preferred underwater. The acoustic communication, while more reliable and robust, is bandwidth limited. Underwater acoustic rates are between 5 kb/s and 20 kb/s, which is extremely slow compared to over-the-air RF rates (in Gb/s).

Environmental differences force various technological tradeoffs needed in order to adapt WSNs to underwater situations. The technological changes drive the changes in network architectures. While UWSN architectures use a hybrid

approach (different node types, mix of static and mobile nodes, mix of surface mount and seabed mount nodes) to construct a robust networking solution, TWSNs are usually less diverse (one node type, same deployment procedure, same functionality).

In this section, we explore a vast field of UWSN applications and their correlation with terrestrial PEWSN-based WSNs. We focus on challenges faced by UWSN-based applications. We also show how those challenges drive UWSN architectural and design decisions. We also explore the use of UWSNs in commercial, industrial, and military applications given the challenges. At the end we give an overview of currently available UWSN architectures and we compare them with PEWSN architecture.

6.2.2 UWSN Applications

The number of UWSN-based applications is constantly increasing. This trend is partially due to advances in areas such as electronics, micro-electro-mechanical systems (MEMS), underwater communications, underwater sensors, and equipment waterproofing. The trend is also due to the fact that oceans, while covering the vast areas of Earth (70% of the Earth's surface with an average depth of 4km [101]), are still considered largely unexplored and insufficiently known.

A great number of UWSN applications can be classified as monitoring applications. Water quality analysis, pollution monitoring (chemical, biological and nuclear), monitoring of ocean currents, tracking of fishes or micro-organisms, pressure and temperature measurements, as well as conductivity and turbidity

analysis, are all examples of environmental monitoring [102]. Monitoring underwater structures such as oil platforms, oil and gas pipes, buried communication high-speed cables and other equipment monitoring can all be achieved using underwater WSNs.

Seismic monitoring is another very popular application, because most of the gas and oil reserves are underwater. The frequent 3-D and 4-D seismic monitoring is required for oil extraction [103]. In 3-D and 4-D seismic surveys, close coordination among sensors is required because those surveys rely on mapping the acquired data with the exact location of data-providing sensors.

In addition to monitoring applications, UWSNs are used for assisted navigation and control. Autonomous underwater vehicles (AUVs), remotely operated vehicles (ROVs), and underwater unmanned vehicles (UUVs) use UWSNs as location reference points. For example, sensors anchored at the ocean's bottom at known locations can provide the location reference, as well as valuable water characteristics information, to passing AUVs, ROVs, and UUVs. Underwater sensors can also provide ships with valuable information about where to anchor or trespass shallow corridors. Communication with divers is another UWSN area of usage.

Unmanned underwater exploration, as well as object localization, also benefit from an UWSN infrastructure. AUVs, ROVs, UUVs were critical in the 1985 discovery of the Titanic by the Woods Hole Oceanographic Institution. A number of successful lost treasure discoveries were made with the help of UWSNs.

UWSNs are extensively used in military and homeland security applications [104]. Underwater sensors anchored to the ocean floor are used as a powerful surveillance tool. Underwater warfare, submarine navigation, submarine attacks, and submarine hunting can be initiated and controlled via UWSNs. Future UWSNs applications also envision using unmanned submarines and UUVs for active attack purposes. UWSNs are also used in securing critical infrastructure such as port facilities, ships and submarines anchored in ports. Further applications focus on enemy targeting and intrusion detection.

Mine reconnaissance and de-mining activities are aided with UWSNs. Minefields are historically very difficult to detect early enough to avoid a disaster. AUVs, ROVs, and UUVs can be an effective mine discovery and deactivation tool.

Disaster prevention and disaster recovery are areas of active UWSN research. As Akyildiz et al. [102] points out, the UWSNs measuring seismic activity from remote locations can provide tsunami warnings to coastal areas. Various animal and coral activities properly picked up via UWSNs can serve as an early storm warning system. Hurricane disaster recovery can also be aided via detection by time-critical UWSNs. Disaster recoveries such as the ones caused by marine incidents, such as chemical pollutions and oil spills, can be aided by UWSNs.

UWSNs can also be used for corrosion detection in underwater oil and gas pipes, as well as corrosion detection of oil platform structures. Corrosion, a slow but steady process, is difficult to detect thousands of meters underwater.

Corrosion detection sensors networked wirelessly into UWSNs can be a valuable tool for corrosion damage prevention.

Using UWSNs for exploring and harvesting natural undersea resources such as minerals, corals and coral reefs, fisheries, and rare metals, is becoming more attractive with the advancements in UWSNs. Imaging sensors applications can be used to visualize, classify, count, or simply observe various underwater species.

Marine incident investigations can use UWSN as a powerful tool to unwind the sequence of events leading to the incidents. Incidents could also be prevented by creating early-warning UWSN systems.

6.2.3 UWSN Challenges

UWSNs face a number of technological challenges. UWSN challenges are inherently different than TWSN challenges.

The fabrication, deployment, maintenance and recovery costs are very high compared to TWSNs. Typically a UWSN node costs around 10K, whereas a comparative terrestrial node costs only $100. Fabricating a rugged pressure housing costs $3k, whereas a simple underwater connector costs $100 [101]. Oceanographic research ships cost between 5k and 25k per day.

Power consumption and power harvesting are challenging in UWSNs. Since underwater sensors are residing deep in the water, power re-charging is logistically difficult. Therefore, UWSN node design must take power scarcity into

consideration. This is especially true for AUVs, ROVs, and UUVs, which need additional propulsion power. As Partan et al. [101] note, the non-propulsion power (sensors, communication, electronics) is typically 30W, with propulsion power ranging from 15W to 110W. Power harvesting is problematic because the common power-rich energy sources, such as solar, are not available underwater. New power-efficient robust protocols such as one described by Xie et al. [105] must be developed for underwater applications.

The WSN localization issue is amplified underwater. Placing/anchoring a node at exact geo-locations, as well as preventing the node from occasional displacements over time, is a difficult task. Most of the TWSN localization techniques use time of arrival (ToA) or received signal strength indication (RSSI) to estimate the exact node position. However, since UWSN nodes are sparsely deployed, localization accuracy using those techniques greatly suffers. In addition, some other localization techniques, such as techniques using GPS, do not work underwater. Chandrasekhar et al. [106] classify the localization schemes into range-based and range-free schemes (schemes that do not use range or bearing information). They also compare different localization schemes, and they outline the challenges facing each scheme.

Time-synchronization is an issue, even in RF-based TWSNs. Elson et al. [107] proposed a synchronization solution that generally performs well in TWSNs, because the RF propagation delay is negligible. Unlike TWSNs, most UWSNs use acoustics for communication. Underwater sound propagation speed is 1500 m/s compared to $3x10^8$ m/s for RF terrestrial signal. The five-orders-of-magnitude

difference in propagation delay makes TWSN's time-synchronization techniques impractical. Also, clock drifting between nodes can create synchronization issues. Heidemann et al. [103] calculate that a drift rate of 50ppm can create a clock difference as big as 130s after 30 days. Syed and Heidemann [108] suggest the use of a clock synchronization-reducing protocol called Network Time Protocol (NTP). TWSN's preferred Time Division Multiple Access (TDMA) channel access method, which relies on precise clock-synchronization among nodes, is replaced with Code Division Multiple Access (CDMA) in UWSN. Unlike TDMA, CDMA can tolerate occasional loss of synchronization and various clock drifts and jitters.

UWSN lifetime is an area of extensive research. UWSNs suffer from a sensor's fouling and corrosion [109]. Electronics components, such as the battery, tend to degrade faster under extremely low temperatures such as the one found in deep underwater. As a consequence, the USWN lifetime is much shorter than the lifetime of a comparable TWSN. A shorter lifetime increases the replacement and maintenance costs. An example is an oil exploration survey that is run intermittently every 1 to 3 years and then only for a short survey time. The temporal nature of UWSN narrows the field of potential underwater applications that can greatly benefit from UWSN technology.

Communication among UWSNs is probably the biggest challenge facing UWSNs. Akyildiz et al. [109] point out that path loss (attenuation and geometric spreading), noise (man-made and ambient), multi-path, high propagation delays, and Doppler spread, can significantly disrupt or degrade the underwater

communication channel. Another problem is that standard acoustic transducers cannot simultaneously transmit and receive [101]. Underwater network communications are therefore always half-duplex. In addition, transmit power is about 100 times more than receive power, making the communication channel asymmetric. Asymmetry, on the other hand, forces significant UWSN architectural changes and the shift from the traditional data sense-and-send approach to a rather more complex store-and-forward approach. High propagation delays, and low data rates also pose a challenge for carrier-sense based transmission schemes.

Underwater data collection, storage, and retrieval are a challenge. Long and varying propagation delays greatly affect the underwater, often multi-hop, data transfers. UWSN nodes often require more memory for data caching in order to offset the intermittent nature of underwater channels, as well as to efficiently implement the data store-and-forward-based architectures. Hardware (including storage and memory), and software extreme reliability requirements, create additional technological challenges. The control of vast numbers of sensors, especially the mobile ones, also contribute to the overall problem of data transfer efficiency.

Repair and replacement of defective nodes, as well as failure detection in underwater systems, is challenging and costly. Individual nodes might fail, either due to hardware malfunction or simply by reaching their end of life. To identify, recover, and replace the failing node causes the network's full coverage interruption because the expensive underwater nodes rarely have a backup

covering the exact same area. In addition, the hardware rarely experiences a total breakdown. A more common case is an intermittent hardware malfunction that becomes more frequent over time, until a total breakdown is experienced. The intermittent, non-fatal hardware problems are more difficult to diagnose and repair.

The reliability and robustness of the underwater communication link is greatly affected by the environment. Variations in pressure, salinity, ocean currents, marine life, motion-induced Doppler effects, and man-made noise can create significant variations in link reliability. High bit error rates create a challenge in designing low-power error correction coding schemes.

Spatial data correlation among UWSN nodes is almost nonexistent. UWSN nodes are sparsely deployed because a large population within a small area can cause conflicts with throughput and navigation [103]. On the other hand, spatially correlated data can be very helpful in increasing the reliability of the event-to-sink data path. For example, Ozgur and Akyildiz [110] propose an event-to-sink reliable transport (ESRT) protocol that heavily relies on spatial correlation to achieve reliability with the least amount of energy.

Real-time data sampling and transfer imposes several challenges. Slow intermittent channels, long and variable propagation delays, asymmetric transmission, and high bit error rates (BERs) force any underwater network to be more store-and-forward oriented with unbounded network latency and variable throughput.

The capacity of underwater communication channels is limited. Kong et al. [111] report that the underwater *range x rate* product can barely exceed 40 km-kbps. Consequently, a rate of 5kbps can be achieved only on communication links 8 km long and shorter. Partan et al. report that the maximum data rate in shallow water is 5 kbits/s at the range of 2 km, but it can drop to as low as 80 bits/s.

UWSNs security is a challenge that is often neglected by network architects. The security is difficult to achieve because it requires additional hardware and software layers that often translate into additional product cost. UWSN node's power budget is also very dependent on the transmission packet lengths; encrypted messages are longer and require more power to transmit. Data privacy, authentication and keying require additional computation capabilities [112]. Architectures integrating floating buoys are also vulnerable to weather conditions, tampering, and pilfering [102].

6.2.4 UWSN Architecture

UWSN architectures can be classified in various ways. One classification discriminates between static, semi-mobile, and mobile architectures [104]. Another popular UWSN classification method is to divide UWSNs into two-dimensional (cover ocean floor) and three-dimensional (includes depth as a dimension) [102]. UWSN can also be single-hop, multi-hop, or hybrid (single-hop individual sensors, multi-hop clusters). Architectures can be grouped into short-term, time-critical applications, and long-term, non-time-critical applications [113]. RF, optical, and acoustic wave based architectures are another way to look

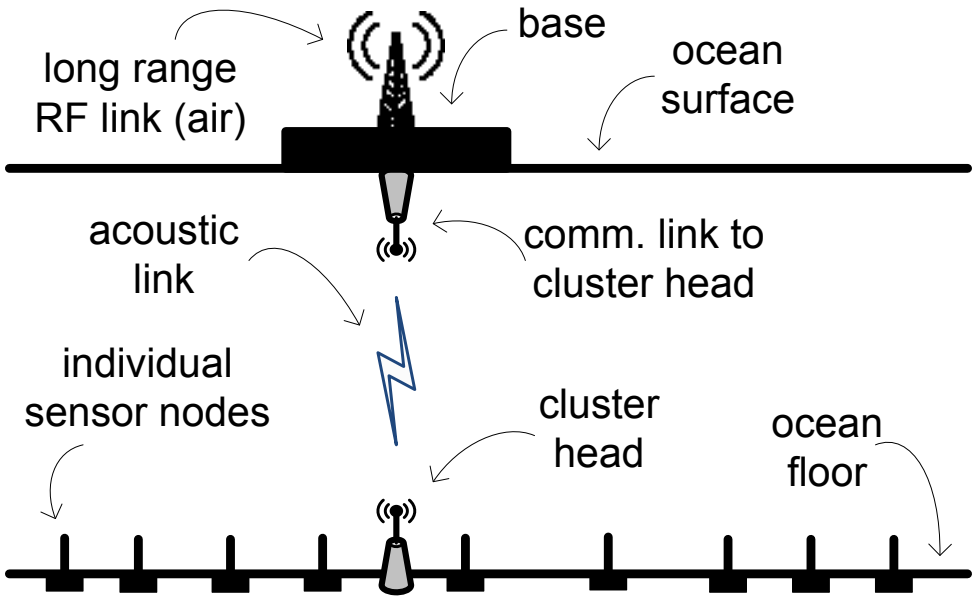


Fig. 37. 2D UWSN

at the available UWSNs. Pompili et al. [114] classifies them into delay-sensitive and delay-insensitive applications.

Fig. 37 shows the most common UWSN architecture. The individual nodes are anchored at the ocean floor. They are usually smaller in size, battery operated, and they mostly transmit data via acoustic modems. The cluster heads are also anchored to the ocean floor. In addition to having acoustic modems, cluster heads are equipped with long-range vertical-direction modems, allowing them to communicate with a base station located at the ocean surface. The long-range modems can be acoustical, optical, or even RF for shallow waters. Acoustic modems are most widely used. Cluster heads communicate via horizontal acoustic modes with all other individual nodes within the cluster. The data transfer from

node to cluster head can be single-hop (each node communicated to the cluster head directly) or multi-hop. The multi-hop approach is generally more power-efficient, because the signals have to travel shorter distances between two nodes. The network maintenance and configuration tasks are more complex in the multi-hop case.

Unlike PEWSN architecture, the hardware of the cluster head node is different from all other nodes, because it has additional functionalities such as a direct communication link with the ocean surface. Therefore, a popular PEWSN's cluster head switching feature (which increases the overall network lifetime by efficiently distributing the power consumption among nodes) cannot be utilized in UWSNs. Also, the cluster head is potentially the most security-vulnerable component in UWSNs military applications, because it is a single point of failure node.

Fig. 38 shows an alternative 3D UWSN architecture. 3D architecture can use the same nodes as those used in 2D UWSN architecture, although the nodes are anchored at different heights from the ocean floor. The buoyancy-controlled node positioning is achieved via a controllable tether anchored at the ocean floor. 3D architecture can have all nodes directly communicate to the surface base or can have only cluster heads communicate directly to the base. In the former case, all nodes are of the same type, but communication might be more energy intensive than that of the cluster head approach. The cluster head approach requires only the cluster head to carry a long-range communication modem. On the other hand, the clustered approach is vulnerable to single point of failure.

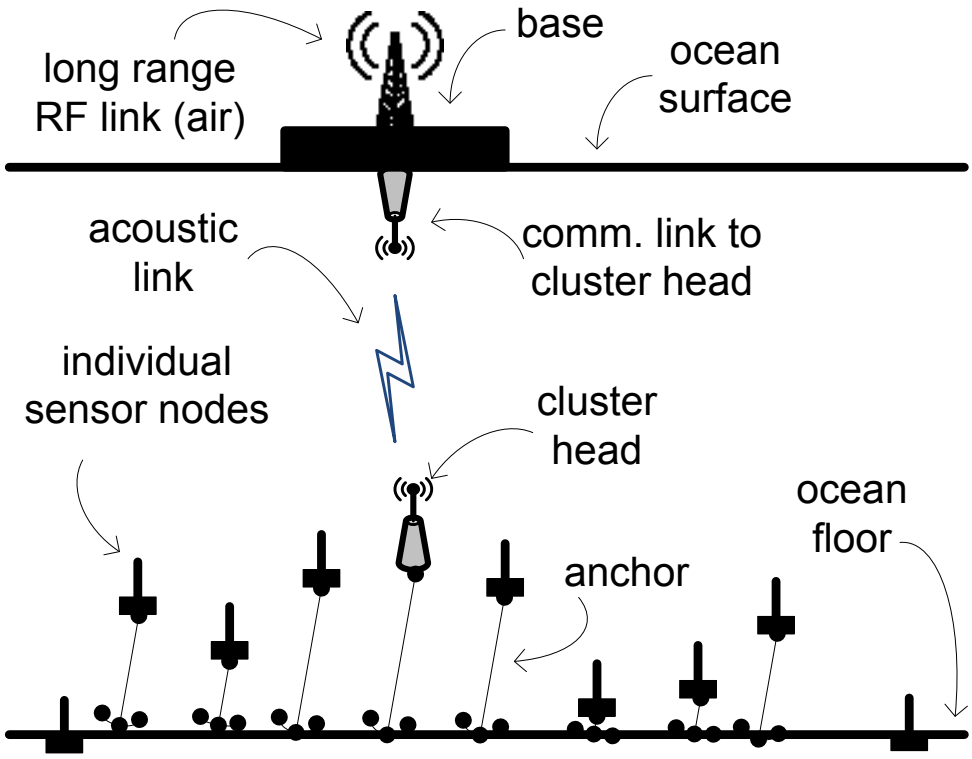


Fig. 38. 3D UWSN

Military applications are extremely sensitive to single point of failure hardware components and there they might not prefer this type of architecture. On the other hand, the same architecture might introduce multiple cluster heads within the each cluster for redundancy purposes.

While 3D architecture provides more complete images of a surveyed area, the challenge is to position all nodes in a structure that can ensure uninterrupted communication and full area coverage at all times. This is difficult to achieve because ocean currents, animals, passing ships and submarines might destroy some of the placed nodes, which can ultimately cause communication breakdowns.

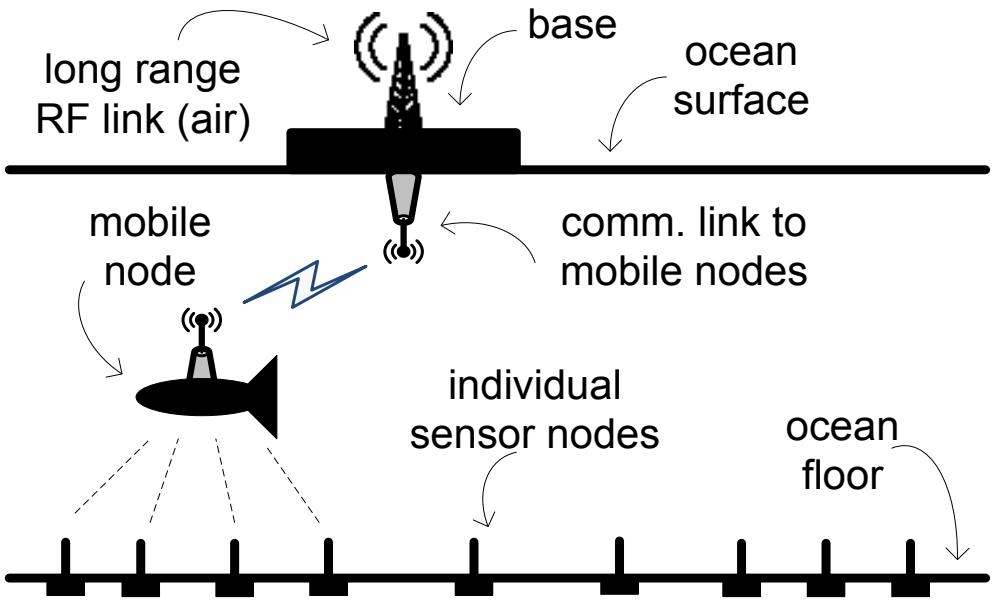


Fig. 39. Mobile UWSN

The next architecture uses AUVs, ROVs, and UUVs as network nodes. Fig. 39 shows an example of the architecture. The feature that stands out in this architecture is the mobility of nodes. The mobility of nodes allows for easier network reconfiguration and adjustment, but that mobility causes an increase in network control complexity. In addition, mobile networks tend to consume more power, because they consume extra power due to propulsion. The power issue is somewhat offset by the use of low-speed gliders and drifters. Mobile network elements, however, tend to be less reliable, and their lifetime is shorter.

Both static and mobile UWSN architectures have their advantages and disadvantages. In order to emphasize the advantages of both architectures, a new, hybrid architecture is needed. Vasilescu et al. [115] propose a new, hybrid UWSN architecture that includes a mix of static and mobile nodes networked together so

that data can be transferred efficiently from the ocean floor sensors to the ocean surface. The solution uses a combination of acoustic and optical communication. The low speed acoustic communication would mostly be used for data broadcasts, network health and welfare diagnostics, and overall network maintenance. The optical communication would be used for high data rate point-to-point communication. The point-to-point optical data communication is achieved via mobile nodes (AUVs, ROVs, and UUVs) traversing over the static field of nodes anchored to the ocean floor.

6.2.5 Conclusion

In this section, we present our published work on UWSNs [116]. We introduce UWSNs as a means of monitoring, exploring, and tracking marine life. We present an overview of commercial, industrial and military UWSN applications. Applications such as pollution monitoring, monitoring of ocean currents, tracking of fishes and micro-organisms, oil exploration and extraction, and assisted navigation represent already deployed and functional solutions. We also introduce a number of challenges facing the development and deployment of such applications. The fabrication, deployment, maintenance and recovery costs, power consumption and power harvesting, nodes localization and time-synchronization, underwater communication, data collection, storage and retrieval, repair and replacement of defective nodes are just some to mention. We make the case that one-to-one mapping between PEWSN and UWSNs is not practical. At the end we describe a number of novel, practical UWSN architectures such as 2D UWSN, 3D UWSN, and mobile UWSN.

6.3 Polysomnography

6.3.1 Introduction

Sleep-disordered breathing can negatively reflect on human health. The most prominent disorders, sleep apnea and hypopnea, are characterized by abnormal pauses in breathing during sleep. Apnea is a complete cession of airflow for at least 10 seconds followed by an arousal (shifts in brain wave activity); hypopnea is a 50% decrease in airflow for at least 10 seconds, followed by an arousal.

Mooe et al. [117], Peker et al. [118], Reishtein [119], and Dart et al. [120] associate sleep-disordered breathing with coronary artery disease. Bagai [121], Dyken and Im [122], Kaneko et al. [123], Parra et al. [124], and Wessendorf et al. [125] associate obstructive sleep apnea with stroke. Lattimore et al. [126] conclude that prolonged sleep-disordered breathing can also cause cardiac failure. Further examples of sleep-disordered breathing issues are seen in psychiatric disorders such as depression [127].

In order to diagnose and treat sleep-disordered breathing, an extensive diagnostic procedure called polysomnography is applied. Polysomnography (sleep study) uses multiple sensors attached to a patient to monitor the patient's body functions such as brain activity (EEG), eye movements (EOG), muscle activity (EMG), and heart rhythm (ECG) during sleep. The obtained test result, polysomnogram (PSG), is then used to diagnose sleep disorders such as sleep apnea, parasomnias, REM behavior disorder, narcolepsy and others.

A traditional polysomnogram requires at least 22 wire attachments to the

Fig. 40**.** Patient under traditional sleep-study test (polysomnogram).

patient [128]. Fig. 40 shows the attachments on a patient.

The traditional method is very uncomfortable for the patient. For example, each time a patient needs to use the restroom, the entire setup needs to be unhooked, then, a short time later, to be hooked back on. In addition, patients need to be tied to the bed in order to prevent unhooking the cables while unintentionally changing the position in sleep. Some rapid patient's movements in sleep can cause damage to the setup. The traditional setup is also expensive since it requires at least one attending polysomnogram technician per patient. A technician is responsible for monitoring live data and making sure all sensors are properly attached during the study (test duration is usually 1 night).

In this section, we propose a wireless polysomnogram based on PEWSN architecture. The wireless polysomnogram uses a network of wireless sensors to

communicate sensed data to the central data gathering point (base). The solution does not use any wires, making the test comfortable for the patient. The absence of wires also decreases the risk of accidental test interruption due to an unhooked wire. Leaving the bed can now be unattended, creating an option to monitor multiple patients by a single attendant, and ultimately saving the overall procedure cost.

We first present related work in the field of polysomnography. Then we describe the currently available options including some quasi-wireless solutions. Finally, we propose a truly wireless polysomnogram (TWPSG) that is fully based on PEWSN architecture. We also describe TWPSG's underlying technology along with other technical aspects and operational principals.

6.3.2 Related Work

6.3.2.1 Wired Polysomnography

A standard wired polysomnography procedure starts with "wiring up" a patient; putting all the required electrodes (sensors) and channels in place. Sensors are positioned in such a way to measure airflow, chin muscle tone, leg movements, eye movements, heart rate and rhythm, oxygen saturation, and chest and upper abdominal wall movements. Typical sensors include pressure transducers (airflow, nasal pressure), piezoelectric bands (respiratory effort), a tracheal microphone (snoring), and accelerometers (body movements). All sensors are connected via wires to a central processing unit. The central processing unit, usually a custom-made computer, receives analog sensor signals,

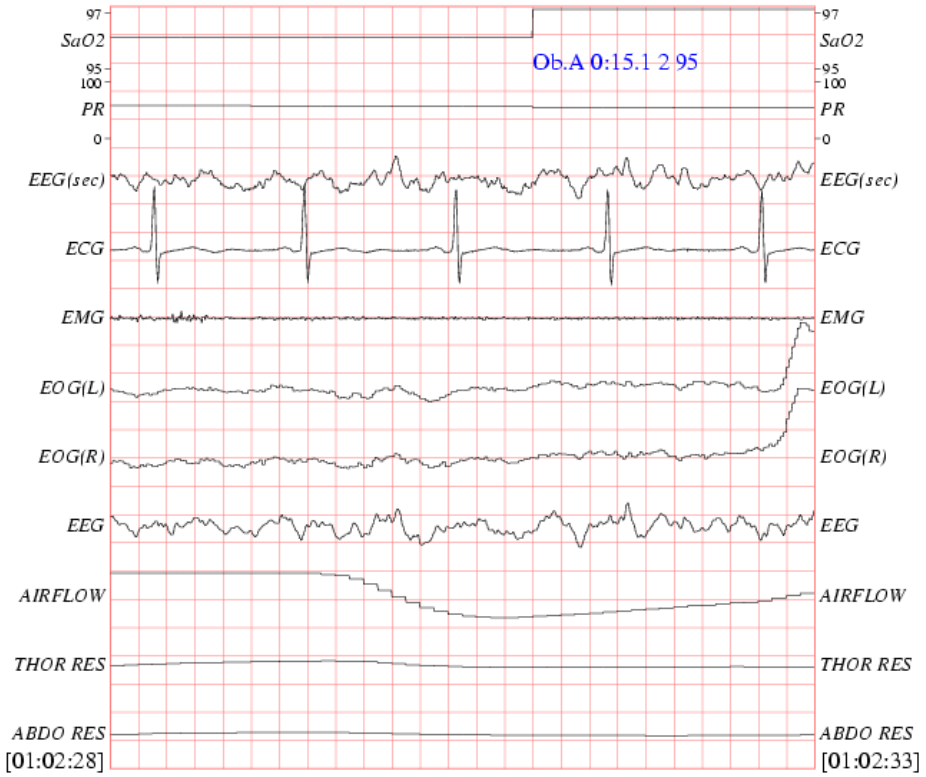


Fig. 41. Polysomnogram data output

digitalizes them, and processes them via post-processing scripts. The final test output displaying all channels with corresponding sensor data is depicted in Fig. 41. While the traditional wired PSG can successfully diagnose sleep-disordered breathing, the test is far from perfect.

The data obtained via body-attached sensors is analog, and it has to travel long distances via attached wires to reach the data digitalizing box. The analog data is prone to noise that reflects as a measuring error on the final results. In addition, the high speed analog signals traveling through adjacent wires can create a “coupling” issue, which reflects on measurement accuracy.

In addition to electric issues with the wired polysomnogram, there are also

a number of mechanical issues. A wired person attached to a data acquisition system should preferably not move during the test – a difficult requirement for a sleeping patient. Any movement can cause the momentum in the moving wires, which can reflect in the sensor's displacement. Mechanical displacement of wires, sensors, or any other part of the acquisition system almost certainly will cause either a full or a partial invalidation of the test. Therefore, a PSG trained technician is required in the room at all times during the test, adding to already significant test cost.

Lastly, a patient undergoing the test might be forced to leave the setup to fulfill his/her basic needs. Leaving the setup creates the necessity to remove all the wires from a patient, and then, shortly thereafter, to place them on the patient again. Wired polysomnography, while non-invasive, is not a comfortable procedure.

6.3.2.2 Quasi-wireless Polysomnography

Quasi-wireless polysomnography is a method in which some of the wires within the system are removed. There are a number of commercially available PSGs that consist of sensors, a patient unit, and a computer unit. Fig. 42 depicts one such PSG.

Just as in the traditional wired PSGs, sensors are still connected via wires to a patient unit. The patient unit performs digitalization of data and eventually sends that digital data, via a wireless sensor link, to a computer unit. The computer unit is responsible for data interpretation and data storage for further

Fig. 42**.** Quasi-wireless PSG

analysis.

The described system greatly increases the comfort of the patient. The patient is also not tied to the bed for the entire duration of the test. The test itself can now be entirely performed from home, significantly lowering the procedure cost. The system, in industry also branded as wireless, still does not solve the major electrical underlining issues. Quasi-wireless PSG still has wires carrying analog signals from the sensors to the patient unit. While the wires are shorter than in traditional fully wired PSGs, they are still prone to the measurement errors found in wired PSG.

Farney et al. [129] present an alternative PSG architecture (Fig. 43). In this architecture, the improvement is in the ability of the main processing unit to communicate wirelessly with the data interpreting PSG technician. In other words, the patient's results can be remotely monitored from another room or

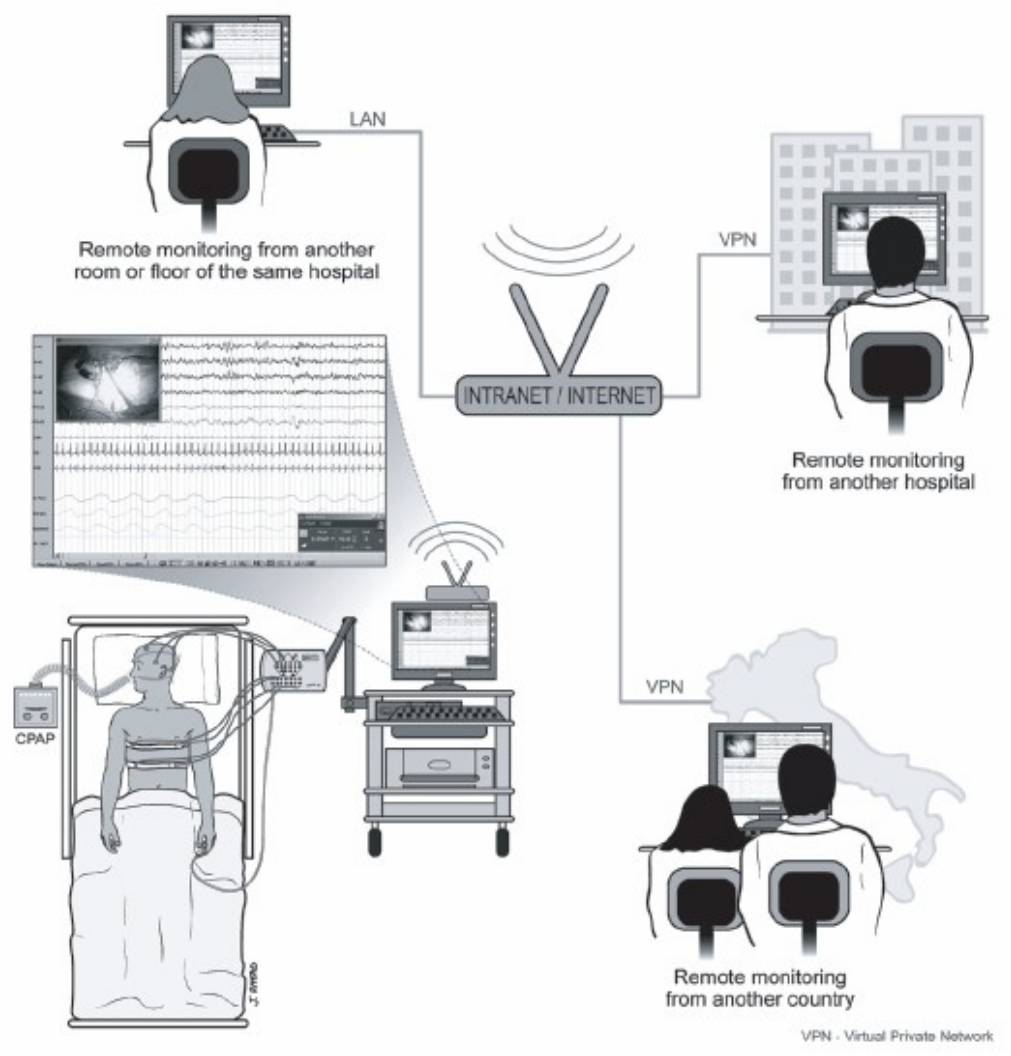


Fig. 43. Farney et al. wireless polysomnography architecture

floor, from another hospital, or even from another country. Although the proposed architecture lowers the cost of the overall polysomnography procedure, it still does not solve the issue of making the solution end-to-end wireless. The patient is still wired up via cables carrying analog signals to a main processing unit. Leaving the bed or changing the sleep position during the test might significantly distort test results. The proposed solution also does not adequately address the patient's comfort.

6.3.3 Proposed Solution

6.3.3.1 Architecture

Fig. 44 shows the proposed Truly Wireless Polysomnography (TWPSG) architecture that is fully based on PEWSN.

PEWSN-based TWPSG uses tiny sensor nodes consisting of:

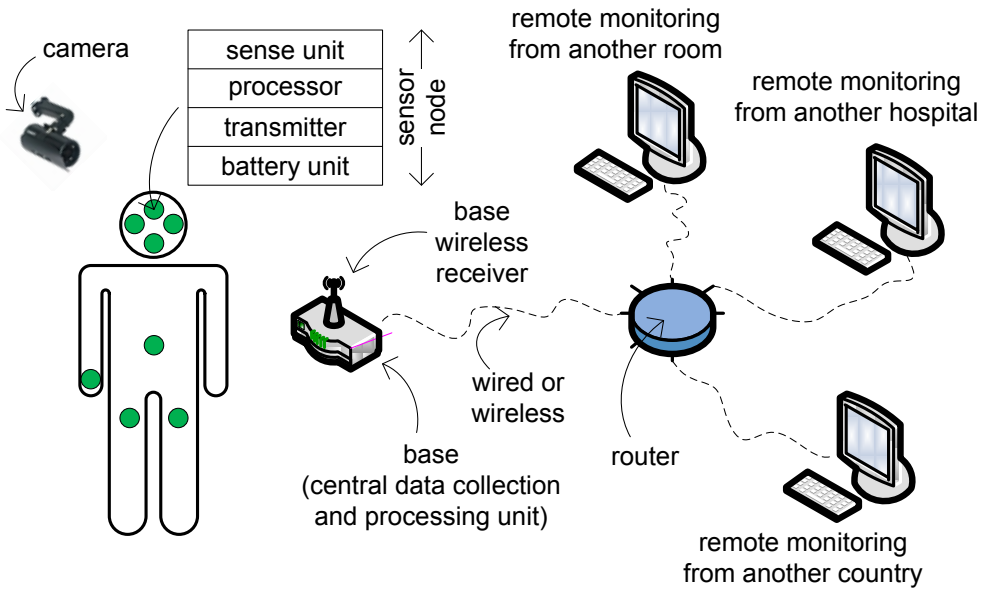


Fig. 44**.** TWPSG architecture

- *sensor unit* – pressure transducer, microphone, piezoelectric bands, accelerometer, etc. This unit is no different from what any currently available PSG is using.
- *processor* – digitalizes sensed analog data, optimizes data, and sends data to the base, via a transmitter. The processor is not available in current PSGs.

- *transmitter* – transfers digital data, using Time Division Multiple Access (TDMA), to the base.
- *battery unit* – powers the node during the test.

Sensor nodes, preferably utilizing low-power ultra-small-size MEMS (microelectromechanical system) technology, are directly attached to a patient. Their sensed data is first locally digitalized and processed using a built-in processor. The processed data is then wirelessly transmitted to the base. In this application, PEWSN architecture consists of a single cluster with the central data collection and processing unit serving as the base but also as the cluster head. In other words, individual sensors never communicate among each other but rather send their sensed data directly to the base. Therefore, TWPSG proposed system is a single-cluster PEWSN-based integrated solution.

The base post-processes the obtained raw data, puts it in the correct format, and sends it to the PSG technician via a wired or wireless link. The PSG technician location can be on-site or remote. All the technician needs is the computer and the installed back-end PSG-data-displaying software.

An additional TWPSG feature is the camera, which can record video and audio of the patient in sleep. The camera not only can be used as a real-time data check for a supervising PSG technician, but also it can be used as the educational material for the patient after the test.

PEWSN-based TWPSG is an all-wireless solution. TWPSG allows the patient to sleep comfortably wire-free during the test. TWPSG also allows the patient to leave the bed, if necessary, without having to hook/unhook any wires.

TWPSG measuring error is significantly reduced because the analog data does not have to travel through the cables, but rather it is digitalized right at the source. Lastly, the solution includes all the benefits of the existing solutions, such as remote data viewing and remote patient monitoring.

6.3.3.2 Sensor Data Collection

TWPSG-underlying technology is PEWSN architecture. In accordance with PEWSN, TWPSG uses TDMA to send data from all sensor nodes back to the base. Therefore, in order to avoid wireless packet collision by the sensor nodes, each node has its own reserved slot within each TDMA frame.

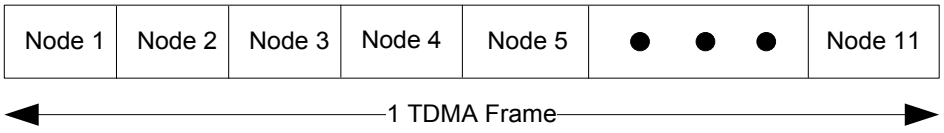


Fig. 45. Single 11-slot TDMA Frame.

Fig. 45 shows a single 11-slot TDMA frame. Each sensor node has one slot allocated for its data within each frame. The slot/frame duration is system-requirements-dependent, and it could be as low as 1ms. Each sensor node can also choose to collect samples during the entire frame, and then send them all together to the base.

The clock synchronization among the nodes, traditionally a difficult problem, is achieved by the base station sending a small beacon at the beginning of each frame. The clock-synchronizing beacon is received by all nodes at the same time, forcing them to adjust their local clocks.

Camera data is usually high bandwidth, and it can run independently from sensor node communication. The camera wireless link can run at a separate

Fig. 46**.** STM32W-SK and STM32W-EXT test sensor nodes.

frequency, therefore avoiding the interface with sensor nodes. Upon reception of both sensor nodes data and camera data, the power rich base station can choose any commercially available protocol, i.e. IP/TCP, to transfer data to the rest of the network.

6.3.3.3 TWPSG Test and Verification

To verify our architecture, we used STMElectronics STM32W108 chipset (system-on-chip that includes a processor and wireless IEEE 802.15.4 based transceiver). Fig. 46 shows the commercial off-the-shelf (COTS) sensor nodes [130]. Eight nodes were positioned on a dummy in the same way as depicted in Fig. 44 (four head nodes, one arm wrist, one abdominal, two on quadriceps (one on each leg)). All nodes included accelerometers to measure the artificial movements we created using the dummy. The goal of the test was to verify TWPSG's communication aspect, so the lack of real polysomnographic sensors did not adversely influence the test. The slot duration was 5ms, resulting in a 40ms (8 * 5 ms) TDMA frame period.

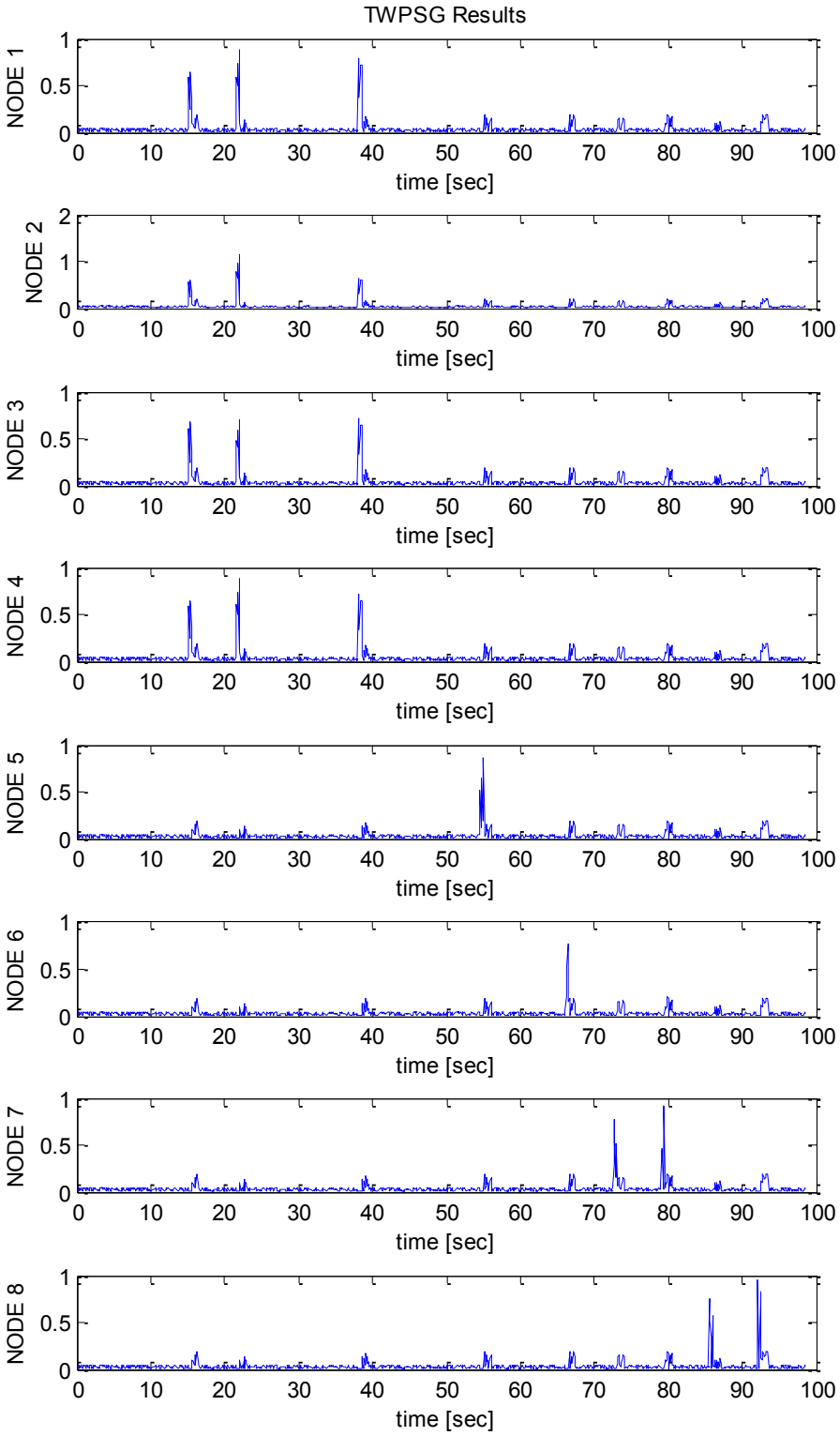


Fig. 47. TWPSG Results

Fig. 47 shows our test results. The data is displayed in a way similar to the way it will appear at the PSG technician's display. We used MATLAB to display the final results. In order to excite the accelerometers we performed 9 dummy shakes (kicks) at various parts of the dummy. The first three were head shakes, the next was an arm shake, then an abdomen shake, and finally two shakes for each leg. As Fig. 47 shows, the first four nodes (NODE 1 through NODE 4) mounted on the dummy's head responded to three instances of head shaking. NODE 5 and NODE 6 responded to single instances of arm and abdomen shaking, NODE 7 and NODE 8 responded to double instances of leg shakes. All 8 nodes were able to detect, and wirelessly transfer, data to the base.

6.3.4 Conclusion

In this section we present our published work [131] on PEWSN-based Truly Wireless Polysomnography (TWPSG) as an alternative to already existing commercial PSG solutions. TWPSG insures an exceptional comfort for the patient, because it excludes any wiring found in traditional and quasi-wireless PSGs. The solution is also more accurate because it digitalizes the analog data right at the source, preventing noise coupling between adjacent cables. TWPSG capability of remote monitoring and controlling of the test procedure makes the architecture cost-effective and appealing for simultaneous multi-patient test environments.

6.4 Temporary Structures Protection

6.4.1 Introduction

Very often there is a need to install temporary structures such as camping grounds, open-field entertainment venues, structures to confine heard of cattle, and structures to allow uninterrupted, undisturbed experimental data collection. Often those structures are setup quickly and do not follow all security and safety measures necessary to protect them. Specifically, those structures are exposed to perimeter intrusions.

This application uses PEWSN-based architecture to protect temporary structures against various types of perimeter intrusions. Our application shall satisfy the following requirements:

- Ground-based perimeter intrusion on a temporary structure shall be detectable
- The location/direction from which the intrusion is launched shall be detectable.
- The intruder's strength shall be detectable.
- The latency from intrusion event detection to triggering the alerting system shall be less than 5 s.
- Network placement shall be simple and intuitive without requiring special skills by network placing personnel.
- Network initialization and configuration shall be automatic.
- Targeted network life shall be up to 1 year with performance emphasis on the first month.
- Network equipment shall be lightweight and easy to carry.

6.4.2 Proposed Solution

6.4.2.1 Network Configuration

The network architecture is based on PEWSN. Fig. 48. shows PEWSN network configuration with 25 clusters (C1 to C25) each having 100 sensors (Sx[0] to Sx[99]). The network's base (data processing hub) is in the center.

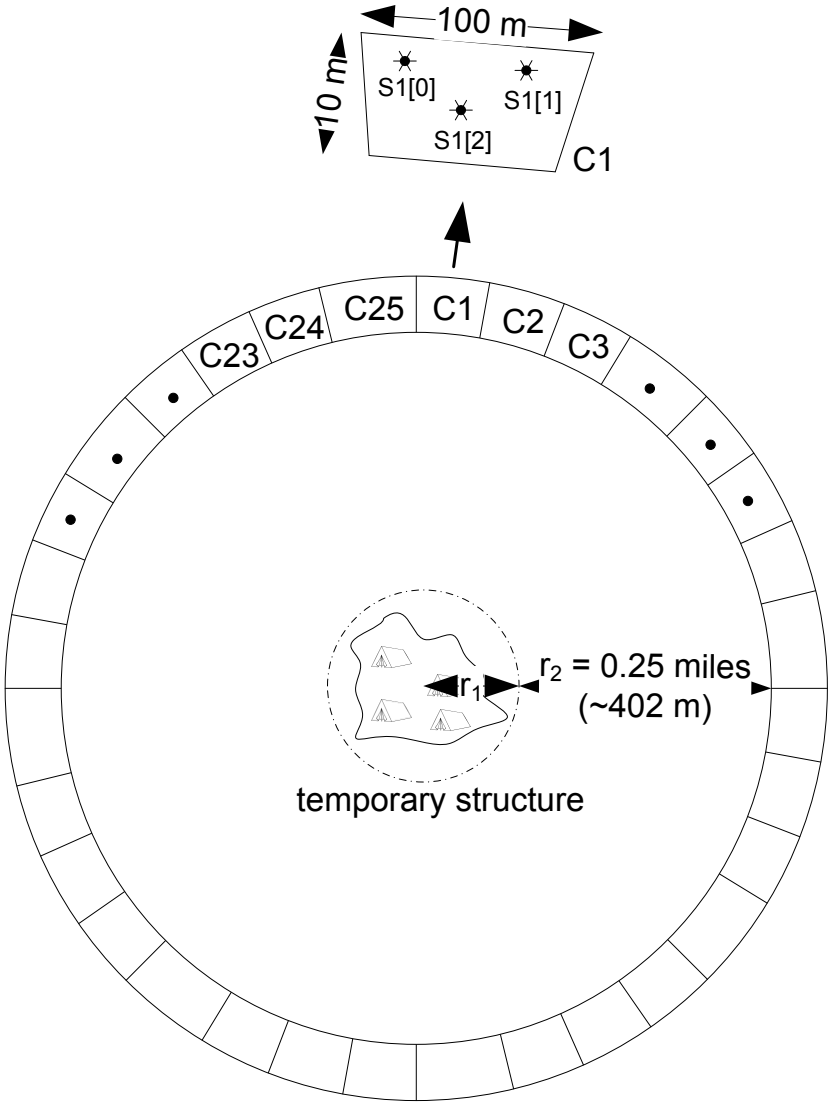


Fig. 48. Temporary structures protection

The configuration is governed by the following parameters:

1. *Number of Clusters* – The requirement is to have the network positioned approximately 0.25 mi (~ 402 m) away from the outer base perimeter (r_1). Therefore, the inner perimeter of the network circle is $2*\pi*(r_1 + r_2)$. For a very small base ($r_1 + r_2 \approx r_2$), the inner perimeter of the network circle is 1.57 mi (~2527 m). According to PEWSN architecture, all sensors are grouped in clusters.

   The cluster's area is chosen by taking into account a number of factors, including the need to simulate the network using IEEE 802.15.4-based transmitters. The range of IEEE 802.15.4 is specified to ~120 m (transmitter power 0 dBm, receiver sensitivity -100 dBm), which is enough to allow the communication of diagonally positioned peer sensors within the same cluster. The specified range, however, might not be optimal in the real environment such as the one targeted by this application. As Kvaksrud [71] points out, the range greatly decreases due to the ground reflection. Signal fading and shadowing might present a significant obstacle to signal range because the sensors in this application are expected to be close to the ground. In addition, environmental range testing by Merrill et al. [72] concluded that the range not only depends on sensor proximity to the ground, but also on the soil type (grass, concrete, iron) onto which the sensors are placed. The solution to a range-unfriendly environment is to either increase each node's transmission power or populate the perimeter area more densely (reducing the distance among individual nodes). Another option is to place nodes as high as

possible above the ground.

Another important factor is the cluster width, which must be long enough to prevent an intruder from crossing the circle undetected. Widening the network perimeter decreases the chance of an undetected intrusion.

In our case, we chose each cluster to cover an area of about 100 m X 10 m. Therefore, to cover a ~2500-m long circle perimeter, we chose a 25-cluster configuration.

2. *Number of Sensors per Cluster* – This number depends on how conveniently a network installer randomly throwing sensors can cover a 100 m X 10 m cluster area. The number also depends, as previously discussed, on the environmental conditions and the potential sensor transmission range. The equipment weight requirement also must be taken into account. We chose 100 sensors per cluster. If each installer covers 1/8 of the circle (Fig. 49), then carrying about 3 clusters or 300 sensors (each sensor ~5 g) results in an overall extra weight of 1.5 kg, which is an acceptable incremental backpack weight. Having 100 sensors per cluster means a coverage area of ~10m^2 per individual sensor, which is a reasonable sensing area for acoustic, motion and photodetecting sensors.
3. *Network Base* – The network base placed in the center serves as the data collection hub, receiving all clusters' data. Since the base is the data hub in the PEWSN network, it should be connected to a high-capacity battery, preferably a rechargeable one (assuming there is no permanent power supply). Since the base is also a single point of failure for the network, a

backup base should also be part of the initial setup.

4. *Alerter Watch* – An alerter watch is a small piece of hardware wrapped around security guards' wrists that receives alerts from the networks base specifying the nature of the threat, the intruders' location, and potentially, the intruders' size/count. The shape and size of the guard's alerter watch should be similar to that of a wristwatch. Ideally, each guard should carry one to ensure redundancy and a rapid response during an intrusion.

6.4.2.2 Network Placement

One of the most important requirements outlined above is network placement. As mentioned before, the placement must be simple and intuitive without requiring special skills by network placing personnel. Fig. 49 outlines the manual placement procedure.

First, the network's base is placed in the middle of the camp. While the temporary base is being built, eight network installers can begin manually placing the sensors in a circle 0.25 mi away from the base. Each network installer is responsible for covering 1/8 of the overall circle, which is about 0.2 mi (~ 314 m). The network installer's only requirements are to stay ~0.25 mi away from the center (easily done via a Global Positioning System (GPS)) and place sensors as evenly as possible across the area covered. The sensor distribution should be done in a similar fashion to spreading corn seeds across a corn field. Eventually, all sensors should build a circle that is ~0.25 mi away from the center of the camp. The uneven shape of the built circle and the somewhat random sensor placement do not diminish the overall network's performance. If faster placement is

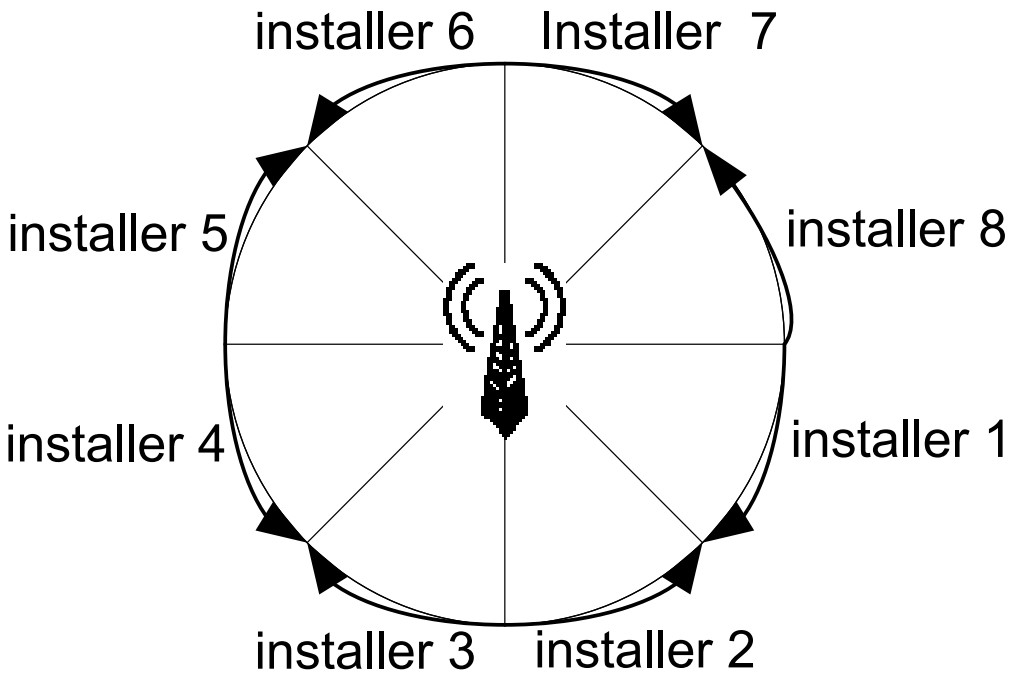


Fig. 49. Sensors placement procedure

necessary or the terrain is extremely difficult for quick placement, the overall network circle can be split into 16 sections (instead of 8); now 16 network installers need to cover half of the area that 8 installers would need to cover.

6.4.2.3 Network Setup and Initialization

The next requirement addresses automatic network setup and initialization. The PEWSN setup and initialization concept inherently satisfies this requirement. Since the network's base is connected directly to all clusters (no cluster-to-cluster hopping needed), the base is where all initial cluster heads are set up. Therefore, the network's base will start interrogating all sensors within the C1 cluster and then choose the C1 cluster head. Then it will move to interrogate all sensors within the C2 cluster and choose the C2 cluster head, and so on until all 25 clusters are set up.

However, before the setup process begins, all sensors must be awakened from deep sleep mode. All sensors are also missing their geolocation. Network installers placing the sensors in the field carry a hardware device that awakens the sensors from deep sleep mode and programs them with their approximate location coordinates. For example, an installer carries a GPS device that sends a location programming message to the nearest placed sensors. In order to prevent large location accuracy errors, the reachable distance for the message should be just about enough to reach only the closest sensors around the installer. The accuracy error can also be minimized by having a short reach of GPS programming messages or by recalibration procedures done after the network is set up. If the application requires high level of location fidelity, various triangulation techniques can be applied to reach the needed precision.

After all sensors are placed and the network completed its setup (initialization) phase, we can assume the network is fully operational and ready to protect the temporary structure.

6.4.2.4 Network Operation Description

As per PEWSN architecture the network operates in TDMA mode. Since there are 25 clusters and 100 sensors per each cluster, a TDMA round comprises of 150 slots. (CLUSTERUP=25, SENSING=100, CLUSTERDOWN=25).

To satisfy the 5-s event reporting latency requirement, each slot has a period of 30 ms (150*30 ms = 4.5 s < 5 s). Therefore, each sensor will report to its cluster head every 30 ms, and each cluster head will report to the networks base every 30 ms (unless there is no event (data) of interest to be reported).

Fig. 50 depicts an intrusion event. An event of interest starts when an intruder crosses the network's circle through cluster C4. C4 sensors as well as some close neighboring sensors will pick up motion, sound, and potentially pressure and will start reporting the event immediately to their cluster heads. Cluster heads will forward the data to the networks base. The base will process the data and send alerts to all alert watches on the base. The alert will contain information about the intruder's location and strength (the force's strength might be determined by the strength of the motion/sound/pressure picked up by the signal).

After an event, a potentially damaged network will have to be repaired manually by replenishing the damaged area with new sensor nodes.

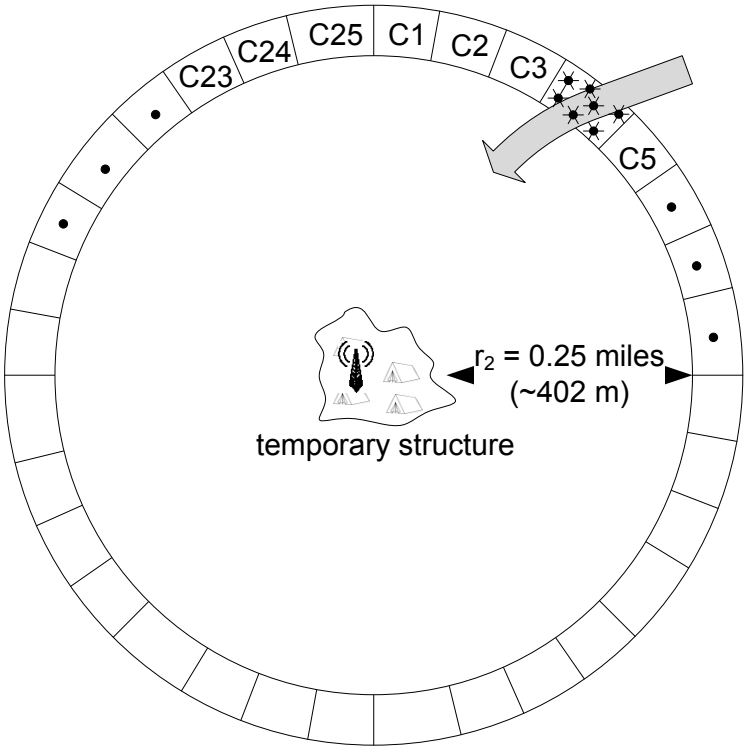


Fig. 50. Intrusion event through cluster C4

6.4.3 Performance

6.4.3.1 Environment

We simulated our perimeter intrusion detection system (PIDS) using PEWSN simulator that is based on OMNET++ framework. Individual sensor nodes, the interaction among the individual nodes, and the interaction between cluster heads and the base use our home-grown custom-developed code. The entire code was written in C++ (in order to make the solution dynamically scalable, we avoided the use of NED OMNET++ native language). Fig. 51 represents OMNET++ graphical view of the network simulation.

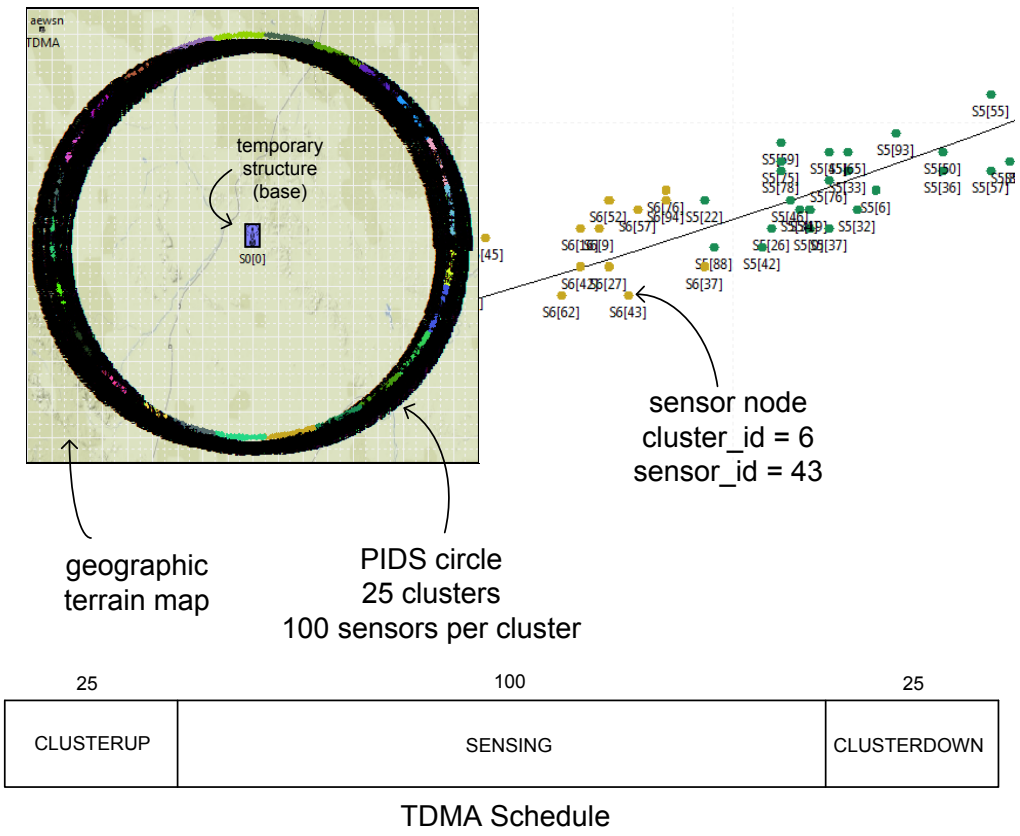


Fig. 51. PEWSN-based simulated PIDS circle

Our simulation environment uses following perimeters:

- No intrusion event (steady-state case)
- 25 clusters with 100 sensors in each cluster
- Network circle radius 0.25 mi (~402 m)
- Each cluster covers area: 100 m X 10 m
- TDMA round had 150 slots with 30-ms slot period (TDMA round latency 4.5 s)
- Packet Size:
    - From sensor to cluster head 47 bits
    - From cluster head to cluster head 151 bits
- Communication standard: IEEE 802.15.4
    - Frequency / rate: 2.4GHz / 250kbps
- Transceiver power consumption (TI CC2420) [68]:
    - RX = 18.8 mA
    - TX = 17.4 mA
    - Crystal oscillator
        - startup time: 0.86 ms
        - current consumption: 426 μA
    - Regulator
        - Startup time: 0.3 ms
        - Current consumption: 20 μA
- Microprocessor power consumption (TI MSP430F2619) [69]:

  - 280 nA (WDT) (sleep)
  - 365 μA at 1 MHz (active)
  - 100 clocks for sensor data processing
- Motion Sensor power consumption (Panasonic EKMB) [132]:
  - 1 μA (sleep)
  - 1 μA (standby for our application)
- Sensor battery capacity (Thinergy MEC201) [133]:
  - Solid-state, thin-film
  - Capacity: 1 mAh
  - Recharge cycles: 100,000
  - Lifetime: 15 years
- Solar Harvester [134]:
  - Average insolation: 3522.5 Whr/m^2/day
  - Average daylight: 12.21 h
  - Solar cell size: from 1 mm^2 to 10 mm^2
  - Conversation efficiency: 3%
  - Loss due to angle of incidence: 20%
  - Spatial and temporal variability ±7%
- Energy-Harvesting Charger and Protector power consumption (MAX17710) [135]:
  - 150 nA (sleep)
  - 725 nA (active)

6.4.3.2 Performance Results

6.4.3.2.1 Setup Phase

Fig. 52 depicts the setup (initialization) phase. The setup phase took 225 s, which is consistent with expectations since it takes 2 TDMA rounds to set up each cluster. Each TDMA round is about 4.5 s. 25 clusters take 50 TDMA rounds to set up the network.

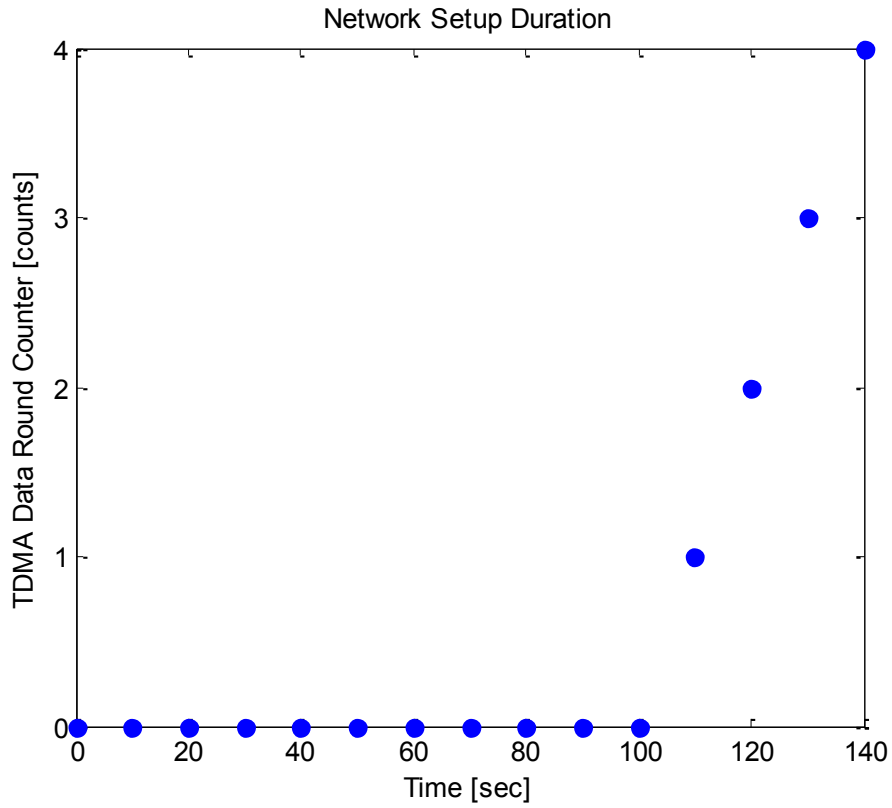


Fig. 52. Network setup time.

6.4.3.2.2 Cluster Head vs. Individual Sensor Lifetime

Fig. 53 plots the network lifetime as the function of the node's battery capacity. This particular simulation case assumes no cluster head switching. Therefore, the cluster head node drains much quicker than any other node in the system. In addition, this test assumes that the nodes do not harvest any energy. No harvesting capability coupled with no cluster head switching capability produces a network lifetime of only 37 h.

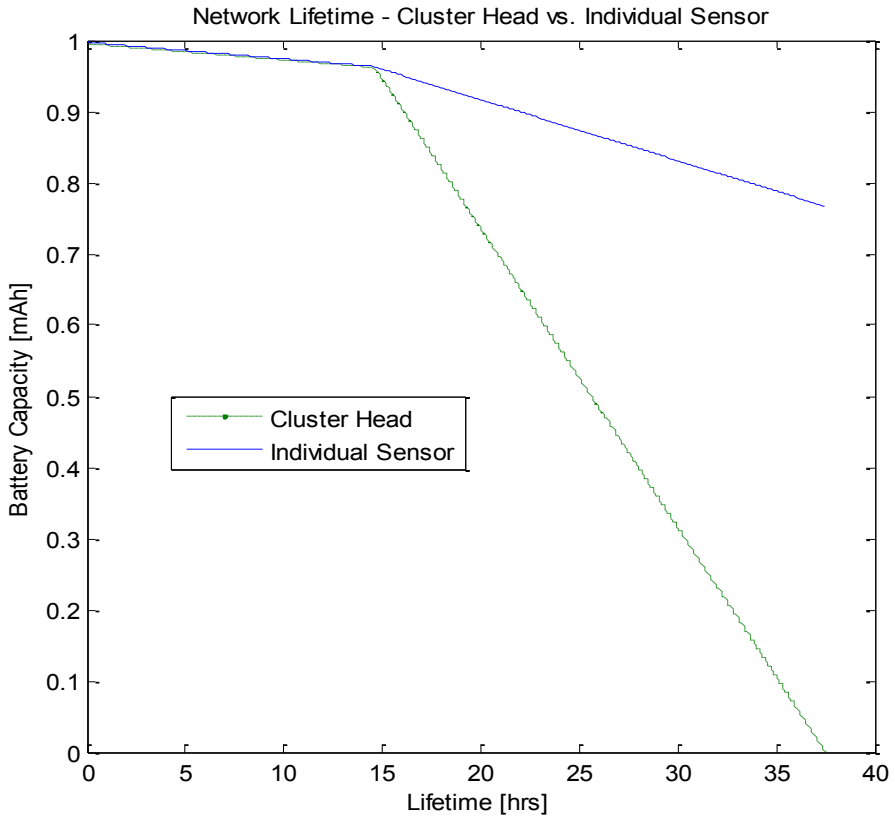


Fig. 53. Cluster head vs. individual sensor lifetime.

The simulation assumes that there are no intrusion events (steady-state case). While simulating an intrusion event is tempting, the goal of this test was to figure out the network's lifetime even when there is no single intrusion event.

Intrusion events are expected to cause faster battery depletion. However, PEWSN automatic communication bandwidth adjustment will offset the power loss due to an intrusion event. Intrusion events are expected to create an issue for the network only if multiple intrusion events happen simultaneously and at the same time the nodes' power harvesting capability is greatly reduced.

6.4.3.2.3 Network life

Fig. 54 depicts a case where energy harvesting capability is used. The goal of this test was to measure the network lifetime under various energy harvesting scenarios.

Additional network's operational assumptions include:

- Each individual sensor is awake for 2 TDMA slots within each TDMA round (150 slots). One slot is used for sensor detection, processing, and transmission, and one slot is used for receiving potential messages from the cluster head. In all other slots, individual sensors are in sleep mode (99% of the time).
- Each cluster head is awake for 100 TDMA slots within each TDMA round (150 slots). The cluster head spends 99 TDMA slots listening and receiving data from other 99 individual sensors within the same cluster. An additional slot is needed for the cluster head to transfer sensed data to the base (CLUSTERDOWN phase). Therefore, in our simulation, the cluster head is awake 67% of the time and asleep 33% of the time. During active time, some cluster head's components are active while other might still stay in sleep (transmitter vs. receiver, processor vs. sense detector etc.)

- The simulator also simulates the case where high power consuming components such as transmitters do not get turned on during sensor's active time if the sensed data are of no value to the base (no intrusion event detected).
- Power consumption during periods such as transmitter turn-on time, clock oscillator turn-on time, phase lock loop calibration time are included in this simulation. On the other hand, some smaller, IC specific power consumption instigators are not taken into account.

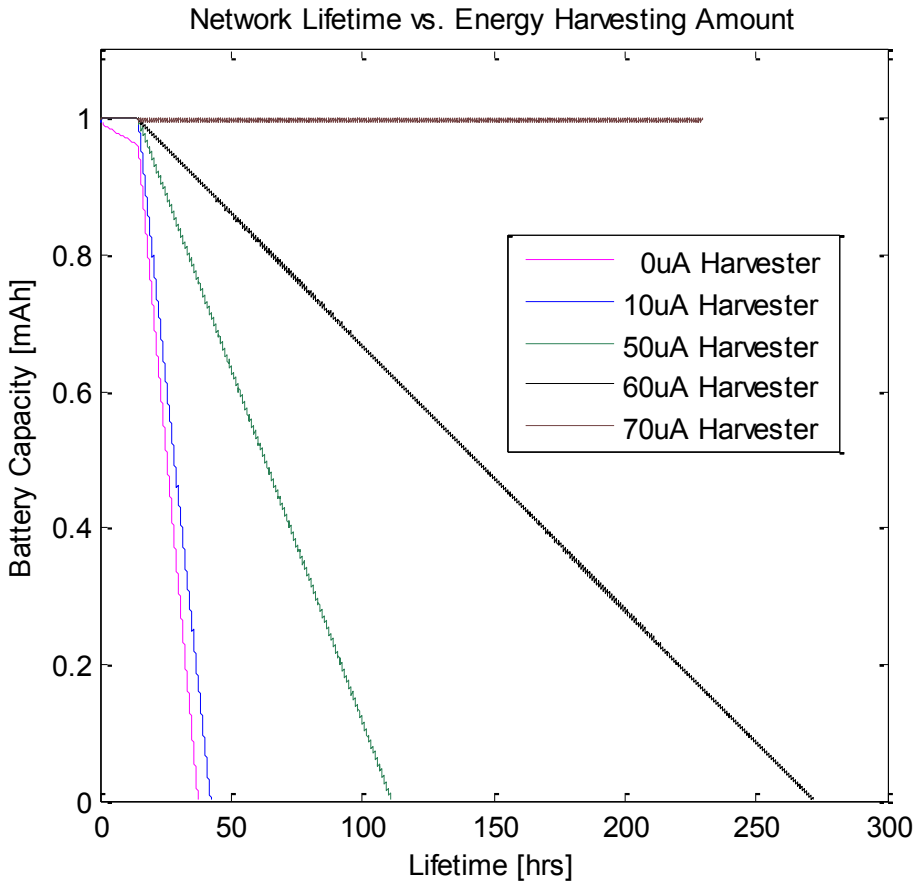


Fig. 54. Network lifetime vs. energy harvesting.

	Case 1	Case 2	Case 3	Case 4	Case 5
Power Harvester [in μA]	0	10	50	60	70
Total life [in h]	37	42	111	271	equilibrium

Table 4. Network lifetime vs. power harvesting

Table 4 shows the test results. Depending on the amount of average current harvested during the test period, the network lifetime varies from 37 h (0 μA) all the way to the equilibrium case. The equilibrium case is reached when the harvesting capability (70 μA in this case) is enough to offset any network power consumption caused by network sensing and reporting the surveillance data.

While the average current is expected to oscillate (day vs. night, sunny vs. rainy day etc.), the network lifetime is not expected to change. This is due to the PEWSN's automatic adjustment of the communication bandwidth; an PEWSN-based network will communicate less and consume less power during nonoptimal energy harvesting periods. Similarly, the network will be very active during periods where the harvesting capabilities match or surpass the equilibrium threshold.

The same applies to intrusion event cases. While the number of intrusions might vary from case to case, the network lifetime is expected to stay constant. The consistency is ensured via the network's adjustable bandwidth.

6.4.4 Conclusion

In this section, we presented PEWSN-based application aimed at protecting temporary structures. The process was to develop PEWSN-based network topology that would satisfy the predefined application requirements.

The PEWSN-based network architecture was able to fulfill all our network-related requirements (event detection latency, network lifetime, coverage area size) as well as specific application-related requirements such as the ease of deployment and automatic setup/initialization.

As our network simulation suggested, the network setup (initialization) was performed within a reasonable time period (225 s). The network was also able to fulfill the application's 1-year lifetime requirement, but only by making an energy harvester part of each sensor. In fact, as our simulation results reveal, an energy harvester (in our case solar cell) capable of generating 70 μA (average current) is needed to keep the network alive for extended periods of time.

CHAPTER 7. CONCLUSION AND FUTURE WORK

7.1 Conclusion

In this dissertation, we have presented a novel WSN hierarchical architecture called PEWSN that is specifically designed to extend network lifetime while ensuring predictable worst-case network latency.

As discussed in CHAPTER 1, our motivation was to reach power equilibrium within a WSN, so that the network lifetime becomes limited only by its hardware components. This effectively means that if sensors are developed using highly reliable parts such as military parts, the network lifetime can be extended significantly. In addition to a long network lifetime, we were also motivated to design an architecture that will have a predictable worst-case latency so that trigger event latency sensitive applications can use WSN technology. The predictable latency, in this case, is defined as the latency that is achieved in nominal environmental conditions where packets can be transferred among individual nodes and sensors without any hindering circumstances. Any hostile environment can degrade the network latency since the transfer might require multiple retransmissions.

CHAPTER 2 presented the results of our extensive survey on prior art architectures. We classified all major architectures into data-centric, hierarchical, location-based, mobility-based, network flow, quality of service, multi-path, and heterogeneity-based.

CHAPTER 3 presented the PEWSN architecture. We described all aspects of the new architecture such as building blocks, TDMA schedule, messages and

transmission phases. Finally, we used the PEWSN (5,3) example to tie all the architectural aspects together.

CHAPTER 4 compared PEWSN performance versus benchmark WSN architectures (directed, MTE, static, and LEACH). We tabulated our results and provided comprehensive results analysis. Our results show that PEWSN outperforms other benchmark architectures using both performance measuring methods (i.e. first dead node and last dead node).

CHAPTER 5 described a self-developed PEWSN simulator that we extensively used in our PEWSN applications.

Finally, CHAPTER 6 presented four PEWSN-based applications. These real-world applications present various PEWSN features and configurations that PEWSN is capable of supporting. They also report PEWSN real-environment performance results that are in-line with our simulated results.

7.2 Future Work

Our immediate future work will be based upon further development of the PEWSN-based applications.

For the *Airport Protection Using Wireless Sensor Networks* application, the next step in our research is to expand our field testing to larger areas such as a 300 feet long chain-link fence populated with about 100 sensors per network line. The goal is to further test the reliability and soundness of the network as well as to observe the environmental impact on the network. We also plan to design a custom sensor node that will be smaller in size than the current commercially

available STMicroelectronics' test node. The custom node will also have specially designed waterproof housing, so that we can test the impact of rain on the network's communication.

For the *Underwater Wireless Sensor Networks,* we plan to further map the PEWSN terrestrial architecture to the underwater networking needs. To do so, we plan to continue our UWSN study, focusing more on the underwater network stack. We expect to spend a fair amount of time on physical layer, because many of the challenges are directly related to the UWSN's physical layer.

For the *Truly Wireless Polysomnography* application, we plan to further validate our architecture by transferring our test environment into a hospital environment that includes a patient, along with actual airflow, pressure and piezoelectric sensors. We also plan to embed camera video and audio data into the final data streams.

Finally, the next step in the *Temporary Structures Protection* application is to generate various realistic intrusion events and monitor the network communication bandwidth response. Our goal is to find out if the network can survive an active intruder's wireless sensor network. For example, if an intruder were to place its own network tasked to drain the power from our network by constantly requesting data, would our network be able to adjust its communication power to survive such an attack? Our future work in regard will also include protecting temporary structures with three concentric circles instead of one. Three circles at distances of 0.25 mi, 0.35 mi, and 0.45 mi will allow us to accurately determine how fast the intruder is advancing, which in turn can trigger some

interesting options in terms of auto-guided response. Tradeoffs will include ease of placement (since network installers would have three circles to field), network management (additional clusters to address), battery consumption, weight, and other application requirements. We also plan to explore transmission power reduction methods such as those described in [136] and [137]. We hope this effort will lead to a reduction in overall sensor's power consumption and therefore reduction in size and weight of the required harvester unit.

BIBLIOGRAPHY

[1] W.B. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," in *Proceedings of the 33rd Hawaii International Conference on System Sciences*, vol. 1, October 2000, pp. 660 - 670.

[2] I.F. Akyildiz, Weilian Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102 -114, August 2002.

[3] K Akkaya and M. Younis, "A survey on routing protocols for wireles ssensor networks," *Elsevier - Ad Hoc Networks*, vol. 3, no. 3, pp. 325-349, May 2005.

[4] Z. Yang and Mohammed A., "A Survey of Routing Protocols of Wireless Sensor Networks," Blekinge Institute of Technology, Sweden, White Paper 2010.

[5] S.K. Singh, M.P. Singh, and D.K Singh, "Routing Protocols in Wireless Sensor Networks – A Survey," *International Journal of Computer Science & Engineering Survey (IJCSES)*, vol. 1, no. 2, pp. 63-83, November 2010.

[6] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *The International Journal of Computer and Telecommunications Networking*, vol. 52, no. 12, pp. 2292-2330, August 2008.

[7] A. Davis and H. Chang, "A Survey of Wireless Sensor Network

Architectures," *International Journal of Computer Science & Engineering Survey (IJCSES)*, vol. 3, no. 6, pp. 1-22, December 2012.

[8] S. Hedetniemi and A. Liestman, "A survey of gossiping and broadcasting in communication networks," *Networks*, vol. 18, no. 4, pp. 319-349, 1988.

[9] W.R. Heinzelman, J. Kulik, and H. Balakrishnan, "Adaptive protocols for information dissemination in wireless sensor networks," in *MobiCom '99 Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, 1999, pp. 174-185.

[10] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: A scalable and robust communication paradigm for sensor networks," in *MobiCom '00 Proceedings of the 6th annual international conference on Mobile computing and networking*, 2000, pp. 56 - 67.

[11] D. Braginsky and D. Estrin, "Rumor Routing Algorithm for Sensor Networks," in *WSNA '02 Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, Atlanta, 2002, pp. 22-31.

[12] C. Schurgers and M.B. Srivastava, "Energy efficient routing in wireless sensor networks," in *MILCOM 2001. Communications for Network-Centric Operations: Creating the Information Force*, 2001, pp. 357-361 vol.1.

[13] R.C. Shah and J.M. Rabaey, "Energy aware routing for low energy ad hoc sensor networks," in *2002 IEEE WCNC2002. Wireless Communications*

and Networking Conference, 2002, pp. 350-355 vol.1.

[14] Y. Yao and J. Gehrke, "The cougar approach to in-network query processing in sensor networks," *SIGMOD*, vol. 31, no. 3, pp. 9-18, September 2002.

[15] N. Sadagopan, B. Krishnamachari, and A. Helmy, "The ACQUIRE mechanism for efficient querying in sensor networks," in *Proceedings of the First IEEE International Workshop on Sensor Network Protocols and Applications*, 2003, pp. 149-155.

[16] D Liu, X. Hu, and X Jia, "Energy efficient information dissemination protocols by negotiation for wireless sensor networks," *Computer Communications*, vol. 29, no. 11, pp. 2136-2149, July 2006.

[17] A. Boukerche, X. Cheng, and J. Linus, "Energy-aware data-centric routing in microsensor networks," in *MSWIM '03 Proceedings of the 6th ACM international workshop on Modeling analysis and simulation of wireless and mobile systems*, 2003, pp. 42-49.

[18] J. Liu, Feng Z., and D. Petrovic, "Information-directed routing in ad hoc sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 4, pp. 851-861, April 2005.

[19] G. Pei and C. Chien, "Low Power TDMA in Large Wireless Sensor Networks," in *Military Communications Conference, 2001. MILCOM 2001. Communications for Network-Centric Operations: Creating the Information Force. IEEE*, 2001, pp. 347 - 351 vol.1.

[20] M. Garla, T.J. Kwon, and G. Pei, "On demend Routing in Large Ad Hoc Wireless Networks with Passive Clustering," in *Proceedings IEEE WCNC 2000*, Chicago, IL, 2000.

[21] O. Younis and S. Fahmy, "HEED: A Hybrid, Energy-Efficient Distributed Clustering Approach for Ad Hoc Sensor Networks," in *IEEE Transactions on Mobile Computing*, 2004, pp. 366-379 Vol.3 No.4.

[22] S. Lindsey and C.S. Raghavendra, "PEGASIS: Power-efficient gathering in sensor information systems," in *IEEE Aerospace Conference Proceedings*, 2002, pp. 3-1125 - 3-1130 vol.3.

[23] S. Lindsay, C.S. Raghavendra, and K.M. Sivalingam, "Data Gathering in Sensor Networks using the Energy Delay Metric," in *IPDPS '01 Proceedings of the 15th International Parallel & Distributed* , 2001, p. 188.

[24] A. Manjeshwar and D.P. Agrawal, "TEEN: a routing protocol for enhanced efficiency in wireless sensor networks," in *Proceedings 15th International Parallel and Distributed Processing Symposium*, 2000, pp. 2009 - 2015.

[25] A. Manjeshwar and D.P. Agrawal, "APTEEN: a hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks," in *Proceedings International Parallel and Distributed Processing Symposium*, 2002, pp. 195 - 202.

[26] M. Younis, M. Youssef, and K. Arisha, "Energy-aware routing in cluster-based sensor networks," in *10th IEEE International Symposium on*

Modeling, Analysis and Simulation of Computer and Telecommunications Systems, 2002, pp. 129 - 136.

[27] M.A. Youssef, M.F. Younis, and K.A. Arisha, "A constrained shortest-path energy-aware routing algorithm for wireless sensor networks," in *IEEE Wireless Communications and Networking Conference*, Orlando, FL, 2002, pp. 794-799 vol.2.

[28] J. Yin and S. Madria, "SecRout: a secure routing protocol for sensor networks," in *20th International Conference on Advanced Information Networking and Applications*, 2006, p. vol 1.

[29] X. Du and F. Lin, "Secure cell relay routing protocol for sensor networks," in *24th IEEE International Performance, Computing, and Communications Conference*, 2005, pp. 477 - 482.

[30] M. McGlynn and S. Borbash, "Birthday protocols for low energy deployment and flexible neighbor discovery in ad hoc wireless networks," in *Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing*, New York, NY, 2001, pp. 137 - 145.

[31] G. Peng, T. Jiang, K. Zhang, and H. Chen, "Clustering algorithm in initialization of multi-hop wireless sensor networks," in *IEEE Transactions on Wireless Communications*, 2009, pp. 5713 - 5717 Vol.8 Issue 12.

[32] Z. A. Eu, H.Tan, and W.K.G. Seah, "Routing and Relay Node Placement in Wireless Sensor Networks Powered by Ambient Energy Harvesting," in *IEEE Wireless Communications and Networking Conference*, 2009, pp. 1-

6.

[33] Y. Xu, J. Heidemann, and D. Estrin, "Geography-informed energy conservation for Ad Hoc routing," in *Proceedings of the 7th annual international conference on Mobile computing and networking MobiCom*, 2001, pp. 70-84.

[34] Y. Yu, R. Govindan, and D. Estrin, "Geographical and Energy Aware Routing: A Recursive Data Dissemination Protocol for Wireless Sensor Networks," University of California Los Angeles , Los Angeles, Technical Report No. UCLA/CSD-TR-01-0023, 2001.

[35] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "Span: AEnergy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks," in *Proceedings ACM MobiCom'01*, Rome, Italy, 2001, pp. 85-96.

[36] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "Span: An Energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks," *Wireless Networks*, vol. 8, no. 5, pp. 481-494, September 2002.

[37] B. Nath and D. Niculescu, "Routing on a curve," *ACM SIGCOMM Computer Communication Review*, vol. 33, no. 1, pp. 155-160, January 2003.

[38] M. Zorzi and R. R. Rao, "Geographic random forwarding (GeRaF) for ad hoc and sensor networks: Mutlihop performance," *IEEE Transactions on*

mobile Computing, vol. 2, no. 4, pp. 337-348, Oct.-Dec. 2003.

[39] R. Zhang, H. Zhao, and M.A. Labrador, "The anchor location service (ALS) protocol for large-scale wireless sensor networks," in *Proceedings of the First International on Integrated Internet Ad hoc and Sensor Networks*, Nice, France, 2006.

[40] G. Xing, C. Lu, R. Pless, and Q. Huang, "On greedy geographic routing algorithms in sensing-covered networks," in *Proceedings ACM MobiHoc'04*, Tokyo, Japan, 2004, pp. 31-42.

[41] V. Rodoplu and T.H. Meng, "Minimum energy mobile wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 8, pp. 1333-1344, August 1999.

[42] L. Li and J.Y. Halpern, "Minimum-energy mobile wireless networks revisited," in *IEEE International Conference on Communications*, 2001, pp. 278-283.

[43] K. Seada, M. Zuniga, A. Helmy, and B. Krishnamachari, "Energy-efficient forwarding strategies for geographic routing in lossy wireless sensor networks," in *Proceedings of the Sensys'04*, Baltimore, MD, 2004.

[44] H.S. Kim, T.F. Abdelzaher, and W.H. Kwon, "Minimum-Energy asynchronous dissemination to mobile sinks in wireless sensor networks," in *Proceedings of the 1st international conference on Embedded networked sensor systems*, 2003, pp. 193-204.

[45] F. Ye, H. Luo, J. Cheng, S. Lu, and L. Zhang, "A two-tier data

dissemination model for large-scale wireless sensor networks," in *Proceedings of the 8th annual international conference on Mobile computing and networking*, 2002, pp. 148-159.

[46] J. Luo and J. P. Hubaux, "Joint mobility and routing for lifetime elongation in wireless sensor networks," in *Proceedings IEEE INFOCOM'05*, Miami, FL, 2005, pp. 1735-1746.

[47] R.C. Shah, S. Roy, S. Jain, and W. Brunette, "Data MULEs: Modeling a three-tier architecture for sparse sensor networks," in *Proceedings SNPA '03*, Anchorage, AK, 2003, pp. 30-41.

[48] W. Chang, G. Cao, and T. La Porta, "Dynamic proxy tree-based data dissemination schemes for wireless sensor networks," in *Proceedings IEEE MASS'04*, Fort Lauderdale, FL, 2004, pp. 21-30.

[49] 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, October 2000.

[50] T. He, J. Stankovic, and C. Abdelzaher, T. Lu, "SPEED: A stateless protocol for real-time communication in sensor networks," in *Proceedings of International Conference on Distributed Computing Systems*, Providence, RI, 2003.

[51] K. Akkaya and M. Younis, "An Energy-Aware QoS Routing Protocol for Wireless Sensor Networks," in *Proceedings of the IEEE Workshop on Mobile and Wireless Networks*, Providence, RI, 2003.

[52] Z. Liu and I. Elhanany, "RL-MAC: A QoS-Aware Reinforcement Learning based MAC Protocol for Wireless Sensor Networks," in *Proceedings of the 2006 IEEE International Conference on Networking, Sensing and Control*, 2006, pp. 768-773.

[53] E. Felemban, Chang-Gun Lee, and E. Ekici, "MMSPEED: Multipath Multi-SPEED Protocol for QoS Guarantee of Reliability and Timeliness in Wireless Sensor Networks," *IEEE Transactions on Mobile Computing*, vol. 5, no. 6, pp. 738-754, June 2006.

[54] M. Perillo and W. Heinzelman, "DAPR: a protocol for wireless sensor networks utilizing an application-based routing cost," *IEEE Wireless Communications and Networking Conference*, vol. 3, pp. 1540-1545, March 2004.

[55] 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, August 2004.

[56] K. Kalpakis, K. Dasgupta, and P. Namjoshi, "Efficient algorithms for maximum lifetime data gathering and aggregation in wireless sensor networks," *Computer Networks*, vol. 42, no. 6, pp. 697-716, August 2003.

[57] F. Ye, A. Chen, S. Lu, and L. Zhang, "A scalable solution to minimum cost forwarding in large sensor networks," in *Tenth International Conference on Computer Communications and Networks*, 2001, pp. 304-309.

[58] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin, "Highly-Resilient,

Energy-Effcient Multipath Routing in Wireless Sensor Networks," *ACM SIGMOBILE*, vol. 5, no. 4, pp. 11-25, October 2001.

[59] W. Lou, "An efficient N-to-1 multipath routing protocol in wireless sensor networks," in *IEEE International Conference on Mobile Adhoc and Sensor Systems*, 2005, pp. 672- 8pp.

[60] N. Nasser and Y. Chen, "SEEM: Secure and energy-efficient multipath routing protocol for wireless sensor networks," *Computer Communications*, vol. 30, no. 11-12, pp. 2401-2412, September 2007.

[61] B. Yahya and J. Ben-Othman, "REER: Robust and Energy Efficient Multipath Routing Protocol for Wireless Sensor Networks," in *GLOBECOM - Global Telecommunications Conference*, 2009, pp. 1-7.

[62] Y.H. Wang, H.J. Mao, C.H. Tsai, and C.C. Chuang, "HMRP: Hierarchy-Based Multipath Routing Protocol for Wireless Sensor Networks," in *Embedded and Ubiquitous Computing - EUC 2005 Workshops*, Nagasaki, Japan, 2005, pp. 6-9.

[63] M. Chu, H. Haussecker, and F. Zhao, "Scalable information-driven sensor querying and routing for ad hoc heterogeneous sensor networks," *International Journal of High Performance Computing Applications*, vol. 16, no. 3, pp. 293-313, February 2002.

[64] X. Du and F. Lin, "Improving routing in sensor networks with heterogeneous sensor nodes," in *Proceedings IEEE VTC'05*, Dallas, TX, 2005, pp. 2528-2532.

[65] A. Hadjidj, A. Bouabdallah, and Y. Challal, "HDMRP: An Efficient Fault-Tolerant Multipath Routing Protocol for Heterogeneous Wireless Sensor Networks," in *7th International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness (Qshine)*, Houston, TX, 2010.

[66] G. Smaragdakis, I. Matta, and A. Bestavros, "SEP: A Stable Election Protocol for clustered heterogeneous wireless sensor networks," in *International Workshop on SANPA*, 2004.

[67] D. Kumara, T. C. Aserib, and R.B. Patel, "EEHC: Energy efficient heterogeneous clustered scheme for wireless sensor networks," *Computer Communications*, vol. 32, no. 4, pp. 662-667, March 2009.

[68] Texas Instruments, "Single-Chip 2.4 GHz IEEE 802.15.4 Compliant and ZigBee™ Ready RF Transceiver," Datasheet 20 Mar 2007.

[69] Texas Instruments, "MSP430F241x, MSP430F261x Mixed Signal Microcontroller," Datasheet 12 Dec 2011.

[70] A. Shrestha and L. Xing, "A Performance Comparison of Different Topologies for Wireless Sensor Networks," in *2007 IEEE Conference on Technologies for Homeland Security*, 2007, pp. 280 - 285.

[71] T. Kvaksrud, "Range Measurements in an Open Field Environment," Texas Instruments Design Note DN018.

[72] W.M. Merrill, H.L.N. Liu, J. Leong, K. Sohrabi, and G.J. Pottie, "Quantifying short-range surface-to-surface communications links," *IEEE*

Antennas and Propagation Magazine, vol. 46, no. 3, pp. 36-46, June 2004.

[73] Infinite Power Solutions, "THINERGY MEC 201," Datasheet 2011.

[74] H. Kopetz and W. Ochsenreiter, "Clock Synchronization in Distributed Real-Time Systems," in *IEEE Transactions on Computers*, Aug. 1987, pp. 933 - 940; C-36, Issue: 8.

[75] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcasts," in *Proceedings of the 5th symposium on Operating systems design and implementation*, New York, USA, 2002, p. Volume 36 Issue SI.

[76] I. Bekmezci and F. Alagoz, "A New TDMA Based Sensor Network for Military Monitoring (MIL-MON)," in *IEEE Military Communications Conference, MILCOM 2005.*, 2005, pp. 2238 - 2243 Vol. 4.

[77] S.F. Hallowell and P.Z. Jankowski, "Transportation security technologies research and development," in *IEEE Military Communications Conference (MILCOM), 2005.*, 2005, pp. 1753 - 1756 Vol. 3.

[78] M.C. Schiefelbein, "Information Architecture for Threat Detection Systems," in *IEEE Conference on Technologies for Homeland Security*, 2008, pp. 589 - 592.

[79] A. Waters, "The design and use of BALANCE for homeland security," in *The Signal Processing Solutions for Homeland Security* , 2005, pp. 0-10.

[80] J. Pita et al., "Deployed ARMOR protection: the application of a game theoretic model for security at the Los Angeles International Airport," in

7th international joint conference on Autonomous agents and multiagent systems, 2008, pp. 125-132.

[81] G. Pestana, "LocON: A location based services platform to improve airport safety," in *IEEE Aerospace Conference*, 2011, pp. 1-10.

[82] K.B. Lee and M.E. Reichardt, "Open standards for homeland security sensor networks," in *IEEE Instrumentation & Measurement Magazine*, 2005, pp. 14 - 21 , Volume: 8 , Issue: 5.

[83] E. Saputra, K.A. Bakar, H. Herman, and S. Hassan, "Novel Framework of Integrated Security and Safety System Using Hybrid Network Technology," in *11th International Conference on Computer Modelling and Simulation*, 2009, pp. 368-373.

[84] "Security technology," *IEEE Aerospace and Electronic Systems Magazine*, vol. 15, no. 10, pp. 131 - 136, October 2000.

[85] R. Gomery and G. Leach, "Fence vibrations," in *IEEE 33rd Annual 1999 International Carnahan Conference on Security Technology*, 1999, pp. 377-381.

[86] A.S. Barry and J. Czechanski, "Ground surveillance radar for perimeter intrusion detection," in *The 19th Digital Avionics Systems Conference*, 2000, pp. 7B5/1 - 7B5/7 vol.2.

[87] A.S. Barry and D.S. Mazel, "The Secure Perimeter Awareness Network (SPAN) at John F. Kennedy International Airport," in *41st Annual IEEE International Carnahan Conference on Security Technology*, 2007, pp.

183-188.

[88] David S. Mazel and Ann Barry, "Mobile Ravin: Intrusion Detection and Tracking with Organic Airport Radar and Video Systems," in *40th Annual IEEE International Carnahan Conferences Security Technology*, 2006, pp. 30-33.

[89] A.A. Dibazar et al., "Intelligent acoustic and vibration recognition/alert systems for security breaching detection, close proximity danger identification, and perimeter protection," in *IEEE International Conference on Technologies for Homeland Security (HST)*, 2010, pp. 351-356.

[90] T. Kumagai, A. Ogura, K. Tan, W. Ohnuki, and T. Sato, "Optical Fiber Vibration Sensor for Intrusion Detection," Hitachi Cable Review, 2008.

[91] S. Dewar, "Opportunities for increased use of standards in the integration of Perimeter Intrusion Detection Systems," in *42nd Annual IEEE International Carnahan Conference on Security Technology (ICCST)*, 2008, pp. 305-311.

[92] S. Hennin, G. Germana, and L. Garcia, "Integrated Perimeter Security System," in *IEEE Conference on Technologies for Homeland Security*, 2007, pp. 70-75.

[93] A.H. Navin, B. Asadi, S.H. Pour, and M. Mirnia, "Solving Coverage Problem in Wireless Camera-Based Sensor Networks by Using Genetic Algorithm," in *2010 International Conference on Computational Intelligence and Communication Networks (CICN)*, 2010, pp. 226-229.

[94] Y. Liu, Chao Li, Yang He, Jing Wu, and Zhang Xiong, "A Perimeter Intrusion Detection System Using Dual-Mode Wireless Sensor Networks," in *CHINACOM '07. Second International Conference on Communications and Networking in China*, 2007, pp. 861-865.

[95] M. Maki, C. Hill, and C.R. Malone, "User performance testing of the Perimitrax buried cable sensor," in *IEEE 33rd Annual 1999 International Carnahan Conference on Security Technology*, 1999, pp. 112-119.

[96] STMicroelectronics. (2010, September) Starter and extension kits for STM32W108xx microcontrollers.

[97] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-Efficient Communication Protocol for Wireless Microsensor Networks," *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 660-670, October 2002.

[98] G. Pei and C. Chien, "Low Power TDMA in Large Wireless Sensor Networks," in *Military Communications Conference, 2001. MILCOM 2001. Communications for Network-Centric Operations: Creating the Information Force. IEEE*, 2001, pp. 347 - 351 vol.1.

[99] STMicroelectronics. (2012, March) STM32W108xx - High-performance, IEEE 802.15.4 wireless system-on-chip Datasheet.

[100] A. Davis and H. Chang, "Airport Protection using Wireless Sensor Networks," in *2012 IEEE Internation Conference on Technologies for Homeland Security*, Waltham, MA, 2012.

[101] J. Partan, J. Kurose, and B.N. Levine, "A survey of practical issues in underwater networks," in *1st ACM international workshop on Underwater networks*, New York, 2006, pp. 17-24.

[102] I.F. Akyildiz, D. Pompili, and T. Melodia, "Underwater acoustic sensor networks: research challenges," *Elsevier*, vol. Ad Hoc Networks, no. 3, pp. 257-279, February 2005.

[103] J. Heidemann, Y. Wei, J. Wills, A. Syed, and Yuan L., "Research challenges and applications for underwater sensor networking," in *IEEE Wireless Communications and Networking Conference*, 2006, pp. 228-235.

[104] J. Heidemann, M. Stojanovic, and M. Zorzi, "Underwater sensor networks: applications, advances and challenges," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* , vol. 370, no. 1958, pp. 158-175, January 2012.

[105] P. Xie et al., "Efficient vector-based forwarding for underwater sensor networks," *EURASIP Journal on Wireless Communications and Networking - Special issue on radar and sonar sensor networks*, vol. 2010, p. 4, April 2010.

[106] V. Chandrasekhar, W. KG. Seah, Y.S. Choo, and H.V. Ee, "Localization in underwater sensor networks: survey and challenges," in *WUWNet '06 Proceedings of the 1st ACM international workshop on Underwater networks*, 2006, pp. 33-40.

[107] J. Elson, L. Girod, and E. Estrin, "Fine-grained network time

synchronization using reference broadcasts," in *Fifth Symposium on Operating Systems Design and Implementation*, Boston, MA, 2002, pp. 147-163.

[108] A.A. Syed and J. Heidemann, "Time Synchronization for High Latency Acoustic Networks," in *25th IEEE International Conference on Computer Communications*, 2006, pp. 1-12.

[109] I.F. Akyildiz, D. Pompili, and T. Melodia, "Challenges for efficient communication in underwater acoustic sensor networks," *ACM SIGBED Review - Special issue on embedded sensor networks and wireless computing*, vol. 1, no. 2, pp. 3-8, July 2004.

[110] A. Ozgur and I.F. Akyildiz, "Event-to-sink reliable transport in wireless sensor networks," *IEEE/ACM Transactions on Networking (TON)*, vol. 13, no. 5, pp. 1003-1016, October 2005.

[111] L. Liu, S. Zhou, and Cui J.H., "Prospects and problems of wireless communication for underwater sensor networks," *Wireless Communications & Mobile Computing - Underwater Sensor Networks: Architectures and Protocols*, vol. 8, no. 8, pp. 977-994, October 2008.

[112] J. Kong, J.H. Cui, D. Wu, and M. Gerla, "Building underwater ad-hoc networks and sensor networks for large scale real-time aquatic applications," in *IEEE Military Communication Conference (MILCOM)*, 2005, pp. 1535-1541.

[113] J.H. Cui, J. Kong, M. Gerla, and S. Zhou, "The challenges of building

mobile underwater wireless networks for aquatic applications," *IEEE Network*, vol. 20, no. 3, pp. 12-18, May-june 2006.

[114] D. Pompili, T. Melodia, and I.F. Akyildiz, "Routing algorithms for delay-insensitive and delay-sensitive applications in underwater sensor networks," in *12th annual international conference on Mobile computing and networking*, 2006, pp. 298-309.

[115] I. Vasilescu, K. Kotay, D. Rus, M. Dunbabin, and P. Corke, "Data collection, storage, and retrieval with an underwater sensor network," in *3rd international conference on Embedded networked sensor systems*, 2005, pp. 154-165.

[116] A. Davis and H. Chang, "Underwater Wireless Sensor Networks," in *IEEE OCEANS 2012 Hampton Roads*, Virginia Beach, VA, 2012, pp. 1-5.

[117] T. Mooe, K.A. Franklin, U. Wiklund, T. Rabben, and K. Holmstrom, "Sleep-Disordered Breathing and Myocardial Ischemia in Patients With Coronary Artery Disease," *CHEST*, vol. 117, no. 6, pp. 1597-1602, June 2000.

[118] Y. Peker et al., "An independent association between obstructive sleep apnoea and coronary artery disease," *Eur Respir J.*, vol. 14, no. 1, pp. 179-184, July 1999.

[119] J.L. Reishtein, "Obstructive sleep apnea: a risk factor for cardiovascular disease," *Journal of Cardiovascular Nursing*, vol. 26, no. 2, pp. 106-116, March-April 2011.

[120] R.A. Dart, J.R. Gregoire, D.D. Gutterman, and Woolf S.H., "The association of hypertension and secondary cardiovascular disease with sleep-disordered breathing," *Chest*, vol. 123, no. 1, pp. 244-260, January 2003.

[121] K. Bagai, "Obstructive sleep apnea, stroke, and cardiovascular diseases.," *Neurologist*, vol. 16, no. 6, pp. 329-339, November 2010.

[122] M.E. Dyken and K.B. Im, "Obstructive sleep apnea and stroke," *Chest*, vol. 136, no. 6, pp. 1668-77, December 2009.

[123] Y. Kaneko, V.E. Hajek, V. Zivanovic, J. Raboud, and T.D. Bradley, "Relationship of sleep apnea to functional capacity and length of hospitalization following stroke," *Sleep*, vol. 26, no. 3, pp. 293-297, May 2003.

[124] O. Parra, A Arboix, S Bechich, L Garcia-Eroles, and J Montserrat, "Time course of sleep-related breathing disorders in first-ever stroke or transient ischemic attack," *American Journal Of Respiratory and Critical Care Medicine*, vol. 161, no. 2, pp. 375-380, February 2000.

[125] T.E. Wessendorf, H. Teschler, Y.M. Wang, N. Konietzko, and A.F. Thilmann, "Sleep-disordered breathing among patients with first-ever stroke," *Journal of Neurology*, vol. 247, no. 1, pp. 41-47, January 2000.

[126] J.D. Lattimore, D.S. Celermajer, and I. Wilcox, "Obstructive sleep apnea and cardiovascular disease," *Journal of the American College of Cardiology*, vol. 41, no. 9, pp. 1429-37, May 2003.

[127] A.S. Baran and Richert A.C., "Obstractive sleep apnea and depression," *CNS Spectrums*, vol. 8, no. 2, pp. 128-34, 2003.

[128] (2012, August) Wikipedia. [Online]. http://en.wikipedia.org/wiki/Polysomnography

[129] R.J. Farney et al., "Polysomnography in Hospitalized Patients Using a Wireless Wide Area Network," *Journal of Clinical Sleep Medicine*, vol. 2, no. 1, pp. 28-34, 2006.

[130] (2012, July) STMElectronics. [Online]. http://www.st.com/internet/com/TECHNICAL_RESOURCES/TECHNICAL_LITERATURE/DATA_BRIEF/CD00285442.pdf

[131] A. Davis and H. Chang, "TWPSG: Truly Wireless Polysomnography," in *IEEE Conference on Translational Engineering in Health & Medicine*, Houston, TX, 2012.

[132] Panasonic, "Standard Profile (1,2,6 μA) Passive Infrared Motion Sensor," Datasheet 2010.

[133] Infinite Power Solutions. (2011) Thinergy MEC 201 Datasheet.

[134] S. Wilcox and C.A. Gueymard. Spatial and Temporal Variability of the Solar Resource in the United States. [Online]. http://rredc.nrel.gov/solar/new_data/variability/Documentation/ASES_47760_final.pdf

[135] Maxim, "MAX17710 Energy-Harvesting Charger and Protector," Datasheet Nov 2011.

[136] S. Panichpapiboon, G. Ferrari, and O.K. Tonguz, "Optimal Transmit Power in Wireless Sensor Networks," *IEEE Transactions on Mobile Computing*, vol. 5, no. 10, pp. 1432 - 1447 , October 2006.

[137] Z. Eu, Hwee-Pink Tan, and W.K.G. Seah, "Routing and Relay Node Placement in Wireless Sensor Networks Powered by Ambient Energy Harvesting," in *IEEE WCNC 2009. Wireless Communications and Networking Conference*, 2009, pp. 1-6.