Using genetic algorithms to optimise Wireless Sensor Network Design

Using Genetic Algorithms to optimize Wireless Sensor Network Design

by

Jin Fan

A Doctoral Thesis

Submitted in partial fulfilment

of the requirements for the award of

Doctor of Philosophy

of

Loughborough University

19th October 2009

Loughborough University

Thesis Access Form

Copy No…………...…………………….**Location**……………………………………………………….……………...…

Author…………...………………………………………………………………………………………..…….

Title……..

Status of access OPEN / RESTRICTED / CONFIDENTIAL

Moratorium Period:…………………………………years,ending…………../…………200……………………….

Conditions of access approved by (CAPITALS):…………………………………………………………………

Director of Research (Signature)……………………………………………...……………………………………...

Department of……………………………………………………………………...…………………………………

Author's Declaration: *I agree the following conditions:*

OPEN access work shall be made available (in the University and externally) and reproduced as necessary at the discretion of the University Librarian or Head of Department. It may also be copied by the British Library in microfilm or other form for supply to requesting libraries or individuals, subject to an indication of intended use for non-publishing purposes in the following form, placed on the copy and on any covering document or label.
*The statement itself shall apply to **ALL** copies:*

This copy has been supplied on the understanding that it is copyright material and that no quotation from the thesis may be published without proper acknowledgement.

Restricted/confidential work: All access and any photocopying shall be strictly subject to written permission from the University Head of Department and any external sponsor, if any.

Author's signature……………………………………….Date…………………………………...…………...……...

users declaration: for signature during any Moratorium period (Not Open work): ***I undertake to uphold the above conditions:***			
Date	Name (CAPITALS)	Signature	Address

Research Student Office, Academic Registry
Loughborough University, Leicestershire, LE11 3TU, UK
Switchboard: +44 (0)1509 263171 Fax: +44 (0)1509 223938

Loughborough University

Certificate of Originality

This is to certify that I am responsible for the work submitted in this thesis, that the original work is my own except as specified in acknowledgements or in footnotes, and that neither the thesis nor the original work contained therein has been submitted to this or any other institution for a higher degree.

. . . .

Jin Fan

19th November 2009

Abstract

Wireless Sensor Networks(WSNs) have gained a lot of attention because of their potential to immerse deeper into people' lives. The applications of WSNs range from small home environment networks to large habitat monitoring. These highly diverse scenarios impose different requirements on WSNs and lead to distinct design and implementation decisions.

This thesis presents an optimization framework for WSN design which selects a proper set of protocols and number of nodes before a practical network deployment. A Genetic Algorithm(GA)-based Sensor Network Design Tool(SNDT) is proposed in this work for wireless sensor network design in terms of performance, considering application-specific requirements, deployment constrains and energy characteristics. SNDT relies on offline simulation analysis to help resolve design decisions. A GA is used as the optimization tool of the proposed system and an appropriate fitness function is derived to incorporate many aspects of network performance. The configuration attributes optimized by SNDT comprise the communication protocol selection and the number of nodes deployed in a fixed area.

Three specific cases : a periodic-measuring application, an event detection type of application and a tracking-based application are considered to demonstrate and assess how the proposed framework performs. Considering the initial requirements of each case, the solutions provided by SNDT were proven to be favourable in terms of energy consumption, end-to-end delay and loss. The user-defined application requirements were successfully achieved.

Acknowledgements

I would like to thank my supervisor Prof. David Parish for giving me guidance and advice everytime I needed. He is my mentor not only in my research area, but also in career.

I would also like to thank all of my colleagues in High Speed Networks group for their support and for sharing their knowledge, especially Dr John Whitley who give me loads of help on latex. Thanks to Dr. Mark Withall and Dr. Marcelline Shirantha de Silva for forcing me use linux at the very first day of my PhD.Thanks to Dr. Konstantinos Kyriakopoulos for his valuable support in various aspects of research life.

Special thanks to Junfeng WU for being with me during these years in Loughborough. Without you, I can not even imagine I could come to the end of my PhD. And finally, I must thank my parents for their support throughout my time at Loughborough.

Contents

List of Figures

List of Tables

List of Corrections

Chapter 1

Wireless Sensor Network Design

A Wireless Sensor Network(WSN) consists of a set of nodes of typically low performance. They collaborate with each other to perform sensing tasks in a given environment. A wireless sensor network may contain one or more sink nodes (Base Stations) to collect sensed data and relay it to a central processing and storage system. A sensor node is typically powered by a battery and can be divided into three main functioned units: a sensing unit, a communication unit and a processor unit. Recent advances in micro-electro-mechanical systems technology, wireless communications and digital electronics have boosted the development of sensor nodes. This brings the blooming prospect of WSNs into practical feasibility.

1.1 Typical WSN Applications

Wireless sensor networks may consist of many different types of sensors such as seismic, thermal, visual, infrared, acoustic and rader etc. [1], which enables wireless sensor networks to be projected into a broad range of applications. The concept of micro-sensing and wireless communication of these nodes promise many new application areas. The understanding of these different application areas and their requirements is crucial to the design of a wireless sensor network. An overview of WSN applications is introduced in this section.

1.1.1 Environment Monitoring

Wireless sensor networks can be used to monitor environmental conditions such as temperature and humidity, which enables applications such as wildlife habitat monitoring, forest fire detection and flood detection. Habitat monitoring has been proposed in [3] as a driver application for wireless sensor networks.

The Great Duck Island project [4] aimed at non-intrusive monitoring of the Pertel(sea bird) nesting behaviour. A sensor network was deployed to collect environmental variables such as relative humidity, temperature conditions and pressure level. These variables were collected in and around bird nests without disturbing the birds.

The researchers in University of Hawaii use sensor networks equipped with environmental sensors and cameras to investigate why endangered vegetable spices are more likely to grow in a certain areas other than else [5]. Sensor networks can also be deployed to monitor forest fires. The FireBug project at UC Berkeley uses sensor nodes to collect temperature, relative humidity and barometric pressure. This data is used in order to detect the initiation and evolution of wildfires. Early deployment served as a proof of concept that wireless sensor networks have the potential use for real-time fire monitoring[6] .

1.1.2 Home Applications

As sensor nodes can be designed into very small shapes, it becomes possible to bury these nodes into home appliances, such as the hoover, wash machine, etc. [7]. A sensor network can be formed and interact with the outside world, for example via the Internet, to allow applications such as remote management of home devices. Moreover, these nodes can be integrated with existing embeded devices to form a self-organizing, self-regulated, even an adaptive system.

The Intelligent Home Project (IHome) at the University of Massachusetts uses a set of distributed smart agents deployed throughout the house to coordinate shared resource utilization[8]. Shared resources include water, electricity and heating. A cost function can be associated with the usage of each resource with sensor data used to decide levels of activation of different resources. For example, the heater activation can be controlled by sensors detecting the presence or absence of individuals in different rooms. The Aware Home Project at Georgia Tech aimed to offer a better care of the elderly and the Smart Kindergarten projects at UCLA applying sensor networks to early childhood nursery.

1.1.3 Health Systems

People believe that a reactive system can help in the reduction of the rising overall cost of health care worldwide. The use of sensor networks can help create a more proactive health system as opposed to the current one. Wearable health monitoring systems are a key to health monitoring, which can lead to early detection of disease signs allowing its prevention or early treatment[9]. In addition, sensor networks could be used to monitor and report on the elderly and patients requiring continuous supervision.

Other potential applications of wireless sensor networks in the health care field introduced by [1] include tracking and monitoring of doctors and patients inside a hospital, collection of human physiological data and drug administration in hospitals.

1.1.4 Military Applications

Wireless sensor networks have a great potential for use in the military field [10] to monitor friendly forces and equipment; track and target enemy forces; assess battle damages and participate in reconnaissance operations. The rapid deployment, self-organization and fault tolerance characteristics of sensor networks make them a very promising sensing technique for military needs.

1.1.5 Other Applications

Wireless sensor networks can be designed and implemented into industrial applications by taking the different requirements of each type of industry into account. WSNs are capable of monitoring the quality of the air, and the temperature of a building. Moreover, they could play a vital role in detecting an emergency such as a fire. Besides, they could control the complex machinery set, the goods producing line, even the conditions of the production system of a group of factories.

Intel research labs and BP(British Petrol), one of the world's largest petroleum and petrochemicals companies, are collaborating on a joint research project using a wireless sensor network to provide continuous vibration monitoring of the engines on one of BP's oil tankers off the Shetland Islands in northern Scotland [11].

Other applications including traffic monitoring where sensor networks are used for vehicle detection and classification [12] and international border monitoring

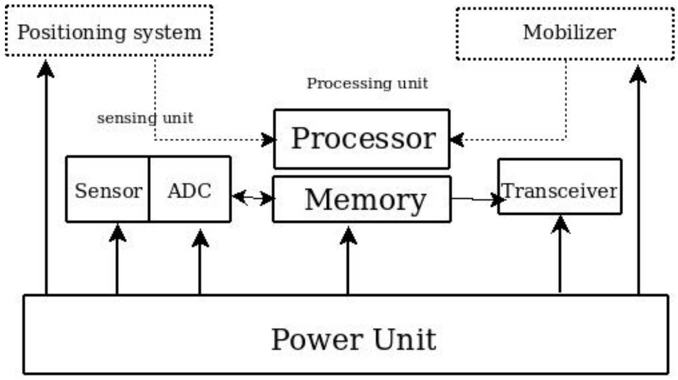


Figure 1.1: Sensor node Architecture [1]

where a sensor network is used to detect intrusions along the US-Mexican borders are proposed [13].

1.2 Sensor node architecture

A sensor node has to be equipped with the right sensors, the necessary computation unit, memory resources, and adequate communication facilities to fulfill certain task. Normally, a sensor node is comprised of four basic components: one or more sensor elements, a battery (power unit), a memory and processor unit, and a transceiver, as shown in Figure 1.1.

A node may also have additional application-dependent components such as a position finding system, a mobilizer or a power generator. Sensing units are usually composed of sensors and analogue to digital converters (ADCs). The analogue signals produced by the sensors (based on the observed phenomena) are converted to digital signals by the ADC. The converted signals are received by the processing unit. The processing unit, which is associated with a certain amount of memory, manages the node working with others when executing the sensing task.

Available sensors in the market include generic (multi-purpose) nodes and gateway (bridge) nodes. A generic (multi-purpose) sensor node's task is to take measurements from the monitored environment. Gateway(bridge) nodes gather data from generic sensors and relay them to the base station. Gateway nodes have higher pro-cessing capability, battery power, and transmission (radio)range. A combination of generic and gateway nodes is typically deployed to form a WSN.

Crossbow [14] makes three Mote processor radio module families MICA [MPR300] (first generation), MICA2 [MPR400] and MICA2-DOT [MPR500] (second generation). MICA2 is the most versatile wireless sensor network device in the market for prototyping purposes in terms of wide usage. MICA2 nodes come with five sensors installed (i.e. Temperature, Light, Acoustic (Microphone), Acceleration/Seismic, and Magnetic). These nodes are especially suitable for monitoring networks for people or vehicles. Different sensors can be installed if required. The operating frequency is in an ISM band, either 916 MHz or 868 MHz, with a data rate of 38.4 Kbits/sec and a communication range of 30 ft to 500 ft [15]. Each node has a low power microcontroller processor with a clock speed of 4MHz, a flash memory with 128 Kbytes, and SRAM and EEPROM of 4K bytes each. A MICA2 node requires with two AA batteries . It could have more than one year life in the sleeping mode. The operating system is Tiny-OS, a tiny micro-threading distributed operating system developed by UC Berkeley, with a NES-C (Nested C) source code language (similar to C).

1.3 The differences between WSNs and traditional networks

Wireless sensor networks, on the one hand, share the similarity of self-configuration without manual management with Mobile ad-hoc networks, on the other hand, they are different from traditional networks in many aspects due to their strict energy constraints and application-specific characteristics.

- NO one-size-fits-all solution

  A WSN is organized as a collection of sensor nodes which co-ordinate with each other to fulfill a certain task. The entire network infrastructure depends directly on the specific application scenario. It is unlikely that a one-size-fits-all solution exists for all these different applications [16]. The old fixed protocol stack which applied successfully to traditional networks is no longer suitable for WSNs. Many new communication algorithms have been developed for different applications. As one example, WSNs are deployed with very different network densities, from sparse to dense deployments. Each case requires unique network configuration.

- Environment interaction

  The traffic loads relayed in WSNs are generated by the sensors which interact entirely with the environment. By contrast, the traffic loads of tradition network are mainly driven by human behaviour. Moreover, the environment plays a key role in determining the size of the network, the deployment scheme, and the network topology. The size of the network varies with the monitored environment. For indoor environments, fewer nodes are required to form a network in a limited space whereas outdoor environments may require more nodes to cover a larger area.

- Resource constraints

  Resource constraints include a limited amount of energy, short communication range, low bandwidth, and limited processing and storage in each node. For wireless sensor networks, energy is a scare resource. This is unlike wireless ad-hoc networks which can recharge or replace batteries quite easily. In some cases, the need to prolong the lifetime of a sensor node has a deep impact on the entire WSN system architecture.

- Reliability and QoS

  The WSNs exhibit very different concepts of reliability and quality of service from traditional networks. They totally depend on the task assigned. In some emergency cases, only occasional delivery of packets can be more than enough; in other cases, very high reliability requirements exist. Packet delivery ratio in WSNs is no longer an sufficient metric, instead, different applications may take their own requirements into consideration.

1.4 Design challenges

WSNs distinguish themselves from traditional networks due to their application-specific and energy constraints. Their structure and characteristics depend on their electronic, mechanical and communication limitations but also on application-specific requirements.

One of the major and probably most important challenges in the design of WSNs is their application-specific characteristic. A sensor network is set up to fulfill a specific task and the data collected from the network may be of different types due to various application scenarios. Respectively, different types of applications have their own specific requirements. These requirements are turned into specific design properties of a WSN. In other words, a WSN's architecture directly depends on the assigned application scenarios. For the acceptable performance of a given task, the optimal WSN infrastructure should be selected out of the hundreds of network solutions before the practical deployment.

Equally, an issue that has been frequently emphasized in the research literature is the fact that energy resources are significantly limited[1]. Recharging or replacing the battery of sensor nodes may be difficult or impossible. Hence, power efficiency often turns out to be the major performance metric, directly influencing the network lifetime. Power consumption according to the functioning of a sensor node can be divided into three domains: sensing, communication, and data processing. There has been research effort in hardware improvements to optimize the energy consumed by sensing and data processing. Several studies of energy efficiency of WSNs have been discussed and several algorithms that lead to optimal connectivity topologies for power conservation have been proposed [17][18].

Another issue in the design of WSNs is that performance assessment of a WSN always happens once deployed. The analysis procedure follows the order that people in this field first put more and more effort into inventing new protocols and new applications; then the solutions are built, tested and evaluated either by simulation or testbeds; even sometimes an actual system has to be deployed so that researchers can learn by empirical evidence. A more scientific analysis procedure is ideally required before a WSN is practically deployed. Current WSN designers are mainly experts in wireless sensor networking and hardware who could perceive the communication between each nodes at the bit level. When a new protocol is developed, they could construct algorithms even if the required simulation tool did not exist. As WSNs immerse deeper into people's work, they must begin to include less specialized users.

1.5 Current Design Tools

There are a few design tools for WSNs, however, most of these tools either hardly consider communication protocol selection or ignore the motivation of providing the whole pack solution based on expert knowledge. The approaches of these tools are not based on the performance assessment of a WSN.

Tinker[19] is a high-level design tool for sensor networks which uses actual data streams from the deployment site to decide on data processing algorithms. Various algorithms can be compared to find the one that gives the best result for the given application.

SensDep [20] is a design tool for WSNs which considers the trade-off between coverage and the cost of the system. SensDep considers mobility and differential surveillance requirements. The optimal solution presented in the paper works for small scale systems only and is based on integer mathematical programming. The heuristic solutions work by generating a list of deployment patterns and matching the deployment patterns that perform well with devices.

ANDES [21] is a WSN design tool which is developed by extending the AADL/OSATE framework. AADL/OSATE has been widely used for real-time and embedded systems and provides a component-based framework for modeling hardware and software components as well as the interaction between these components. ANDES enables designers to systematically develop a model for the system, refine it by tuning the system parameters based on existing analysis techniques.

1.6 Thesis Contributions and Outline

There is a growing need for tools that could help WSN system designers in the selection of different protocols before a practical network deployment, since many new communication protocols which consider the WSNs'features have been designed to meet various application requirements. The severe energy constraints of the sensor nodes and the application-specific characteristics of their use, present major configuration challenges for a WSN designer. Properly designing a WSN before deployment is crucial and involves trades-off between many competing goals.

The work described in this thesis considers these issues and proposes a GA-based design tool which aids sensor network designers in system performance tuning before a network is practically deployed. The main aim of the work is to develop a system which could select a suitable set of protocols out of hundreds of configuration possibilities as well as the optimal number of nodes needed in a fixed area in a certain application scenario . This offline procedure is to be achieved by identifying the acceptable performance of a WSN for a given task under different sets of protocols configurations.

Our main contribution lies in introducing a design tool for WSNs where few currently existed, and providing an optimization mechanism for users to find the optimal set of the protocols out of hundreds of configuration possibilities as well as the optimal number of nodes needed in a fixed area in a certain application scenario before physical deployment. Secondly, an informed performance function is derived in this work to model the performance of a WSN which takes application-specific parameters, connectivity and scalability parameters and energy efficiency parameter into account. Mostly, the novelty of this work stands in the development of an integrated GA approach which considers network characteristics as well as emphasizing the application-specific requirements represented in the performance function to give a fair WSN design optimization before a WSN is actually set up.

This thesis is organized in the following way:

Chapter 2 introduces several most commonly used MAC protocols and Routing protocols in WSNs. These protocols have to be designed with concern for the inherent features of the WSN along with the application and architecture requirements. Several open issues of communication protocols are also discussed.

Chapter 3 presents three different approaches to optimization of a wireless sensor network for its specific design criteria. They are introduced in a progressive way as our optimization procedure is proposed for WSNs at the system level.

Chapter 4 introduces the general concept of Genetic Algorithms. Three genetic operators in GA are presented and corresponding pseudo algorithms are rewritten. Several successful applications to WSNs 'design are presented as well.

Chapter 5 extends the motivation of this work at first. The issue of wireless sensor network performance modeling is addressed by investigating a series of performance metrics. The methodology of applying a GA into this work is also presented.

Chapter 6 presents an overview of major components of this framework and the general operation flow. Simulation-related and GA-related implementation challenges are addressed in this chapter respectively.

Chapter 7 is divided into three sections. Each section represents an unique wireless sensor network case study. We use the proposed framework to provide appropriate solutions for these applications and validate the design decisions by analysis.

Chapter 8 provides a summative conclusion of the aim and results of this thesis. It also discusses the possible future work covering all the limitations in this work.

1.7 Summary

In this chapter, recent wireless sensor network applications are first presented and discussed. Sensor node architecture is then introduced to help understanding the differences between wireless sensor networks and traditional networks. Due to the application-specific characteristic and the resource constraints , wireless sensor network design is challenged in many different domains.

The unique characteristics of wireless sensor networks described in this chapter and the nature of their applications motivate the development of new algorithms for WSNs. In next chapter, a short survey on recent communication protocols is presented.

Chapter 2

WSN Communication Protocols

The architecture of WSNs vary in their complexity as a result of different application requirements in terms of latency, accuracy and network lifetimes. Many protocols have been proposed to meet different application requirements. These protocols reduce the complexity of its functions by using in-network, distributed processing tasks at different protocol task levels, as shown in Figure 2.1. However, not all sensor nodes have every protocol layer. For instance, the transport layer is used when a WSN connects with other external networks. In this case, the network layer is used by the nodes at the access points between the two networks.

In the following section, several most commonly used MAC protocols and Routing protocols are discussed respectively.

2.1 Medium access control protocols in WSNs

A medium access control (MAC) protocol plays a vital role in determining the efficiency of wireless channel bandwidth sharing and energy cost of communication in WSNs. Traditional MAC protocols focus on improving fairness, latency, band-

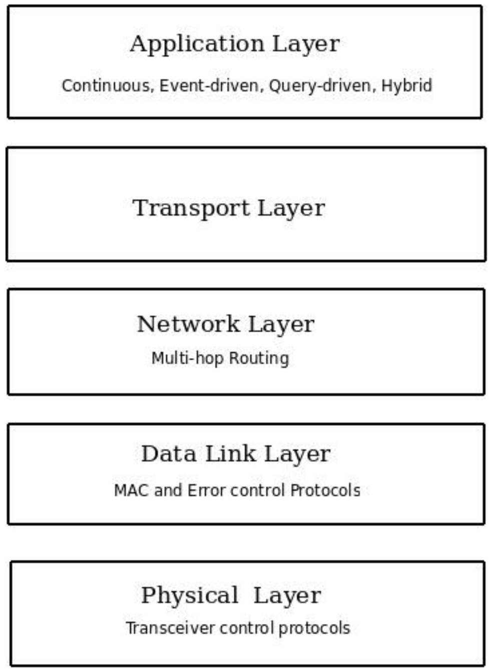


Figure 2.1: WSN Protocol Stack

width utilization and throughput (which are secondary consideration for WSNs). Studies reveal that energy wastage in existing MAC protocols occurs mainly from collision, overhearing, control packet overhead and idle listening . MAC protocols for WSNs should try to avoid the above mentioned energy wastage. While allocating shared wireless channels fairly among sensor nodes, MAC protocols should prevent nodes from transmitting at the same time. The following three protocols introduced below are the most popular ones used at recent research.

2.1.1 IEEE/802.11 standards

IEEE/802.11 standards protocols for wireless local network (WLAN) are based on contention schemes. They have been widely used to observe the sensor network performance at the early stage of development. Even now, IEEE/802.11 standards protocols are still the frequently used MAC protocol for WSN simulations.

The protocols for IEEE/802.11 use a scheme known as carrier-sense, multiple access, collision avoidance (CSMA/CA) which can minimize collision by using four different types of packets: request-to-send(RTS), clear-to-send(CTS), data and acknowledge(ACK) to transmit frames, in a sequential way. In CSMA, a node listens to the channel before transmitting. If it detects a busy channel, it delays access and retries later. Packets that collide are discarded and will be retransmitted later. However, it is easily to point out the drawbacks of IEEE/802.11 protocols when they are applied to WSNs. In terms of energy efficiency, any kinds of collisions in CSMA/CA are definitely energy wasting; In addition, idle listening and overhearing of IEEE/802.11 are the major other sources of energy consumption.

2.1.2 Sensor MAC(S-MAC)

To reduce the energy consumption and support self-configuration, W. Ye [22][23] introduced an energy efficient MAC protocol presented as sensor-MAC(S-MAC) for WSNs. Building on contention-based protocols such like IEEE/802.11 MAC protocols, S-MAC tries to retain the flexibility of contention-based protocols while improving energy efficiency in multi-hop networks. Besides energy efficiency, S-MAC has good scalability and collision avoidance capability by using a combined scheduling and contention scheme. The protocol consists of three major components:

1. Periodic listen and sleep reduce energy consumption; meanwhile synchronization among neighbouring nodes will coordinate with the sensing node to minimize the latency.

2. Collision and overhearing avoidance by the use of in-channel signaling to put each node to sleep when its neighbour is transmitting to another node;

3. Message passing which can reduce application perceived latency and control overhead.

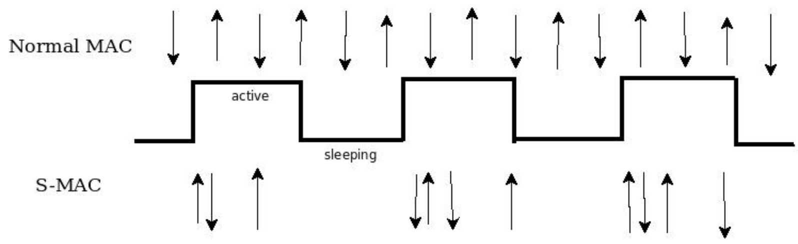


Figure 2.2: S-MAC Duty cycle

The basic idea of this single-frequency contention-based protocol [24] is that time is divided into fairly large frames. Every frame has two parts: an active part and a sleeping part. During the sleeping part, a node turns off its radio to preserve energy(Note that the energy consumption during the state transition is assumed to be little [25]). During the active part, it can communicate with its neighbours and send any messages queued during the sleeping part, as shown in Figure 2.2. Since all messages are packed into the active part, instead of being 'spread out' over the whole frame, the time between adjacent messages, and therefore the energy wasted on idle listening, is reduced.

2.1.3 IEEE/802.15.4 Standard

The IEEE/802.15.4 MAC is a new standard to address the need for low rate, low power, low cost wireless networking, i.e. low rate wireless personal area networks (LR-WPAN). Traditional wireless local area networks (WLANs) aim to provide high throughput, low-latency for generally file size data transfer or multimedia applications. However, the required data rate for LR-WPAN applications is expected to be low, and the tolerant message latency may be of the order of 1s which is a rather rough requirement compared with relative latency requirement in WLANs [26].

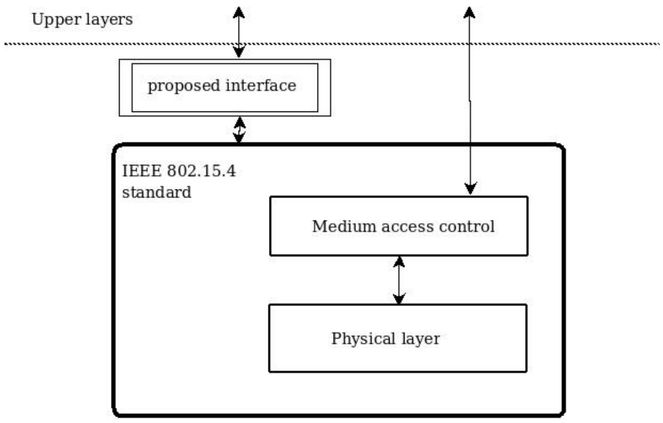


Figure 2.3: IEEE/802.15.4 stack protcols

The standard specifies its lower protocol layers: the physical layer, medium access control (MAC) portion of the data link layer and the operations in this standard to be carried out in the unlicensed 2.4 GHz, 915 MHz and 868 MHz ISM bands [27]. The MAC protocol in IEEE 802.15.4 can operate in both beacon enabled and non-beacon modes. In the beaconless mode, the protocol is just CSMA/CA (carrier sense, multiple access/collision avoidance) protocol; in the beacon mode, there is a superframe structure shown in Figure 2.4 for time-division multiplexing in IEEE/802.15.4.

The superframe may consist of both an active and inactive period. The active portion of the superframe, which contains 16 equally spaced slots, is composed of three parts: a beacon, a contention access period(CAP), and a contention free period (CFP). The beacon is transmitted without the use of CSMA at the start of slot 0 and the CAP commences immediately after the beacon. The coordinator only interacts with nodes during the active period and may sleep during the inactive period. There is a guaranteed timeslot (GTS) option in 802.15.4 to allow

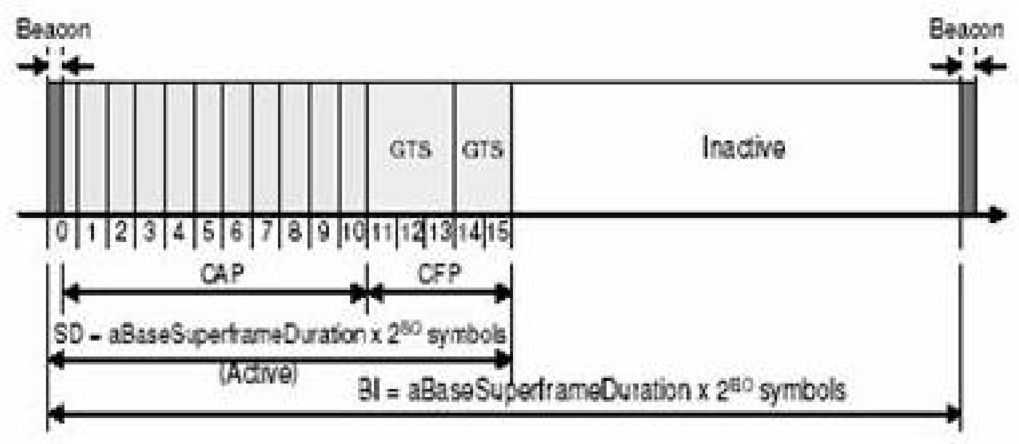


Figure 2.4: An example of Superframe of beacon mode in IEEE/802.15.4 [26]

lower latency operation. There are a maximum of 7 of the 16 available timeslots that can be allocated to nodes, singly or combined.

In a beacon-enabled mode, beacons are transmitted periodically as the synchronization signals. Any device in the network can only perform data transmission with slotted CSMA/CA in the contention access period (CAP) within the superframe period. In this way, IEEE/802.15.4 can achieve the aim of low power consumption.

2.1.4 Other MAC layer algorithms and open issues

DMAC[28] is a traffic adaptive MAC protocol that is based on slotted ALOHA. Transmission slots are assigned to a set of nodes based on a data gathering tree. When a target node has received a slot, all of its children can transmit, thus contending over the medium. As slots are successive in the data transmission path, the end-to-end latency is low. The problem in DMAC is that collisions between nodes in the same level of the tree are common. Also, as knowledge of the data transmission path is required, DMAC is not suitable for dynamic networks.

I.F.Akyildiz et al in [1] also introduced some other MAC protocols for WSNs, such as hybrid TDMA/FDMA which is based on hardware approach to minimize the energy consumption. Though so many MAC schemes have been proposed for WSNs, there are still a lot of open issues for researchers to consider, such as unexplored domains of the error control operations of a WSN, and the MAC protocols for mobile sensor networks.

2.2 Routing protocols in WSNs

Routing problems in WSNs are fairly challenging due to the inherent characteristics of WSNs. Firstly, there is no need to put global addressing into WSNs due to the relatively small number of sensor nodes. Therefore, traditional IP-based protocols can not be applied to WSNs. Secondly, sensor nodes which are deployed in an ad-hoc manner need to be self-organizing. Furthermore, the nodes are constrained in terms of energy, processing, and storage ability so that routing protocols in WSNs require carefully considered resource management functions. Meanwhile, the high probability of data redundancy needs to be exploited by the routing protocols to improve the energy efficiency and bandwidth utilization. A lot of new algorithms (which have taken consideration of these genuine features of WSNs) have been proposed to address the routing problems aforementioned. The routing techniques such as data aggregation, in-network processing, clustering, different node role assignment and data centric methods are employed in different algorithms.

In [29], Al-Karaki classified those protocols into flat, hierarchical, or location-based catalogues according to the network structure. For flat networks, all the involved nodes play the same role; meanwhile hierarchical protocols aim to use the cluster-head nodes for data aggregation and reduction to increase energy efficiency; whilst location-based protocols utilize position information to carry on the assigned task in specific regions rather than the whole network. In the following section, some significant protocols under this classification are discussed.

2.2.1 Flooding

Flooding is the classic mechanism to relay data in a WSN without the need for any routing algorithm and topology maintenance. It is easy to implement with no costly energy need for topology maintenance but has some significant disadvantages. First is implosion, which is caused by duplicate messages being sent to the same node. Secondly, overlapping which occurs when two nodes sensing the same region will send similar packets to the same neighbour. Another is resource blindness caused by consuming large amounts of energy without consideration for the energy constraints when applied to WSNs [1][30]. The following data-centric protocols: SPIN and Directed Diffusion are all designed to address the disadvantages of basic flooding by negotiation and resource adaptation.

2.2.2 Sensor protocols for information via negotiation (SPIN)

Heinzelman et.al. in[31] and [32] proposed a family of adaptive protocols called Sensor Protocols for Information via Negotiation(SPIN) that disseminate all the information at each node to every node in the network assuming that all other nodes in the network are potential base-stations. This enables a user to query any node and get the required information immediately.

The SPIN family protocols are designed from two basic ideas. Firstly, the SPIN family of protocols makes use of the property that nodes in close range might have the similar data. Hence, nodes just need to send the data that describes the sensor message instead of sending the whole data to conserve the energy; meanwhile, each node has its own resource manager which keeps track of resource consumption. It is polled by the nodes before data transmission. Secondly, using the meta-data negotiation solves the classic problem of flooding (i.e. implosion, overlapping and resource blindness) such that SPIN achieves a certain amount of energy efficiency. SPIN has three types of messages ADV, REQ and DATA to communicate. ADV contains a descriptor, i.e. meta-data, and is used to advertise new data, REQ to request data, and DATA is the actual message itself. The protocol starts when a SPIN node obtains new data that it is willing to share. If a neighbour is interested in the data, it sends a REQ message for the DATA and the DATA is sent to this neighbour node. The neighbour sensor node then repeats this process with its neighbours. As a result, the entire sensor area will receive a copy of the data.

The SPIN family of protocols actually includes many protocols. The 3-stage protocol we introduced before is SPIN-1, other protocols of SPIN family are as follows(for details refer to [31][32])

1. SPIN-2: an extension to SPIN-1, incorporates a threshold-based resource awareness mechanism in addition to negotiation .

2. SPIN-BC: This protocol is designed for broadcast channels.

3. SPIN-PP: designed for a point to point communication, i.e., hop-by-hop routing.

4. SPIN-EC: similar to SPIN-PP, but with an energy heuristic added to it.

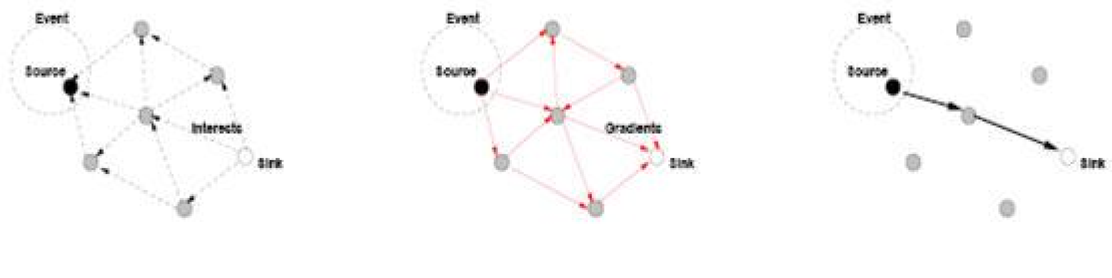
5. SPIN-RL: When a channel is lossy, a protocol called SPIN-RL is used where

adjustments are added to the SPIN-PP protocol to account for the lossy channel.

2.2.3 Directed Diffusion

Directed diffusion is an important data-centric routing protocol. It can be applied to applications which are based on the queries. All sensor nodes in a directed diffusion based network are application-aware, which enables the algorithm to achieve energy efficiency by selecting empirically good paths as well as caching and processing data in the network. The idea aims at using Data named by attribute-value pairs generated by sensor nodes to get rid of unnecessary operations of the network layer in order to save energy.

A node requests data by sending an Interest for named data. Data matching the Interest is then drawn down towards that node. Intermediate nodes can cache, or transform data, and may convey interests based on previously cached data. Figure 2.5 displays a typical example of the working of directed diffusion ((a) sending interests, (b) set up gradients, (c) path reinforcement and send data).



(a) Interest propagation (b) Initial gradients set up (c) Data delivery along reinforced path

Figure 2.5: An example of interest diffusion in sensor network [2]

In directed diffusion, sensor nodes measure events and create gradients of information in their respective neighbourhoods. The sink node requests data by broadcasting interests. Then interest diffuses through the network hop-by-hop, and is broadcast by each node to its neighbours. As the interest is propagated throughout the network, gradients are setup to draw data satisfying the query towards the requesting sink node. A gradient represents both the direction towards which data matching an interest flows and the status of that demand (whether it is active or inactive and possibly the desired update rate) [2]. Each sensor node (which receives the interest) can establish a gradient towards the sensor node from which it receives the interest. This process carries on until gradients are established from the sources back to the sink. In other words, a gradient specifies an attribute value and a direction. At this stage, loops are not checked. The initial data message from the source is marked as exploratory and is sent to all neighbours for which it has matching gradients. After the initial exploratory data message, subsequent messages are sent only on reinforced paths.

When the sink node gets multiple neighbours in Figure 2.5(b), it chooses to receive subsequent data messages for the same interest from a preferred neighbour. Then the sink node reinforces its preferred upstream neighbour in turn. When interests fit gradients, paths of information flow are formed from multiple paths and then the best paths are reinforced. In this way, Directed Diffusion can prevent further flooding and reduce communication costs as well. Data is also aggregated on the way. Normally, the sink node will periodically refresh and re-send the interest when it starts to receive data from the source due to the non-reliability of interest propagation throughout the network [33].

The diffusion procedure discussed before is known as two-phase-pull diffusion. It is very suitable for applications expecting data from persistent queries (such as tracking tasks in a certain area for a fixed period time). When applied to the application of one-time queries, the two-phase-pull protocol is not appropriate as setting up gradients for queries which may use only once is costly. Two other protocols are proposed in the diffusion protocol family to cope with the one-time query scenario: one-phase push and one-phase pull [34]. The main difference between them can be summarized in Table 2.1.

Table 2.1: Comparison of interactions in diffusion algorithms

Protocols	Sink node	Source node	Application
Two Phase Pull	1.Interest Data 3.Positive reinforcement	2.Exploratory Data	Detection of tracked objects
One Phase Pull	1.Interest Data	2. Data	one-time query
Push	2.Positive reinforcement	1.Exploratory Data 3. Interest Data	Many sensor nodes interested in data but the frequency of triggers actually being sent is fairly rare.

2.2.4 GEAR (location based)

Since data queries often include geographic attributes, Yu et al. [35] suggested a protocol named GEAR (Geographic and Energy-Aware Routing) which makes use of the geography and information while disseminating queries to appropriate regions. In other words, GEAR extends diffusion when node location and geographic queries are present. The idea of this new protocol is to restrict the number of interests in Directed Diffusion by only considering certain region rather than sending the interests to the whole network. GEAR complements Directed Diffusion in this way and also saves more energy resource.

In GEAR, each node keeps an estimated cost and a learning cost of reaching the destination through its neighbours. The estimated cost is a combination of residual energy and distance to destination. The learned cost is a refinement of the estimated cost that accounts for routing around holes in the network. A hole occurs when a node does not have any closer neighbour to the target region than itself. If there are no holes, the estimated cost is equal to the learned cost. The learned cost is propagated one hop back every time when a packet reaches the destination so that route setup for next packet will be adjusted.

When added to one-phase or two-phase pull diffusion, GEAR subscribers actively send interests into the network. However, queries expressing interest in a region are sent toward to the region using greedy geographic routing (with support for routing around holes) [35]; flooding occurs only when interests reach the region rather than send throughout the whole network. Exploratory data is sent only on gradients setup by interests, so the limited dissemination of interests also reduces the cost of exploratory data.

For one-phase push diffusion, GEAR uses the same mechanism to send exploratory data messages containing a destination region towards that region. This avoids flooding by allowing data senders to push their information only to subscribers within the desired region, which in turn will send reinforcements resulting in future data messages following a single path to the subscriber.

Based on the directed diffusion algorithm, other extension protocols have been proposed, such as GPSR [36] which is one of the earlier works in geographic routing that uses planar graphs to solve the problem of holes, and so on.

2.2.5 Hierarchical routing LEACH protocols

The LEACH (Low energy adaptive clustering hierarchy) protocol proposed by Heinzelman et. al., is a typical cluster-based or so-called hierarchical routing protocol. Hierarchy routing is a mainly two layer routing where one layer is responsible for selecting the cluster-heads and the other layer is used for routing. In the hierarchical structure, the higher energy node acts as the cluster-head which can be used to process and send information while low energy nodes can be used to perform sensing operations in the proximity of the target. By way of the creation of clusters and assigning special asks to cluster heads, LEACH can greatly contribute to the system lifetime and energy efficiency. The LEACH protocol randomly selects a few sensor nodes as cluster-heads (CHs) and rotates this role to evenly distribute the energy load among the sensor nodes in a WSN. That procedure is the first phase of LEACH operation i.e. the setup phase. The following phase is the steady phase, where the cluster-head (CH) nodes compress data arriving from nodes that belong to the respective clusters. Then the CH sends an aggregated packet to the base station in order to reduce the amount of information transmitted to the base station. LEACH uses a TDMA/CDMA MAC to reduce inter-cluster and intra-cluster collisions. However, data collection is centralized and performed periodically [37].Therefore, the LEACH protocol is the most appropriate one for constant monitoring applications. A user may not need all the data immediately. Hence, periodic data transmissions (which may drain the limited energy of the sensor nodes) are unnecessary. After a given interval of time, a randomized rotation of the role of the CH is conducted so that uniform energy dissipation in the sensor network is obtained. Recent research reported in [29], only 5% of the nodes need to act as cluster heads so that LEACH protocol verifies its capability of increasing the network lifetime(as cluster heads consumed much more energy compared to normal nodes).

As a basic algorithm in cluster-based routing protocols, LEACH gets several extensions as well. LEACH with negotiation was proposed in [37], with the theme of preceding data transfers with high-level negotiation using meta-data descriptors as in the SPIN protocol discussed before. In [38], an enhancement over the LEACH protocol was proposed. The protocol, called Power-Efficient Gathering in Sensor Information Systems (PEGASIS), is an improvement of the LEACH protocol.

2.2.6 QoS based routing

QoS-based routing techniques apply to applications which require QoS. According to its natural characteristics, the WSN has to balance between energy consumption and data quality. QoS-aware protocols need to consider end-to-end delay requirements while setting up the paths in a sensor network. The following section presents the most significant protocols in this catalogue.

- SAR: Sequential Assignment Routing is the first protocol for sensor networks that includes the idea of QoS in its routing decision [1][39]. It is dependent on three elements: energy resources, QoS on each path, and the priority level of each packet. To form multiple paths from a source node, a tree rooted at the source node to the destination nodes (i.e., the set of base-stations (BSs)) is built. One of these paths is selected according to the energy resources and QoS. The paths of the tree are built while avoiding nodes with low energy or QoS guarantees. At the end of this process, each sensor node will be a part of multi-path tree. In this way, SAR is a table-driven multi-path protocol which aims to achieve energy efficiency and fault tolerance.

- Energy-Aware QoS Routing Protocol: A fairly new QoS aware protocol for sensor networks is proposed by Akkaya and Younis [40]. Specially, it is applied to application whose real time traffic is generated by imaging and video sensors. The proposed protocol extends the routing approach in [41] and finds a least cost and energy efficient path that meets certain end-to-end delay demands during the connection. The link cost used is a function that captures the node energy reserve, transmission energy, error rate and other communication parameters.

- SPEED: A QoS routing protocol for sensor networks which provides soft real-time end-to-end guarantees is described in [42]. The protocol requires each node to maintain information about its neighbours and uses geographic forwarding to find the paths. In addition, SPEED strives to ensure a certain speed for each packet in the network so that each application can estimate the end-to-end delay for the packets by dividing the distance to the BS by the speed of the packet before making the admission decision.

2.2.7 Conclusion and open issues

A summary on recent research of data routing in WSNs is presented in Table 2.2. It compares different routing protocols according to a list of metrics.

Recently, the issue of node mobility has drawn increasing research attention. Most of the current protocols assume that sensor nodes and the sink are stationary [29]. However, in the case of, for example, a battle environment, the sink and possibly sensor nodes need to be mobile in a certain region. The problem is that the frequent update of the position of the mobile nodes and the propagation of topology alteration messages through the whole network can take excessive

Table 2.2: Comparison of routing protocols in WSNs

Protocols	Query based	Power usage	Negotiation based	Position awareness	Mobility	Data aggregation	QoS
Flooding	No	Poor	No	No	Yes	No	Possible
Directed diffusion	Yes	Limited	Yes	No	Limited	Yes	No
SPIN	Yes	Limited	Yes	No	Possible	Yes	No
SPEED	Yes	N/A	No	No	No	No	Yes
SAR	No	No	No	Yes	No	No	Yes
LEACH	No	Good	No	No	Fixed BS	Yes	No
GEAR	No	Limited	No	No	No	No	No

amounts of energy. New protocols are needed in order to handle the mobility and topology alterations in such an energy constrained environment. Other future directions such as, secure routing [43][44], integration of sensor network with the wired network etc. focus on the common aim of prolonging the network lifetime whilst offering the appropriate type of service.

2.3 Summary

The novel characteristics of WSNs distinguishes them from other forms of wireless networks. Hence, many new algorithms have been proposed for the communication problem in WSNs. In this chapter, several most used MAC protocols and Routing protocols are discussed. These protocols have to be designed with concern for the inherent features of the WSN along with the application and architecture requirements. Several open issues for WSN communication protocols have been discussed as well.

With many different algorithms having been proposed, the decision to select the optimal set of protocols for a given task before a WSN is practically deployed becomes very important for a WSN designer. In next chapter, an overview of optimization procedures in WSNs is presented.

Chapter 3

Optimizations in Wireless Sensor Network Design

A WSN designer who takes into account all the design issues discussed in section 1.4, has to deal with more than one nonlinear objective function or design criteria [45]. From layer-wise aspect, the optimization goals of each layer for a WSN are summarized in figure 3.1.

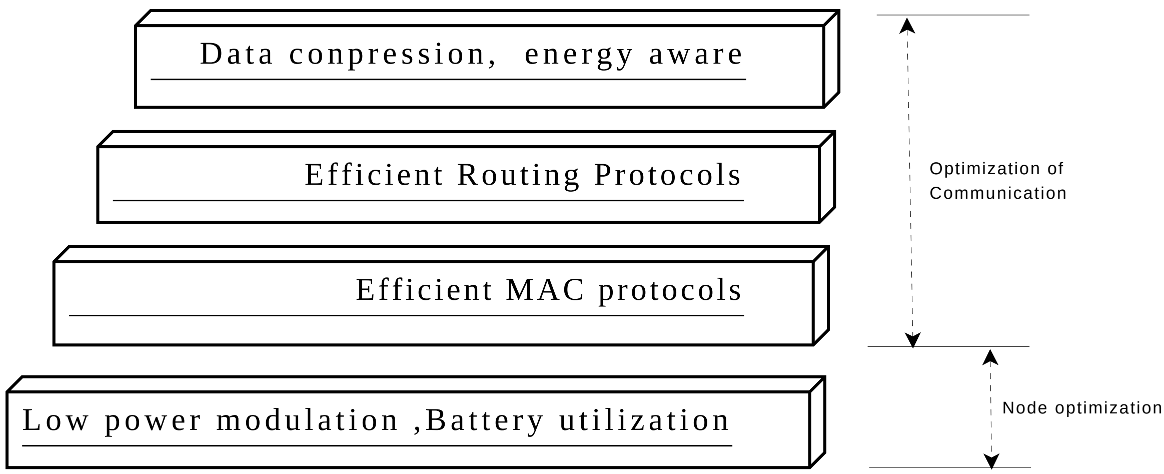


Figure 3.1: Optimization through different layers

In the application layer, the traffic load can be compressed to reduce the data size; other algorithms such as in-network data processing, have been developed to reduce energy consumption compared to transmitting all the raw data to the end node. The routing layer and MAC layer can be optimized by selecting appropriate protocols to gain efficiency. Node optimization can be achieved by improving battery utilization and implementing power-aware hardware design. Three different types of optimizations are classified, as the optimization of the communication layers, the node optimization and cross-layer optimization. In the following paragraghs, those three will be introduced and discussed.

3.1 Optimization of communication layers

Data traffic relayed in a wireless sensor network consumes a lot of energy. The optimization procedure in a WSN often aims at prolonging the network lifetime in the context of accomplishment of a certain task. The goals can be achieved by optimization in single or multiple layers in terms of improving performance in network scale, life-time and node capabilities. Different optimization approaches of each layer are listed in Table3.1.

Table 3.1: Optimization approaches in each layer

Layer	Network Scale	System life-time	Node capabilities
Application Layer	Data fusion and compression	Power aware mode control	Traffic detection and automatic mode decision
Network Layer	Efficient routing, efficient node recovery, node naming	Power-aware routing	Distributed storage
MAC layer	Contention control, resource distribution	Synchronized sleep,Transmission range control	Load-aware channel allocation
Physical	uUtra-wide band	Low power design, high efficient battery	Accessories(GPS)

The application layer is designed to deal with traffic generated by the sensor node. The way that data is manipulated by sensor nodes is a fundamental issue. Information fusion and compression techniques arise as a response to processing data gathered by sensor nodes [46]. By exploiting the connections among the available data, information fusion techniques can also reduce the amount of data traffic and filter noisy measurements such that the optimization of application layer is achieved.

Optimizations applied to routing and the MAC layer mainly focus on improving energy efficiency in recent research. C.Alippi proposed a routing optimization framework based on static/semi-static types of applications[47]. The optimal routing setup was achieved on the knowledge of the target application and the RF mapping of the environment. In [48], an Ant Colony Optimization technique was used to discover the shortest path from source nodes to the sink nodes while maintaining the network lifetime in maximum. Other optimizations in the routing layer use the efficient node recovery technique [49] and naming mechanism [50] to improve the efficiency as well as power consumption.

3.2 Sensor node optimization

From the hardware perspective, radio modulation, control unit and battery utilization can be optimized to achieve higher power efficiency. Four major energy-consuming parts of a sensor node are identified in the following paragraph. Related research in this area are discussed later.

1. The Radio Module: offering wireless communications with its neighbourhood and external world. Several factors have influence in radio module power consumption, such as, modulation scheme, data rate, transmission

power and duty cycle. In most radio modules, the idle state wastes as much energy as the reception state. Therefore, researchers tried to switch the radio modules off completely whenever no transmission or reception is scheduled. A transitory activity generated by working state changes leads to an important source of energy dissipation[51] as well.

2. Control unit: controlling the sensing parts and executing the communication protocols and the data processing algorithms.

3. Transducers: translates physical magnitudes into electric signals. They are classified into active and passive signals. Passive sensor nodes do not need power supply for measuring while active ones require an external energy.

4. Batteries: they play an important role for node lifetime. Many factors such as battery dimensions and type of electrode material, affect the battery performance dramatically. Because a sensor node has to be easily deployable and cheap, the size of a battery does limit the possibilities of increasing the battery energy supply.

Several hardware design algorithms have been developed to gain the optimal energy efficiency. Dynamic Power Management(DPM) algorithms are techniques which reduce system consumed energy by making components go to low power consumption selectively [52]. Algorithms such as Dynamic Voltage Scaling (DVS) [53]and Dynamic Frequency Scaling(DFS) [54][55], make the computational unit work faster when it has load. They optimize the energy consumption by varying a functional characteristic: voltage or frequency respectively.

Although these techniques can reduce power consumption for each node, the system latency is dramatically increased. The inefficiency of communication will lead to more energy consumption of the entire network. Operating system, application layer and communication protocols have to be designed to be power-aware to cooperate with hardware to gain the optimal network performance. An optimization procedure is needed to solve the conflicts and constraints that may exist among different layers responsbilities.

3.3 Cross-layer Optimization

In recent literature, cross-layer optimization has been frequently mentioned for efficiency improvement and power-saving. The work can be roughly classified into four categories according to their optimization goals. Firstly, relaxing the power constrains, in [56], Ganesan studied the necessity and possiblity of taking advantage of cross-layer design to improve the power efficiency in WSN. Secondly, improving system throughput. Theoretical analysis and possible approaches have been pointed out in [1] in order to solve the scalability problem. Fulfilling QoS requirements [57] is another cross-layer optimization goal for a WSN. There are also cross-layer optimizations aimed to improve the communication resource efficiency in [58].

Take power-aware design for instance, the power consumption according to the functioning of a sensor node can be divided into three domains: sensing, communication, and data processing. From the network management perspective, a power-saving routing protocol is required to optimize the power consumption of a WSN. In the routing layer, the protocol which selects the shortest path could be preferred. The smallest distance hops selected by the protocol could save transmission power in turn, compared with a single larger distance hop. However, the data transmission with more hops may result in a big contention possibility.

Hence, the MAC layer has to be optimized accordingly, otherwise the advantages of the routing design will counteracted by the increasing power consumption due to the increasing contention possibility.

3.4 Discussion

Researchers in the area have proposed several different approaches to optimizations of a wireless sensor network design. To meet different design criteria, related researches into optimization of wireless sensor network design can be grouped into three categories: optimization in the communication layers, node hardware optimization and cross-layer optimization.

However, most of the optimization procedures do not take into account the principles, characteristics and requirements of an application-specific WSN at the system level. In other words, the optimization approaches discussed in this section only optimize a certain objective or a few objectives. An optimization approach which is goal oriented, is strongly desired from the system view. This design optimization procedure has to be much more complex as many design criteria have to be optimized simultaneously. Thus, the focus of the problem is how to find a near-optimal non-dominated solution in a practically acceptable computational time.

As we discussed in Chapter 2, many communication protocols have been proposed for different applications. More protocols are being developed as WSNs apply to more real life scenarios. From the system architecture aspect, the decision process employed by a WSN designer presently to build a WSN is not a simple one, especially for a non-expert in WSNs. In this work, we focus on the design optimization of different sets of protocols to compare energy efficiency and trade it off with other user-defined requirements(non-protocol parameters, such as One-way Delay, Loss etc.).

3.5 Summary

In this chapter, different optimization methods for Wireless Sensor Network design are discussed. Most of these methods focus on optimizing a certain goal or several goals for a WSN. A new approach which view the design space at the system angle is required in this work. Taking the user-defined requirements as our starting point, we aim to help a non-expert in WSNs to select a proper set of protocols which provide the optimal performance of the entire network for a given application.

An optimized performance scheme should consider all the performance aspects to evaluate a WSN comprehensively under heterogeneous network configurations. Intelligent optimization techniques are an efficient way of solving the searching part of our system. In the following chapter, an overview of genetic algorithms is presented followed by relative applications of genetic algorithms in wireless sensor networks.

Chapter 4

Applications of Genetic Algorithms in WSNs

Genetic algorithms (GAs) have shown their capability for solving complex problems in many research areas including numerical optimization and business optimization problems. Genetic algorithms try to imitate natural evolution by assigning a fitness value to each solution of the problem and follow the principle of the survival of the fittest. There are also techniques such as simulated annealing, tabu search, etc. [59] in addition to GAs. However, the abilities of searching, fast convergence and fast evaluation distinguish GAs from other decision and optimization algorithms. In addition, the searing procedure in this work is in a range which grows, as more protocols are being developed and more system parameters should be optimized.

In this section, an overview of genetic algorithms is introduced followed by a detailed study of GA implementation. Several successful GA applications to WSNs are also presented.

4.1 Genetic Algorithm Overview

Genetic algorithm (GA) is a stochastic search technique that mimics the natural evolution proposed by Charles Darwin in 1858. GA has been successfully applied to a wide range of combination problems. It is modeled largely on the principles of the evolution via natural selection [60] , employing a population of individuals that carry on the selection procedure in terms of variation-inducing operators such as mutation and recombination (crossover). A fitness function is used to evaluate individuals, and reproductive success varies with fitness.

GA maintains a population of n chromosomes (solutions) with associated fitness values. The whole process is described as follows: parents are selected to mate, on the basis of their fitness, producing offspring via a reproductive plan. Consequently highly fit solutions are given more opportunities to reproduce, so that offspring inherit characteristics from each parent. As parents mate and produce offspring, room must be made for the new arrivals since the population is kept at a static size. Individuals in the population die and then, are replaced by the new solutions, eventually creating a new generation once all mating opportunities in the old population have been exhausted [61]. In this way, it is expected that over successive generations, better solutions will prosper while the least fit solutions die out.

New generations of solutions are produced containing, on average, better genes than a typical solution in a previous generation. Each successive generation will contain more good ‘partial solutions’ than previous generations. Eventually, once the population has converged and is not producing offspring noticeably different from those in previous generations, the algorithm itself is said to have converged to a set of solutions to the problem at hand.

4.2 Genetic Algorithm Implementation

After an initial population is randomly generated, the algorithm evolves through three operators:

1. Selection: gives preference to better individuals, allowing them to pass on their genes to the next generation. The goodness of each individual depends on its fitness. Fitness may be determined by an objective function or by a subjective judgement.

2. Crossover : represents mating between individuals; Two individuals are chosen from the population using the selection operator. A crossover site along the bit strings is randomly chosen. The value of each string is exchanged up to this point.The two new offspring created from this mating are put into the next generation of the population. By recombining portions of good individuals, this process is likely to create even better individuals.

3. Mutation: introduces random modifications. The purpose of mutation is to maintain diversity within the population and prevent premature convergence. With some low probability, a portion of the new individuals will have some of their bits flipped. Mutation and selection (without crossover) create a parallel, noise-tolerant, hill-climbing algorithms [62].

The pseudo code of genetic algorithms is shown below:

In the following paragraphs, the details of three operators are introduced and discussed.

Algorithm 1 Pseudo algorithms of GAs

Step1: Randomly generate an initial population M(0);
Step2: Compute and save the fitness f(m) for each individual m in the current population M(t);
Step3: Define selection probabilities p(m) for each individual m in M(t) so that p(m) is proportional to f(m);
Step4: Generate M(t+1) by probabilistically selecting individuals from M(t) to produce offspring via genetic operators ;
Step5: repeat step 2 until satisfying solution is obtained;

4.2.1 Selection Operator

Proportionate reproduction and tournament selection are considered in this work. Proportionate reproduction describes a group of selection schemes that choose individuals according to their objective function values f. In these schemes, the probability of selection p of an individual from the ith class in the t th generation is calculated as

$$p_{i,t} = \frac{f_i}{\sum_{j=1}^{k} m_{j,t} f_j} \text{ [63]} \tag{4.1}$$

where k classes exist and the total number of individuals sums to n.

Various methods have been suggested for sampling this probability distribution, including Monte Carlo or roulette wheel selection [64], stochastic remainder selection [65], and stochastic universal selection [66]. In this work, roulette wheel selection and stochastic remainder selection are implemented.

Roulette wheel selection, also named stochastic sampling, selects parents according to a spin of a weighted roulette wheel. The roulette wheel is weighted according to the chromosome fitness values . A high-fitness individual will have more area assigned to it on the wheel. Hence, it has a higher probability of ending up as the choice when the biased roulette wheel is spun. A random number between 0 and 1 is generated. In this work, the Subtractive Method [67] is used to fetch this single random number. Two individuals are then chosen randomly based on these probabilities and produce offspring. The pseudo code of Roulette wheel selection is shown below:

Algorithm 2 Pseudo algorithms of Roulette wheel selection

```
for all the members of population do
    sum += fitness of this individual
end for
for all members of population do
    probability = sum of probabilities + (fitness / sum)
    sum of probabilities += probability
end for
loop
    until new population is full
    Do this twice
    number = Random between 0 and 1
    for all members of population do
        if number > probability but less than next probability then
            you have been selected
        end if
    end for
    create offspring
end loop
```

A form of tournament selection[68] [69] chooses some number of individuals randomly from a population (with or without replacement), selects the best individual from this group for further genetic processing, and repeats as often as desired (usually until the mating pool is filled). Tournaments are often held between pairs of individuals (tournament size $s = 2$), although larger tournaments can be used and may be analyzed. When tournament size $s = 1$ selection is equivalent to random selection. The chosen individual can be removed from the population that the selection is made from if desired, otherwise individuals can be selected more than once for the next generation[70].

Algorithm 3 Pseudo algorithms of tournament selection

choose k individuals from the population at random; {k is the tournament size}

loop
  until new population is full
  choose the best individual from tournament with probability p
  choose the second best individual with probability $p * (1 - p)$
  choose the second best individual with probability $p * (p * (1 - p))$
  and so on...
end loop

4.2.2 Crossover Operator

Crossover is a genetic operator used to vary the programming of a chromosome or chromosomes from one generation to the next. It is analogous to reproduction and biological crossover, upon which genetic algorithms are based[71]. Many crossover techniques exist such as one-point crossover and two-point crossover etc.. Take one-point crossover for instance, a single crossover point on both parents' chromosomes strings is selected. All data beyond that point in either Chromosomes string is swapped between the two parent Chromosomes. The resulting chromosomes are the children. This has the disadvantage though that certain schema cannot be passed from one generation to the next as none of the offspring can have the start and end of a parents gene string.

Algorithm 4 Pseudo algorithms of one-point crossover operation

find a crossover point according to the Crossover probablity.
for $i = 0$ to crossover point **do**
  child A gene[i] = parent A gene[i]
  child B gene[i] = parent B gene[i]
end for
for $i = crossoverpoint$ to chromosome length **do**
  child A gene[i] = parent B gene[i]
  child B gene[i] = parent A gene[i]
end for
return children

For ordered chromosomes, a direct swap may not be possible depending on how the chromosome represents the solution. One such case is when the chromosome is an ordered list, such as an ordered list of the cities to be travelled for the traveling salesman problem[72]. A crossover point is selected on the parent strings. Since the chromosome is an ordered list, a direct swap would introduce duplicates and remove necessary candidates from the list. Instead, the chromosome up to the crossover point is retained for each parent. The information after the crossover point is ordered as it is ordered in the other parent. For example, if two parents are ABCDEFGHI and IGAHFDBEC and crossover point is after the fourth character, then the resulting children would be ABCDIGHFE and IGAHBCDEF.

4.2.3 Mutation Operator

Similar to crossover, mutation is analogous to biological mutation and used to maintain genetic diversity from one generation to the next. The aim of mutation in GAs is to allow the algorithm to avoid local minima by preventing the population of chromosomes from becoming too similar to each other, thus slowing or even stopping evolution. It also explains the fact that most GA systems avoid only taking the fittest of the population in generating the next but rather a random (or semi-random) selection with a weighting toward those that are fitter.

Also, the mutation rate can have a great effect on a population. If the rate is too high, the mutation procedure are disrupted and convergence of the population becomes very hard, whereas if it is too low, the population could converge too quickly and ignore possible optima. Mutation is often done after recombination following the biological analogy where after recombination damage can occur causing the genes to mutate.

Mutation techniques depend on how a chromosome represents a solution. For a binary representation chromosome, a mutation operator normally involves a probability that an arbitrary bit in a genetic sequence will be changed from its original state. A common method of implementing the mutation operator involves generating a random variable for each bit in a sequence. This random variable tells whether or not a particular bit will be modified. Below is pseudocode for a simple mutation operator:

Algorithm 5 Pseudo algorithms of Mutation operation

```
for i = 1 to chromosome length do
  generate a random number N_r
  if N_r ≤ mutationprobability then
    child A gene[i] = random value
    child B gene[i] = random value
  end if
end for
```

4.3 Genetic Algorithms applied to WSN design

GAs have been already used in several optimization procedures in wireless sensor networks field. The successful application of GAs in a WSN design in [73] led to the development of several other GA-based approaches in WSN design [74][75] [76]. Qiu etc.[77] proposed an approach that by using distributed a GA, allocates different detection methods to different sensor nodes. As a result, the required detection probability can be achieved while the network lifetime is maximized.

K.Ferentinos etc. in [78] suggested a GA-based methodology for adaptive WSN design so that the status of each sensor node, the appropriate clusterheads and the distance between a pair of nodes can be optimized.

Also, in [79], Multi-Objective Genetic Algorithm (MOGA) was used as an optimizer of a wireless sensor network for monitoring a critical facility in a hostile region. The work examined the optimal placement of nodes for a Wireless Sensor Network (WSN) in three scenarios. MOGA shows its flexibility and possible application to more complex mission scenarios with multiple and diverse targets of observation.

The application of a GA into a problem usually consists of three basic steps: the problem representation (the encoding mechanism of the candidate solutions into genotypes that a GA could manipulate and evolve), the formulation of the fitness function that gives to each individual (possible network solutions) a measure of performance, and finally the choice of the genetic operators and the selection mechanism used. The methodology of GA applied to our problem is discussed in detail in the next chapter.

Chapter 5

Methodology

As we discussed in chapter 2, the architecture of WSNs directly depend on different application requirements in terms of latency, accuracy and network lifetimes. Many new algorithms have been proposed for WSNs. The design of WSN protocols is based on optimizing certain metrics, network's lifetime for instance, by managing its resources in an efficient manner given the condition in which the assigned task must be achieved. It leads to the lack of availability of a common protocol that will suit all applications. The decision to select the optimal set of protocols for a given task before a WSN is practically deployed becomes very important for a WSN designer.

In this chapter, motivation of the work is developed and extended followed by an introduction to the proposed wireless sensor network performance model. The methodology of applying a GA into this work is also presented.

5.1 Motivation

Communication between each node in a WSN (due to its inherent characteristics) distinguishes WSNs from other wireless networks. Hence, many new protocols have been proposed for the communication problems in WSNs. These protocols have to be designed with concern for these inherent features along with the application and architectural requirements. Therefore, the selection of a good set of protocols for a given task before a WSN's practical deployment is an important issue.

With the proper set of protocols selected, the number of nodes deployed in a fixed area draws our attention. Usually, nodes density ranges from few sensor nodes to hundreds in a fixed area. When a large number of nodes are deployed, can users fully utilize the high density nature of the WSN? Can they still maintain elementary coverage in the target area in the case that some nodes fail (Note that failure of several sensor nodes may not harm the overall functioning of a WSN)?

Generally, the design procedure follows the order that people in this field firstly put more and more effort into inventing new protocols and new applications; then the solutions are built, tested and evaluated either by simulation or testbeds; even sometimes an actual system has to be deployed so that researchers can learn by empirical evidence. A more scientific analysis procedure is ideally required before a WSN is practically deployed.

It is accepted that the current designers in the area are mainly experts in wireless sensor networking and hardware who could perceive the communication behaviour between each nodes at the bit level. As WSNs immerse deeper into people's lives, they must begin to include less specialized users. In such cases, a system which can offer optimal solutions based on expert knowledge and can be easily used is strongly desired to support a wider audience of users.

We propose a GA-based design tool which aids sensor network designers in system performance tuning before a network is practically deployed. The main aim of the work of this thesis is to develop a system which could select the proper set of protocols out of hundreds of potential solutions and optimize the number of nodes in a fixed area given a certain application scenario. This offline procedure is to be done by identifying an acceptable performance of a WSN with certain tasks under different sets of protocol configurations. In the following sections, two major issues , WSN performance modeling and the methodology of applying a GA into this problem space , are discussed respectively.

5.2 Wireless Sensor Network Performance Modeling

An important issue of this work is to find a selection of appropriate algorithms for WSNs which provides an acceptable performance of the entire network. An offline procedure is to be achieved by identifying an acceptable performance of a WSN under different configurations for a certain scenario.

A WSN is tailored for a specific application. Correspondingly, each application has its own unique requirements for the WSN. The performance of a WSN, due to the resource constraints and application specific characteristics, cannot be assesed by simply using any single metric alone. As a result, many metrics are interrelated. Assessment of a WSN depends on a series of performing metrics. The question of selecting appropriate metrics to evaluate a WSN is crucial to our work. It will be discussed after an identification of WSN application classfication.

5.2.1 Sensor network application classification

Several most significant WSN applications are investigated in this work. It is believed that the majority of wireless sensor network deployments will fall into one of these three class templates: periodic measurements, event detection and sensor node tracking.

5.2.1.1 periodic measurement application

A periodic measurement application is one where a researcher wants to collect several sensor readings from a set of points in an environment over a period of time in order to detect trends and interdependencies. The researcher would want to collect data from hundreds of points spread throughout a given area and then analyze the data offline [80]. Long-term and seasonal trends would be identified by collecting data over several months or years. For the data to be meaningful, it would have to be collected at regular intervals and the nodes would remain at known locations.

At the network level, the environmental data collection application is characterized by having a large number of nodes continually sensing and transmitting data back to a set of base stations that store the data using traditional methods. These networks generally require very low data rates and extremely long lifetimes. In a typical periodic measurement scenario, the nodes will be evenly distributed over an outdoor environment. This distance between adjacent nodes will be minimal yet the distance across the entire network will be significant.

After deployment, the nodes must first discover the topology of the network and estimate optimal routing strategies [81]. The routing strategy can then be used to route data to a central collection point. In environmental monitoring applications, it is not essential that the nodes develop the optimal routing strategies on their own. Instead, it may be possible to calculate the optimal routing topology outside of the network and then communicate the necessary information to the nodes as required [82]. This is possible because the physical topology of the network is relatively constant. While the periodic sleeping mechanism of nodes may cause connectivity between two nodes to be intermittent, the overall topology of the network will be relatively stable.

Once the network is configured, each node periodically samples its sensors and transmits its data back to the base station. For many scenarios, the interval between these transmissions can be in the order of minutes. Typical reporting periods are expected to be between 1 and 15 minutes; while it is possible for networks to have significantly higher reporting rates. The typical environment parameters being monitored, such as temperature, light intensity, and humidity, do not change quickly enough to require higher reporting rates.

In addition to large sample intervals, periodic measurement applications do not have strict latency requirements. Data samples can be delayed inside the network for moderate periods of time without significantly affecting application performance. In general the data is collected for future analysis, not for real-time operation.

In order to meet lifetime requirements, each communication event must be precisely scheduled. The sensor nodes will remain dormant for a majority of the time; they will only wake to transmit or receive data. If the precise schedule is not met, the communication events will fail.

As the network ages, it is expected that nodes will fail over time. Periodically the network will have to reconfigure to handle node/link failure or to redistribute network load. Additionally, as the researchers learn more about the environment they study, they may want to go in and insert additional sensing points. In both cases, the reconfigurations are relatively infrequent and will not represent a significant amount of the overall system energy usage.

The most important characteristics of the periodic measurement applications requirements are long lifetime, precise synchronization, low data rates and relatively static topologies. Additionally it is not essential that the data be transmitted in real-time back to the central collection point. The data transmissions can be delayed inside the network if required.

5.2.1.2 Event detecting

WSNs for event detecting applications are composed of nodes that are placed at fixed locations throughout an environment that continually monitor one or more sensors to detect an anomaly or emergency. A key difference between event detecting and periodic measuring is that event detecting networks are not actually collecting any data. This has a significant impact on the optimal network architecture. Each node has to frequently check the status of its sensors but it only has to transmit a data report when there is a violation of a threshold. The immediate and reliable communication of alarm messages is the primary system requirement. These are “report by exception ”networks [82].

Additionally, it is essential that it is confirmed that each node is still present and functioning. If a node were to be disabled or fail, it would represent a security violation that should be reported. For security monitoring applications, the network must be configured so that nodes are responsible for confirming the status of each other. One approach is to have each node assigned to a peer that will report if a node is not functioning. The optimal topology of a security monitoring network will look quite different from that of a data collection network.

It is accepted that, for security systems today, each sensor should be checked approximately periodically. Combined with the ability to evenly distribute the load of checking nodes, the energy cost of performing this check becomes minimal. A majority of the energy consumption in an event detecting network is spent on meeting the strict latency requirements associated with the signalling the alarm when a security violation occurs.

Once detected, a security violation must be communicated to the base station immediately. The latency of the data communication across the network to the base station has a critical impact on application performance. Users demand that alarm situations must be reported within a reasonable short period of detection. This means that network nodes must be able to respond quickly to requests from their neighbours to forward data.

In event detecting networks, reducing the latency of an alarm transmission is significantly more important than reducing the energy cost of the transmissions. This is because alarm events are expected to be rare. In a fire security system, alarms would almost never be signalled. In the event that one does occur, a significant amount of energy could be dedicated to the transmission. Reducing the transmission latency leads to higher energy consumption because routing nodes must monitor the radio channel more frequently.

In event detecting networks, a vast majority of the energy will be spend on confirming the functionality of neighbouring nodes and in being prepared to instantly forward alarm announcements. Actual data transmission will consume a small fraction of the network energy.

5.2.1.3 Node tracking scenarios

A third scenario commonly discussed for WSNs is the tracking of a tagged object through a region of space monitored by a sensor network. There are many situations where one would like to track the location of valuable assets or personnel. Current inventory control systems are a good example. These systems attempt to track objects by recording the last checkpoint that an object passed through. However, with these systems it is not possible to determine the current location of an object. In typical work environments, it is impractical to expect objects to be continually passed through checkpoints.

With wireless sensor networks, objects can be tracked by simply tagging them with a small sensor node. The sensor node will be tracked as it moves through a field of sensor nodes that are deployed in the environment at known locations. Instead of sensing environmental data, these nodes will be deployed to sense the messages of the nodes attached to various objects. The nodes can be used as active tags that announce the presence of a device. A database can be used to record the location of tracked objects relative to the set of nodes at known locations. With this system, it becomes possible to ask where an object is currently, not simply where it was last scanned [83].

Unlike periodic measuring or event detecting networks, node tracking applications will continually have topology changes as nodes move through the network. While the connectivity between the nodes at fixed locations will remain relatively stable, the connectivity to mobile nodes will be continually changing [82]. Additionally the set of nodes being tracked will continually change as objects enter and leave the system. It is essential that the network be able to efficiently detect the presence of new nodes that enter the network.

5.2.2 System evaluation metrics

Having investigated the most significant application types, the key evaluation metrics for wireless sensor networks can be summarised into lifetime, coverage, cost and ease of deployment, response time, temporal accuracy, security, and effective sample rate.

In traditional networks, key indicators such as delay/jitter, packet losses and bandwidth utilization can be used to evaluate the network performance. The importance of these indicator changes depends on the application's tolerance and the level of Quality of Service(QoS) required at the destination [84]. However, for a wireless sensor network, the performance of the WSN due to the resource constraints and application specific characteristic, can not be assesed by simply using any of the metrics alone. As a result, many of these metrics are interrelated. Often it may be necessary to decrease performance in one metric, such as delay, in order to increase another, such as lifetime. Taken together, this set of metrics form a multidimensional space that can be used to describe the capabilities of a wireless sensor network. A list of challenges regarding WSN performance evaluation is displayed below:

- Limited energy and communication resources.
- The unavailability of dominant protocols or algorithms that can be used in all applications.
- Application-tailored network.
- Node identification may not be available.
- The direct interaction of nodes with the environment increases the possibility of node failure.

To address those challenges, we have to keep in mind the high-level objectives of the network deployment, the intended usage of the network, and the key advantages of wireless sensor networks over existing technologies. Although different applications may emphasize different metrics when assessing the performance of a WSN, some fundamental metrics such as network lifetime, coverage, end-to-end delay, energy consumption, temporal accuracy etc., which most applications would take account into, are considered in this work. Their importance is discussed below:

1. Network lifetime

   Considering the constrained energy resources, the network lifetime has always been mentioned as a metric of energy efficiency when evaluating the performance of a WSN. The definition of network lifetime may change according to the requirements of different applications.

   - Time to first node death :when the first node in the network runs out of energy
   - Time to half nodes dead: When 50% of the nodes run out the energy and stop operating
   - Time to partition: When the first partitioning of the network in two disconnected parts occurs. This may happen as early as the death of the first node or very late if the network topology is robust enough.
   - Time to failure of the first event notification: when the network can first not report any events due to fact that the responsible node is dead or a partition between source and sink has occurred.

2. End-to-end delay

Some applications such as event detecting still value end-to-end delay as the most important metric for a WSN. This type of application always has a notion of urgent events, such as reporting an emergency occurrence, which needs the WSN to respond as soon as possible. Under this circumstance, delay is the most significant metric to evaluate the network. While, for periodic measurement applications (for example, long term habitat monitoring), data delay is no longer the major metric concerned, instead, the energy consumption is more crucial.

3. Temporal accuracy

   In tracking applications, information from multiple nodes must be cross-correlated in time in order to determine the nature of phenomenon being measured. Packet delivery loss is a key metric to represent the system performance in terms of temporal accuracy. Data packets generated by the sensing source nodes which fail to be delivered to the sink are called packet delivery losses. WSNs are a type of best-effort network as well as ADHOC networks. For instance, applications of objective tracking are concerned about the loss very much, due to the important information carried in each message propagated in the network. For applications requiring periodic reports, the number of loss messages should be as small as possible to fulfill the task.

4. Energy cost per Useful Bit

   The Energy-per-Useful-Bit(EPUB) metric was first proposed by [85] to evaluate the performance of different physical layers. It represents the energy consumption of delivering every useful bit. The function introduced in [86] includes the power consumption of both the transmitting and receiving modes. Assuming a sensor node transmits M packets, each of length L bits, to another node, the total energy cost of both nodes after the successful transmission can be presented by the equation below:

$$E_{total} = \left(T_{sw} + \frac{M}{R * P_p} * L\right) * P_{tx} + \left(\frac{M}{R * P_p} * L\right) * P_{pa} + \left(T_{sw} + \frac{M}{R * P_p} * L\right) * P_{rx} \text{ [86]} \tag{5.1}$$

where P_p is the probability of packet received successfully. T_{sw} represents the state swith time of the transceiver. P_{tx},P_{rx},P_{pa} are the transmission power, receiver power and propagation power respectively. Valued by energy per useful bit(EPUB), the above equation is modified as:

$$E_{epub} = \frac{E_{total}}{M * L} \text{ [87]} \tag{5.2}$$

It is a commonly used metric to assess the physical layer performance, for the future extension of a proposed system, EPUB is selected as one of the measuring metrics.

5. Coverage

Coverage is an important evaluation metric for a wireless network. It is always advantageous to have the ability to deploy a network over a larger physical area. This can significantly increase a systems value to the end user. Multi-hop communication techniques can extend the coverage of the network well beyond the range of the radio technology alone. In theory they have the ability to extend network range indefinitely. However, for a given transmission range, multi-hop networking protocols increase the power consumption of the nodes, which may decrease the network lifetime. Additionally, they require a minimal node density, which may increase the deployment cost.

In our work, a list of metrics is measured through simulations. They are shown in Table 5.1:

Table 5.1: Metrics used when assessing the performance of a WSN

Objectives	Optimization parameters	Performance measures
M1	One-way delay	OWD
M2	Loss	LSS
M3	Time when first node died	TFD
M4	Time when half nodes died	THD
M5	Energy consumption per useful bit	EPUB

On the basis of the listed metrics, a performance function of WSNs is derived in the next section to represent a comprehensive assessment of a WSN performance under different configurations.

5.2.3 Performance function

Heterogeneous network architecture solutions directly affect the network performance in terms of network lifetime, energy consumption, delay,loss and etc.. Due to the application-specific nature of WSNs, a performance function which includes a user defined weight vector is derived in this work to evaluate the performance of a WSN under different configurations.

$$f_{performance}(X) = W * M(X) \tag{5.3}$$

Where $f_{performance}$ is the performance function of each solution of the WSN and vector X is the set of configuration attributes. Vector W is a series of user-defined weight coefficients i.e. $W = [w_{OWD}, w_{LSS}, w_{TFD}, w_{THD}, w_{EPUB}, \cdots]$ which represents the different requirements of heterogeneous application scenarios in terms of marking the importance of each selected metric. Correspondingly, Vector $M(X)$ represents a series metrics measured from the simulation under certain network configuration X.

Then the performance function can be written as :

$$\begin{aligned} f_{performance}(X) &= \sum_{i=1}^{5} w_i * M_i \\ &= w_{OWD} * OWD(X) + w_{LSS} * LSS(X) + w_{TFD} * TFD(X) \\ &\quad + w_{THD} * THD(X) + w_{EPUB} * EPUB(X) \end{aligned} \tag{5.4}$$

Where $OWD(X)$ represents the one-way delay performance of the WSN with a set of configuration parameters X;

$LSS(X)$ is the loss performance of the WSN with a set of configuration parameters X;

$TFD(X)$, $THD(X)$ stand for the network lifetime performance of the WSN with a set of configuration parameters X;

$EPUB(X)$ iterates the energy consumption performance of the WSN with a set of configuration parameters X.

Delay performance metric $OWD(X)$ is defined as the actual one-way delay. The value of one-way delay can be attained by simulations.Loss performance metric $LSS(X)$ is also specified by the ratio of lost packets to the total sending packets in the WSN.

$$LSS(X) = \frac{P_{loss}}{P_{total}} \tag{5.5}$$

While network lifetime performance metrics $TFD\left(X\right)$, $THD\left(X\right)$ are defined as

$$TFD\left(X\right) = \frac{T_{first-node-died}}{T_{total}} \tag{5.6}$$

$$THD\left(X\right) = \frac{T_{half-nodes-died}}{T_{total}} \tag{5.7}$$

where $T_{first-node-died}$,$T_{half-nodes-died}$ can be easily get from the simulations.

Energy consumption performance metric $EPUB\left(X\right)$ represents the practical energy consumption per useful bit.

All these metrics can be obtained by simulation. In the next section, an overview of common used simulation tools for wireless sensor networks is given, followed by a reasoning of simulation tool choice.

5.2.4 Selection of a Simulator for WSNs

The computer simulation method efficiently explores the behaviour of a WSN. It contains a rich infrastructure for developing new protocols, offers opportunities to study a number of protocol interactions in a controlled environment. However, in order to assess a WSN at the system level, communication, collaboration modules and accurate sensing have to be considered. These factors impose more requirements for WSN simulation tools.

Many research groups have developed software simulators which are able to simulator a WSN along with its unique properties. Some of the simulators may provide a connection interface with a real WSN in order to allow the developers to enrich the practical network while enjoying the advantages of hybrid functionality. There also some simulators designed for traditional network which can be used for WSN by adding new modules.

Unfortunately, no one-for-all simulator is capable of precisely emulating all aspects of WSNs. This is because a WSN is an application-specific network which requires on-demand design specifications. A list of commonly used simulators is shown in Table 5.2, each of them has its area expertise in which it excels[88][89].

Table 5.2: Wireless sensor networks simulation software

Simulator	Simulation model	Languages	Description
NS2[90]	ISO/OSI	OTCL, C++(Object-oriented)	Include huge number of protocols, traffic generators and tools to simulate TCP, routing, and multicast protocols over wired and wireless.
NRLs sensor extension to NS-2	ISO/OS	OTCL, C++(Object-oriented)	modeling the presence of phenomena transmitted through a designated channel in NS2[91].
TOSSIM	At bit level	NesC[92]	Simulates TinyOS motes
SENSE	ISO/OSI	C++ (component-port model)	Offers different battery models, simple network and application layers, and a IEEE 802.11 implementation.
GloMoSiM	ISO/OSI	C/Parsec	Standard API used between the different simulation layers. The simulation is built on top of Parsec
OpNet	ISO/OSI	C/C++	Provides a simulation language with network libraries
Matlab	-	M-code	Numerical computing enviroment. Allows easy matrix manipulating,implementation of algorithms etc..

The selected WSN simulator has to be able to simulate a WSN at system level with configuration parameters such as different protocols in each layer or node density. Unfortunately, none of the above mentioned simulators meets the requirement perfectly. NS2 lacks the capability of simulating application-layer behaviour; while its extension NRL is poorly maintained by its developers, more importantly, NRL can only work with NS version 2.26, any new version of NS2 with more WSN features is incompatible with NRL.

TOSSIM was tested at the early stage of the work. By only replacing a few low-level TinyOS components that touch the hardware, it can capture node behaviour at bit level, allowing a wide range of experimentation[93]. However, TOSSIM compiles unchanged TinyOS applications directly into its framework, which result in the absence of protocol-based analysis.

NS2 was chosen as the simulation tool for this work. The NS-2 simulation environment[90] offers great flexibility in investigating the characteristics of sensor networks because it already contains flexible models for energy constrained wireless ad-hoc networks. In the NS-2 environment, a sensor network can be built with many of the same set of protocols and characteristics as those available in the real world. The mobile networking environment in NS-2 includes support for each of the paradigms and protocols shown in Figure 5.1 .

The limited number of WSN protocol modules in NS2 presents a major challenge for the work. Fortunately, many significant protocol modules are introduced in NS codes. With its open-source attribute, NS2 can be easily extended by adding more protocol modules.

5.3 GA Methodology

The application of a GA into a problem usually consists of three basic steps: the problem representation (the encoding mechanism of the candidate solutions into genotypes that a GA could manipulate and evolve), the formulation of the fitness function that gives to each individual (possible network solutions) a measure of

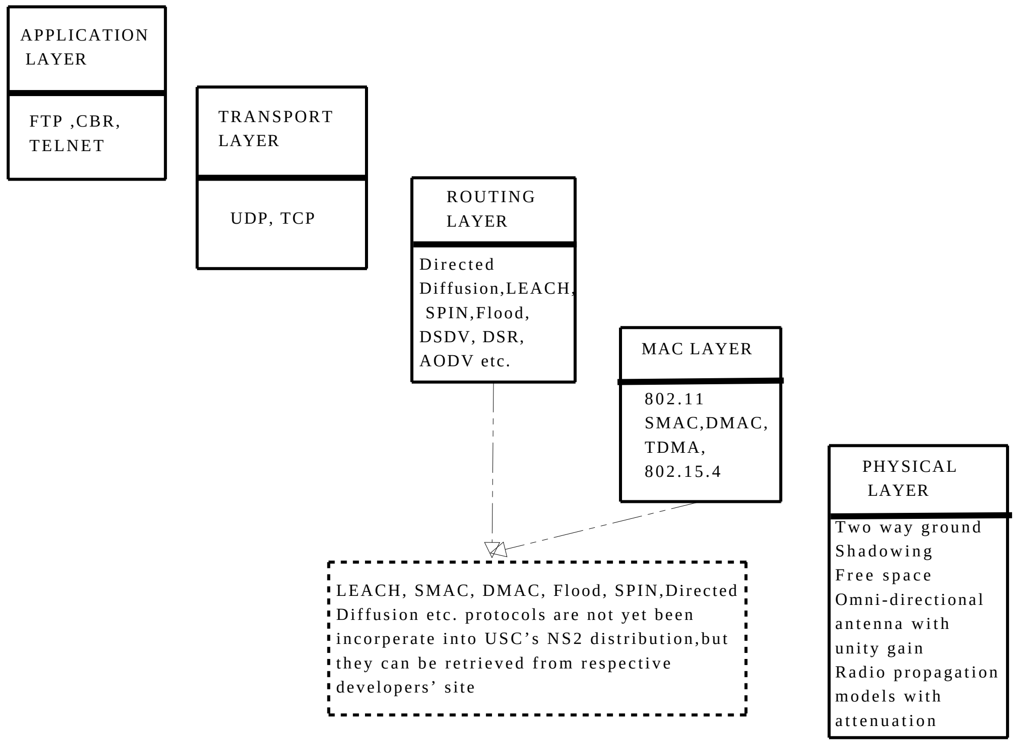


Figure 5.1: Available protocols for a WSN in NS2.31

performance, and finally the choice of the genetic operators and the selection mechanism used. These are discussed in detail in the following sections.

5.3.1 WSN representation

A genetic algorithm in our problem space is used to select the optimal set of protocols out of hundreds of possible solutions and the number of nodes to be deployed in a fixed area for a certain application scenario. In other words, each individual in a GA population represents a WSN solution. It consists of the networking layer protocols, medium access layer protocols and other potential configuration parameters.

An example of an individual chromosome structure is shown in Figure 5.2 . Three concatenated strings of binary integers whose length is defined by the user represent the network layer protocols, the medium access layer protocols and the number of nodes respectively. The length of each string can be adjusted when the number of choices increases. A reserved field is designed to accommodate other configuration parameters. In our work, the reserved field is used to represent the number of nodes involved in an application. Due to the difference in scale of the number of choices, the length of the reserved field is different from the others.

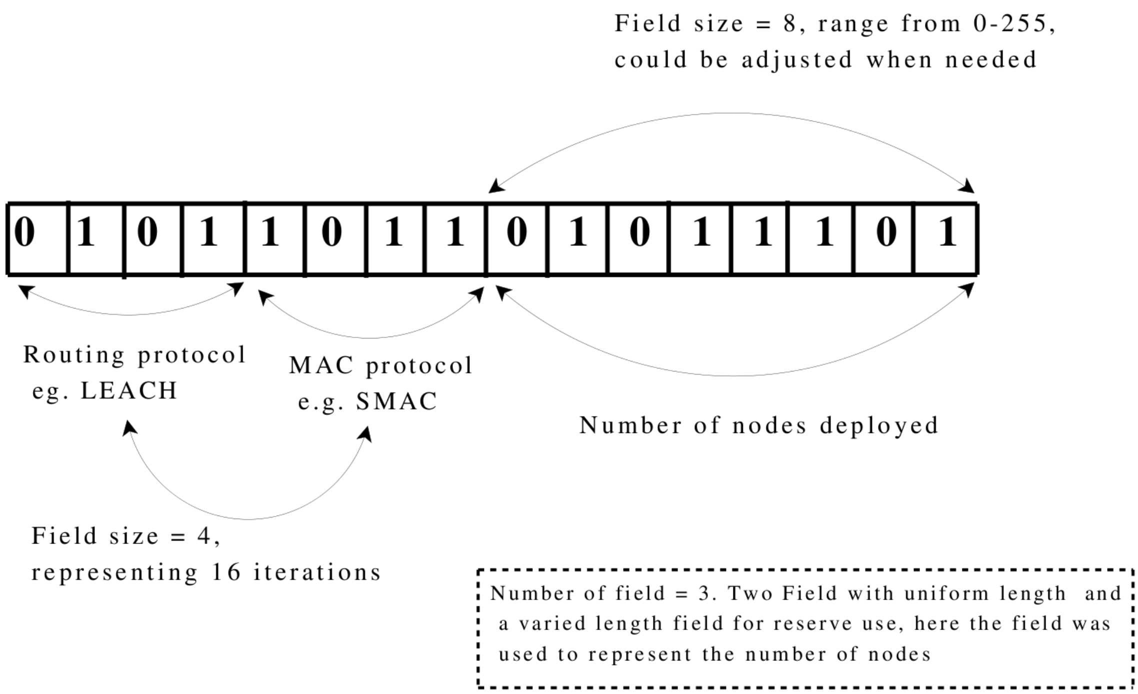


Figure 5.2: Binary representation of an example WSN solution

Furthermore, a more general individual structure can be developed by identifying the number of the fields and the size of each field. Each field can represent one configuration attribute. When the designer deliberates other configuration attributes, the representation of each individual can be slightly changed by varying the number of the fields and the size of each field.

5.3.2 Fitness function

The fitness function in a GA application for the work has to offer an informed measure for the performance of a WSN for a certain application scenario. It is maximized by the GA system in the process of evolutionary selection. With the performance function derived in section 5.2.3, the issue of the importance of each measure from different performance aspects (so that all the parameters can take an equal importance) has to be addressed. The significance of each performance measure is defined by setting a set of appropriate coefficients $\{A|\alpha_i : i = 1, 2, , 5\}$. The final form of the fitness function of a configuration of a WSN is given by equation 5.8 and 5.9:

$$f_{fitness}\left(X\right) = \frac{1}{A \times f_{performance}\left(X\right)} \tag{5.8}$$

$$f_{fitness}(X) = \frac{1}{\sum_{i=1}^{5} \alpha_i * w_i * M_i} \tag{5.9}$$

The use of the importance coefficients is meant to eliminate differences in the scale of different performance metrics. This can be seen as a normalization procedure for different performance metric data. It is clear shown that the list of metrics mentioned in Table 5.1 are of different range: M_{OWD} is in the ranges from 0.01 to 100, M_{LSS}, M_{TFD} and M_{THD} are in the range of 0 to1; M_{EPUB} is in the range of less than 1 to more than 100. For example, the value of EPUB is normally in the order of 1, while the value of the loss metric is in the order of 0.01. Without importance coefficients, those two values can not take equal importance in the fitness function. Therefore, with the importance coefficients multiplied by each metric, the user-defined weight W_i in the fitness function can correctly represent the emphasis from a user and be tuned for different applications.

The value of importance coefficients were determined based on experience to balance the unnormalized value of each performance metric. They were identified by carrying out several runs of simulations for different scenarios. Table 5.3 shows the empirical values used in this work.

Table 5.3: Importance coefficients of GA fitness function

Coefficients	The measures of performance under configuration X	Empirical values
α_1	$OWD(X)$ One-way delay performance	10^{-2}
α_2	$LSS(X)$ Loss performance	1
α_3	$TFD(X)$ network lifetime performance	1
α_4	$HFD(X)$ network lifetime performance	1
α_5	$EPUB(X)$ Energy consumption performance	10^{-1}

5.3.3 Genetic operators and selection mechanism

The types of crossover and mutation are of major importance to the performance of the GA optimization as discussed in Section 4.1. One-point crossover was used in this system while the mutation type selected was the classical one for binary representation, which swaps bits of each string (0 becomes 1 and vice versa) with a specific low probability p_m. Crossover is also applied with some specific probability p_c. Both these probability were tuned after proper experimentation and will be described in section 6.2.

The selection mechanism in the system is the routine roulette wheel selection scheme, also known as Stochastic Sampling. It was chosen based on the non-continuous nature of the fitness function. This mechanism selects parents in a similar way to the spin of a weighted roulette wheel. The roulette wheel is weighted according to the individual fitness values. A high-fit individual will have more area assigned to it on the wheel and hence, a higher probability of ending up as the choice when the biased roulette wheel is spun. Stochastic Sampling is a high-variance process with a fair amount of scatter between expected and actual number of copies.

5.3.4 The optimal sensor network design algorithm

Having completed the development of a representation scheme and forming the fitness function, the optimal sensor network design algorithm can be developed. In this work, the algorithm was implemented in C, the pseudo code is shown below.

Algorithm 6 Pseudo algorithms of the optimial sensor network design algorithm

```
SET population size M;
SET max number of generation G;
Generate random initial population of M WSN design solutions;
for t = 1 to G do
  Evaluate performance metrics in Table 5.1 for each individual in current pop-
  ulation through simulations;
  Calculate fitness value for each individual.
  for i = 1 to M/2 do
    Select two parents;
    Crossover with probability p_c and mutation with probability p_m;
    Decode string, evaluate fitness and record offspring;
  end for
  Replace old population with new offspring to form current population
end for
return best individual in current population
```

The number of iterations performed by the optimal WSN design algorithm in a run is of the order of $G * l * M^2$, where G is the number of generations of the GA, l is the big-string individual length and M is the population size. In other words, $G * l * M^2$ individual simulations are carried out in an optimization operation .

From the individual representation illustrated in Figure 5.2, if n is the number of fields in the individual structure and t_i is the size of each field where $i = 1, 2, ..., n$, then $l = \sum_i^n t_i$.

Before implement this algorithm, some of the issues have to be clarified. These include:

1. The size of the population which is a parameter of exploration that needs further discussion.

2. The genetic operators of crossover and mutation are applied with specific probabilities, this needs further discussion.

These will be discussed in the next chapter.

5.4 Summary

The main aim of this work is to develop a system which could select the proper set of protocols out of hundreds of solutions and optimize the number of nodes in a fixed area given a certain application scenario. A series of performance metrics for WSN assessment is identified in this chapter . Also, a well-informed performance function considering network connectivity, application-specific requirements and energy conservation, is derived to measure a WSN operation.

Several simulation tools are investigated in this chapter as well. The selected WSN simulator has to be able to simulate a WSN at the system level. Configuration parameters, such as the different protocols in each layer or node density must be programmable by a control element.

Intelligent optimization techniques are an efficient way of solving the searching part of our system. A genetic algorithm is used as the optimization tool of the designed system and an appropriate fitness function is derived to incorporate many aspects of network performance. The configuration attributes optimized by the genetic algorithm system comprise the proper communication protocol selection and the number of nodes deployed in a fixed area. An individual structure for WSN representation into GA was introduced. Any other configuration parameters could be optimized by our system by extending our individual structure. In addition to the representation scheme, the crossover operator and mutation operator of this GA system are discussed as well.

Having completed the development of a representation scheme and forming the fitness function, an optimal sensor network design algorithm has been proposed. Several implementation issues of the optimal sensor network design algorithm remain to be clarified in next chapter.

In the next chapter, the system framework and several issues in implementation are introduced and discussed in detail.

Chapter 6

SNDT System Framework and Implementation

The GA-based Sensor Network Design Tool(SNDT) proposed in this work relies on offline simulation analysis to help resolve design decisions. It takes into consideration the application-specific requirements, scenario information and different GA configuration. To design and build the best Sensor Network Design Tool, the foundation provided by the simulation engine must be as flexible as possible. Considering the simulation constraints discussed in section 5.2.4, the NS2 simulator was selected because it is not only easily extended but capable of analyzing network performance.

6.1 System Framework

An overview of the major components of this framework is presented in Figure 6.1. It takes into consideration the application-specific requirements, scenario information and different GA configurations. The user interface consists of query-based input/output and, potentially, visual display as well. The aim of the presentational element of SNDT is to keep the amount of displayed information to the minimum that is both instantly comprehensible and usable by a user.

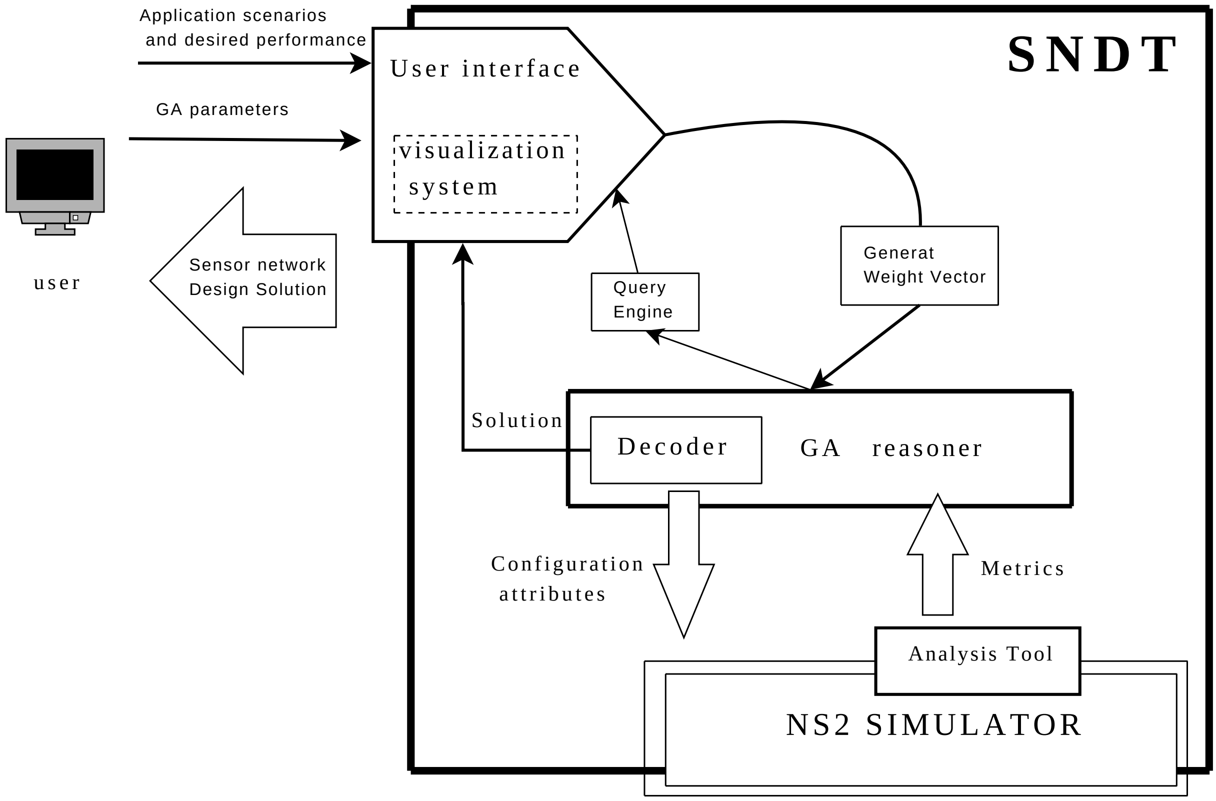


Figure 6.1: Conceptual System Design

The desired performance inputs are used by the GA-based reasoner to generate the weight vector for a specific application scenario. Application scenario parameters help the GA reasoner to setup the simulation environment. If more information is needed from the user, the GA reasoner could formulate more questions via the query engine. With all this information offered, the GA reasoner could start from a hypothesis/random network configuration. After decoding an individual structure into a NS2 readable format, the system calls the NS2 simulator in order to evaluate the current network configuration. Results from NS2 are passed through an analysis tool (to generate relative performance metrics) to the GA reasoner. The GA reasoner will then repeat the procedure until the desired result is reached.

The variables that SNDT takes as inputs and the outputs generated are summarized in Figure 6.2. This is a macro view of the input and output, which shows a higher level abstraction including all the initial inputs and then the resulting output when SNDT reaches its optimal solution. The feedback shown as the user modification represents the situation where the user decides to tune the GA parameters to modify an input.

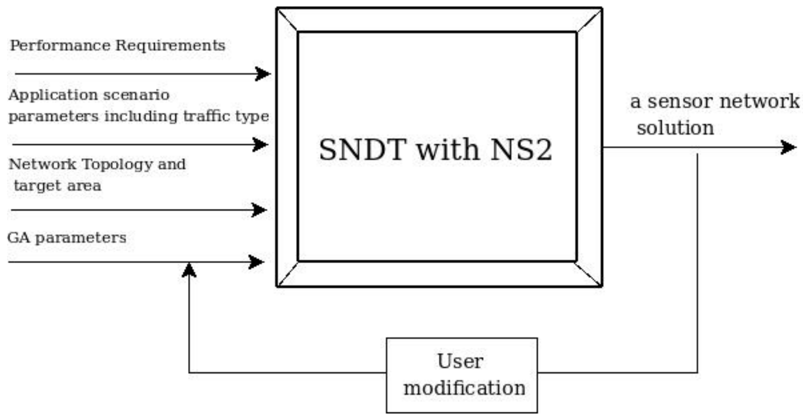


Figure 6.2: Global inputs and outputs

The whole system and the GA optimisation element are implemented in C, Figure 6.3 shows the flow chart of the framework.

The proposed framework works as follows, firstly, the end user will offer the system information for a specific application and relative performance requirements. This information should include the network topology, application traffic characteristics and the importance of each performance measure. The importance weight vector for the different metrics is essential in the performance function aforementioned. Then the optimization is carried out by identifying the fitness

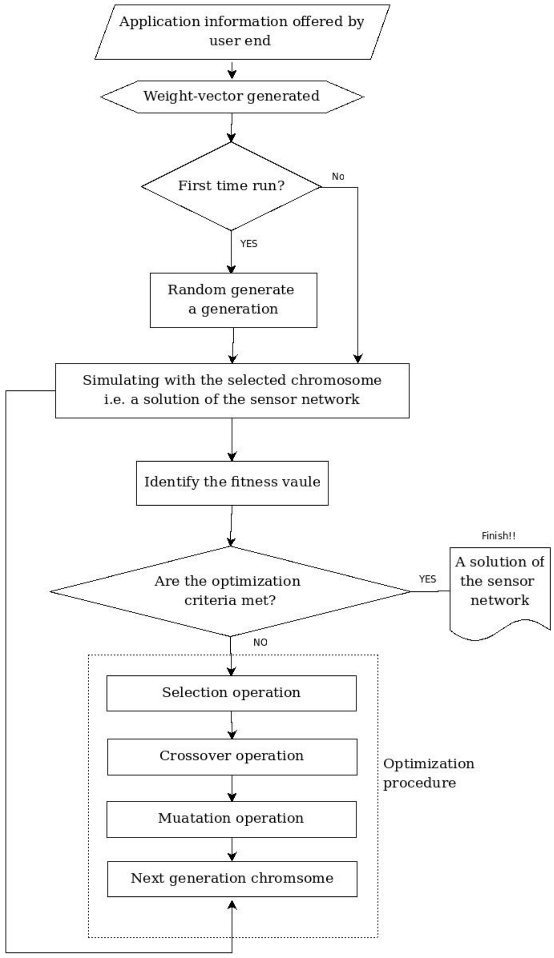


Figure 6.3: Flow chart of the framework

function of each individual candidate through simulations. Finally, after several runs of the GA, the system will offer the user the set of protocols and number of nodes which it deems to be optimal.

6.2 Implementation

6.2.1 Simulation challenges

Simulations with different protocol configurations NS-2 is a well known complex simulation tool for wireless sensor networks. It is based on two languages: an object oriented simulator, written in C++, and a OTcl(an object oriented extension of Tcl) interpreter, used to execute users command scripts. In this way, NS has two class hierarchies: the complied C++ hierarchy and the interpreted OTcl one, with one to one correspondence between them [94][95]. The complied C++ hierarchy allows NS users to achieve efficiency in the simulation and faster execution times. This is in particular useful for the detailed definition and operation of protocols. The NS users provide the OTcl scripts to define a particular network topology, the specific protocols and applications that they wish to simulate (which behaviour is already defined in the complied hierarchy) and the form of the output obtained from the simulator, i.e. the trace file.

This work was carried on under an NS-2.31+Ubuntu7.10 environment. The available protocols include DSR, AODV, SMAC, IEEE/802.11,IEEE/802.15.4 etc.. Other protocols could be obtained from the relative developers' sites or added for new protocols [96]. The available protocols are designed into two data structures shown in Figure 6.4.

Dealing with trace files Trace files record traffic and node movements generated from a NS2 simulation. Those files have to be parsed into analysis files in order to extract the performance information. The format of NS2 trace file is quite complex. In the work, all the trace files follow the new wireless trace file format [97] which can be found from the "trace/cmu-trace.cc" file of the NS

```
enum rtprotocol{

AODV=1
DSDV
DSR
MFlood                    /*module added by myself*/
Direct_Diffusion_TPP      /*Directed diffusion,two phase pull. */
Direct_Diffusion_OPP      /*Directed diffusion, one phse pull*/
Direct_Diffusion_PUSH
GEAR
LEACH
LEACH-C
LEACH-F
LEACH-MTE

}
```

```
enum macprotocol{

mac80211=1
mac802154
macsimple
mactdma
macsmac

}
```

Figure 6.4: Data structrue of available protocols for a WSN in NS2.31

directory . Here is an instance of the new trace format:

$$r - t10.207953583 - Hs3 - Hd - 1 - Ni3 - Nx5.00 - Ny95.00 - Nz0.00 - Ne9999.642613 - NlRTR - Nw - - - - Ma0 - Mdffffffff - Ms0 - Mt800 - Is0.255 - Id8.255 - ItDSR - Il32 - If0 - Ii1 - Iv32 - Pdsr - Ph1 - Pq1 - Ps1 - Pp0 - Pn1 - Pl0 - Pe0 - > 0 - Pw0 - Pm0 - Pc0 - Pb0 - > 0$$

This entry represents a packet which is received (r) at time (t) 10.207953583 sec, from source node (Hs) 3 broadcasting to the whole network (-1). The source node id (Ni) is 3, x-co-ordinate (Nx), y-co-ordinate (Ny),z-co-ordinate (Nz) are $5.00, 95.00, 0.00$ respectively. The energy level of this node(Ne) is 9999.642613. NL stands for the trace level, RTR represents the routing layer. The node event (Nw) is blank. The IP packet level information such as packet id , source address, source port number etc. can be found from fields Ii, Is, Id.

The Ns-2 simulator is called every time when the system calculates the fitness of the current population. Several AWK files were written to calculate metric values from an output of the ns-2 simulation.

6.2.2 GA related challenges

GAs have a number of parameters which are problem-oriented and have to be tuned in order to achieve the best algorithm performance. These parmameters are the population size, the probabilities of crossover and mutation and the type of crossover. Several GA-related issues for the optimal sensor network design algorithm were listed in section 5.3.4 and are further discussed in this section.

In this section, we use a simple periodic listening scenario to identify the best GA performance by tuning these GA-related parameters. The application scenario of these groups of experiments is in an $100m * 100m$ area. The sensor nodes are deployed in a grid fashion. The transmission range of each node can be determined by the number of nodes selected by the SNDT system. It is regulated by the condition where each node can just communicate with 4 neighbours. All the sensor nodes send data from their sensors(of message length 512 bytes) periodically every 20 secs to the base station. The main assumption here is that data are not aggregated at any point. Each node is a source of stream data to the base station. Table 6.1 lists a set of parameters for this group of experiments.

The requirements for the system performance are as follows. The network must be able to stay active as long as the simulation time. Message delay can be slightly tolerated, while a little message loss from each node could be accepted. Moreover, an efficient energy consumption is required for each node. For this specific periodic listening application, a list of weight values for performance metrics is required in the SNDT system (discussed in Section 6.1). Considering all the

Table 6.1: Simulation assumptions for identification of GA-related parameters

WSN charateristics	Assumptions
Application Scenario	Periodic Listening
Topology	grid
Applied area	$100m * 100m$
Communication range	each node can only communicate with 4 neighbours
Size of data message	512 BYTES
The interval between each report	20 secs
Simulation time	10000secs

metrics discussed in section 5.2.2, those requirements can be interpreted into a list of weight values shown in Table 6.2.

Table 6.2: Application requirements for identification of GA-related parameters

Performance requirements	Weight vector Value
Performance requirement for One-way delay w_{OWD}	0.05
Performance requirement for Loss w_{LSS}	0.05
Performance requirement for Time of first node died w_{TFD}	0.2
Performance requirement for Time of half nodes died w_{HFD}	0.35
Performance requirement for Energy consumption per useful bit w_{EPUB}	0.35

Two groups of experiments were carried out. The first group of experiments were used to study the GA performance for different population sizes with fixed crossover and mutation rate. The crossover and mutation rate were random chosen from the empirical ranges. (Note the crossover rate of most of the GA applications falls into the range of $0.3 - 0.9$, the mutation rate is in the range of $0.0001 - 0.01$ [69][78]). The second group of experiments were designed to identify the appropriate probabilities of genetic operators, the population size was set as the same one obtained from the first group of experiments.

I. Population size

At the beginning, a number of experiments were carried out to determine the most appropriate population size. A measure cycle of 20 generations with population size 4 for a periodic listening scenario took about 6 hours running on a Pentium III machine. Population sizes from 2 to 20 individuals were tested.

GA-related parameters such as crossover probability p_c and mutation probability p_m are set as shown in Table 6.3 for this group of experiments. They are chosen based on empirical experience.

Table 6.3: GA-Related parameters for identification of population size

GA-related parameters	Value
Crossover probability p_c	0.5
Mutation probability p_m	0.001
Selection method	roulette wheel selection

Figure 6.5 shows the average fitness of each measuring cycle with different population sizes, ranging from 2 to 20. The best performance in terms of the maximum average fitness function was achieved with a population size of 10 for this specific scenario. As the population size grows, the average fitness of each run increases dramatically at the first stage. This is because a small population results in an insufficient representation of good potential candidates. The average fitness value of each run is fairly low correspondingly. As population size accrues, more potential candidates with good genes are included in the GA selection process, the average value of fitness of each run rises until it gains its local optimum. A big population size could lead to a

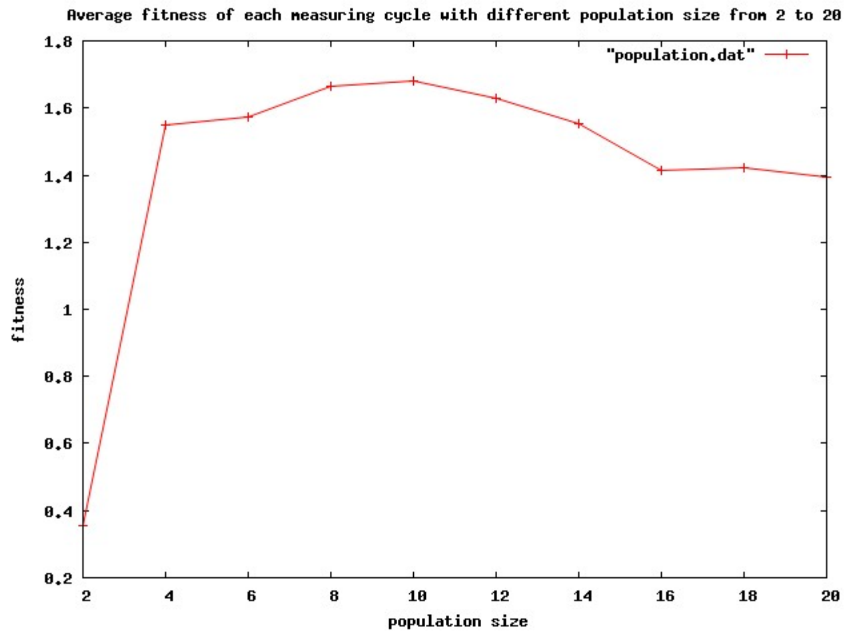


Figure 6.5: Population size VS. average fitness in 10 generations with population size 10

long runtime and the existence of more low fitness chromosomes. For this reason, the average fitness value of each run drops off from its optimal when the population size increases beyond 10.

The evolution progress of average fitness is shown in Figure 6.6, where the average fitness of the entire population of 10 at each generation is drawn. The optimization in the entire GA population can be seen from the general increase of the average fitness, although several fluctuations seen are caused by the search process through the crossover and mutation.

II. The probabilities of Genetic Operators

The one-point crossover was planned for use in the SNDT system with probability p_c while the mutation type selected was the classical one for binary representation, which swaps bits of each string (0 becomes 1 and vice versa)

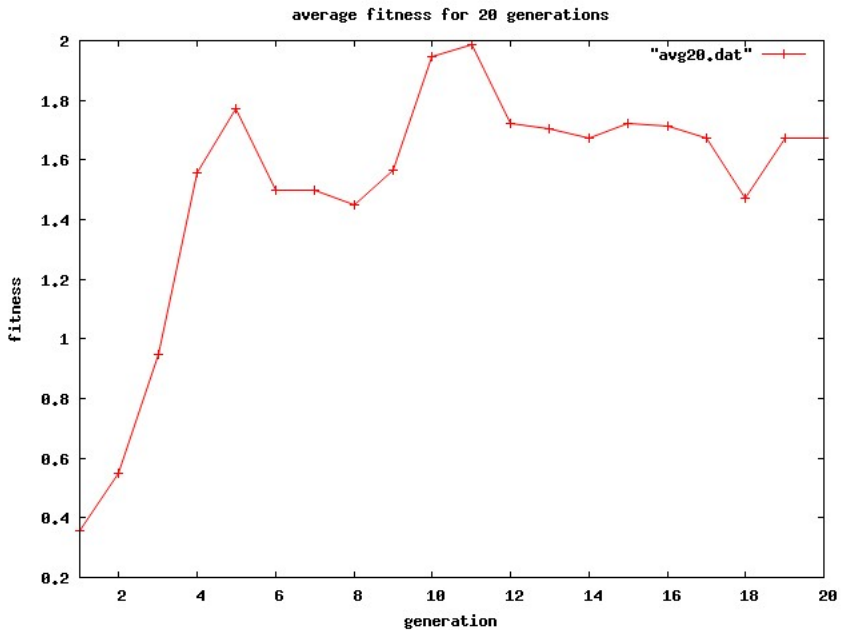


Figure 6.6: Evolution progress of the entire population(average fitness value) with a population of 10

with a specific low probability p_m.

Crossover probability p_c, sometimes known as "crossover rate", is a probability with which the GA will exchange some portion of bits from the two parent chromosomes . The crossover position of the chromosomes is chosen at random in our work. For example, assume crossover rate is 0.9, if a randomly generated number is less than the crossover rate p_c,the GA will crossover the parent chromosomes; otherwise no crossover will take place. The bigger the crossover rate is, the more total crossover operations will occur in a generation.

Similarly, the probability for an alteration or mutation to take place is called the mutation rate. New child chromosomes are created from the old parent chromosomes, and the new chromosomes form parts of a new population. In this work, exploration experiments were conducted to determine the value of p_c and p_m.

An other group of experiments of this periodic listening scenario with probabilities of crossover ranging from 0.3 to 0.9 for one-point crossover were performed to identify the the value of p_c. The simulation details were same as the one for identification of population size. The GA-related parameters are listed in Table 6.4. The selection progress of appropriate crossover probability p_c is shown in Figure 6.7, where the best average fitness of each run at different crossover probabilities is plotted.

Table 6.4: GA-Related parameters for identification of crossover probabilities

GA-related parameters	Value
Population Size M	10
Mutation probability p_m	0.001
Selection method	roulette wheel selection

It is clear that the average fitness value of each run increases dramatically as the crossover rate raises from 0.3 to 0.6. It is explained by the crossover operations could preserve the better genes from parents so that the new generations performance could be improved. However, Figure 6.7 shows explicitly that average fitness value of each run doesn't increase as the crossover probability grows, although crossover guarantees preservation of good genes from parents. This is because excess crossover operations in a generation could lead to low fitness due to the randomness of the crossover point. The average fitness peaked at the position in which crossover probability equals to 0.8. The results led to the use of one-point crossover with probability $p_c = 0.8$ for the following tests of the system.

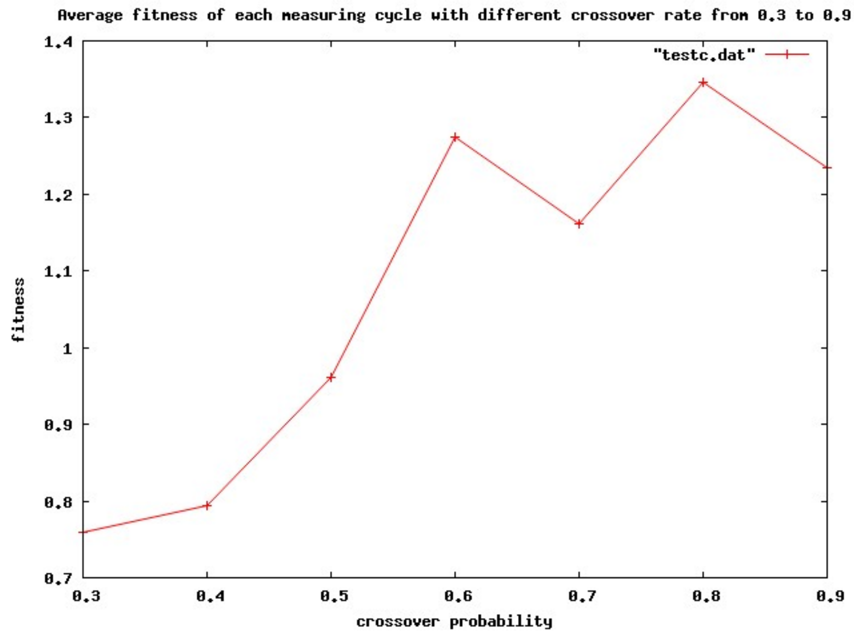


Figure 6.7: Crossover probability Vs Average fitness value in 20 generations with population size 10

Similar simulations were performed with probabilities of mutation ranging from 0.0001 to 0.01. The application scenario was the same as the one for identification of population size. The GA-related parameters has changed. The population size selected was 10 and crossover probability p_c was set to 0.8, which results from Figure 6.7. These are shown in Table 6.5. The selection progress of appropriate mutation probability p_m is shown in Figure 6.8, where the best average fitness of each run at different mutation probabilities is plotted.

Figure 6.8 shows the average fitness of each measuring cycle with different mutation rate, ranging from 0.0001 to 0.01. The mutation rate increases at the order of 0.0005. The best performance in terms of the maximum average fitness function was achieved with a mutation rate of 0.008 for this scenario. The total number of mutations of a generation increases as the

Table 6.5: GA-Related parameters for identification of mutation probabilities

GA-related parameters	Value
Population Size M	10
Crossover probability p_c	0.8
Selection method	roulette wheel selection

mutation rate rises. Many mutation operations in a generation could bring unfavourable modifications of current chromosomes in terms of fitness value. It may reduce the overall fitness performance of the whole generation.

III. Selection of initial population

The GA incorporates stochastic operation during the selection process while the quality of the randomly generated initial population dramatically affects the system performance. Currently, we use an random initial population to run the SNDT system. In the future, different initial populations could be investigated.

The individual length in our current design is 16, which represents two sets of 4-digit protocol parameters and one 8-digit representing number of nodes. Each set of protocol parameters has 16 iterations which can be coded for different protocols. Due to the limited WSN protocol constraints of NS2, some combinations of digits may be meaningless. This could present a serious waste for system runtime. To avoid this situation, a function was developed to check the availability of each new individual before it is sent to the NS2 simulator. If the decoded genes represent available protocols, the new individual is passed to the simulator for fitness calculation. Otherwise, a default protocol for each layer is substituted as a part of the new chromosome.

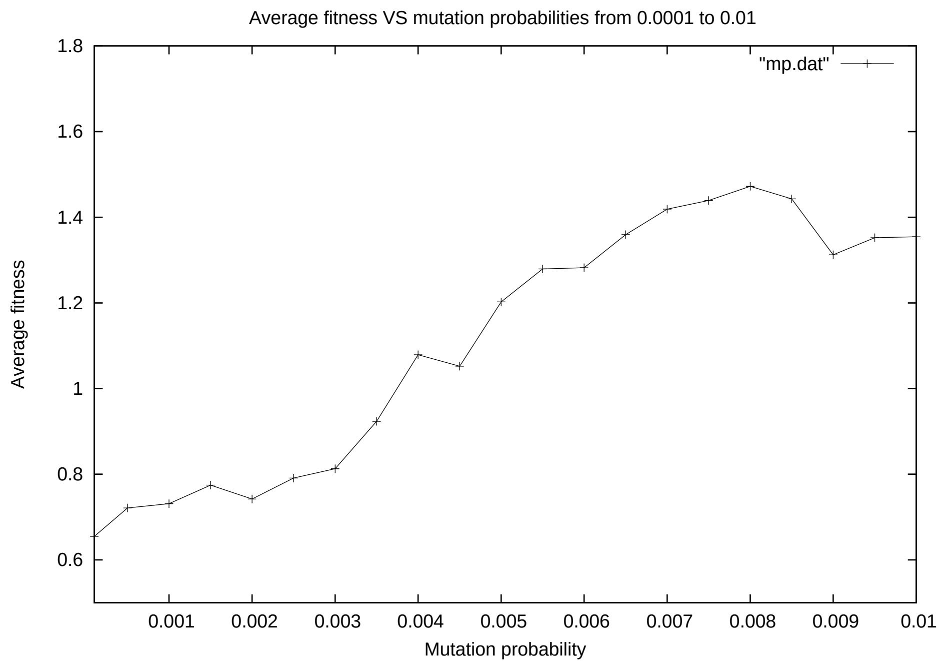


Figure 6.8: Muatation probability Vs Average fitness value in 20 generations at with population size 10

A similar function was designed for node density parameters. For a specific application scenario, the simulation time may change based on the scale of the number of nodes. The node number plays a vital role for the whole system computation time. 8-digits represents node numbers ranging from $0 - 255$. Any number of nodes less than 9 is not appropriate for a valid network solution (simulation constraints).

6.2.3 System Integration

The proposed SNDT framework automates the design procedure of a WSN to find a good set of the protocols out of hundreds of configuration possibilities as well as the number of nodes in a fixed area for a certain application scenario before physical deployment. In the following paragraphs, the approach of setting up the weight vector and the files distributed within the SNDT framework will be described.

Weight vector for performance metrics A list of weight values for performance metrics is required in the SNDT system as discussed in Section 6.1. They are defined by the user and follow the rule that the sum of these weights equals 1(as shown in Function 6.1).

$$w_{OWD} + w_{LSS} + w_{EPUB} + w_{THD} + w_{TFD} = 1 \tag{6.1}$$

A user has to define the value of each weight for different performance metrics which depends on the requirements of a certain application. A respectively greater value (in the range of 0-1) will be given to the performance metrics which are crucial for this application.

Files distributed with SNDT SNDT is mainly implemented in C language. The GA-reasoner is a the C-language translation and extension of the original Pascal SGA code presented by Goldberg [69]. It has some addition features, but its operation is essentially same as that of the original, Pascal version. The simple example problem included with the Pascal version (finding the maximum of x^{10},where x is an integer interpretation of a chromosome) was used to verify the correctness of the operation in the GA-reasoner. The result shows that the Pascal version was successfully translated.

The NS simulator is called when a fitness value of a chromosome is calculated. Several AWK files are written to calculate metric values from an output of the ns-2 simulation. In the following paragraphs, an outline of the file distributed and important routines contained in those files are presented.

sga.h contains declarations of global variables and structures for SNDT. This file is included by *main()*. Both sga.h and external.h have two defines set at the top of the files. The available protocols in each layer are defined in structures MACPROTOCOL and RTPROTOCOL seperately.

external.h contains external declarations for inclusion in all source code files except *main()*. The external declarations in should match the declarations in sga.h.

main.c contains the main genetic algorithm loop, *main()*.

generate.c contains generation(), a routine which generates and evaluates a new GA population.

initial.c contains routines that are called at the beginning of a GA runs and presents an interface for system users.

initialize() is the central initialization routine called by *main()*.

initdata() is a routine to prompt the user for GA-related parameters.

initpop() is a routine that generates a random population. This is designed to trigger the system to run the GA-reasoner. Currently, SNDT includes no facility for using seeded population.

memory.c contains routines for dynamic memory management.

initmalloc() is a routine that dynamically allocates space for the GA population and other necessary data structures.

operator.c contains the routines for genetic operators.

crossover() performs single-point crossover on two parents , producing two children chromosomes.

mutation() performs a point mutation.

random.c contains random number utility programs, including:

randomperc() returns a single, uniformly-distributed, real,pseudo-random number between 0 and 1. The method can be referred in [98].

flip(p) flips a biased coin ,returning 1 with probability p, and 0 with probability $1 - p$.

report.c contains routines used to print a report from each cycle of SNDT's operation. The output of SNDT is based on analysis of the report generated from this file.

report() controls overall reporting.

writepop() writes out the population ar every generation.

writechrom() writes out the chromosomes as a string of ones and zeroes. In the current implementation, the most significant bit is the rightmost bit.

stats.c contains the routine *statstics()*, which calculates populations statistics for each generation.

utility.c contains various utility routines.

ithruj2int() is a most important utility routine which matchs a series of ones and zeroes into different network design parameters. The return value of this routine is an *int* value interpreted from a string of ones and zeroes from bits i through j of a chromosome.

app.c contains application dependent routines. The fitness calculation and chromosome interpretation are carried out in this files.

app-data() is designed as an interface to acquire the weight value of a list metrics for a specific application scenario. This routine is called by *init-data()* in initial.c.

app-report() prints out the candidate network solution interpreted from a chromosome (including routing layer protocol, mac layer protocol and number of nodes in a fixed area). Due to the limited number of protocol modules available in NS2 simulator, combination digits can be examined here to check the availability of protocol decoded before it is sent to the NS2 simulator.

objfunction(string) The objective function of the SNDT system. The variable string is a pointer to an individual(a GA population chromosome), to which this routine must assign a fitness. NS simulator is called in this routine. A bash script *ns-analysis.sh* is designed for carrying out NS simulations and following analysis for desired metrics.

6.3 Summary

This chapter describes the major components of the SNDT framework and summaries the variables that SNDT takes as inputs and outputs. A detailed flowchart of this framework is shown as well. The SNDT takes into consideration the application-specific requirements, scenario informations and GA configurations to automate the design procedure for a good network solution.

Considering the simulation constrains in WSNs, the NS2 simulator was selected as the underlying simulator of the SNDT framework. The way of simulating different protocols and dealing with the tracefile are discussed in section 6.2.1.

GAs have a number of parameters which are problem-oriented, these are population size, the probabilities of crossover and mutation and the type of crossover and mutation. They are identified in section 6.2.2 and analyzed to achieve a good algorithms performance. The population size is determined to be 10 and the crossover rate and mutation rate are identified as 0.8 and 0.008 respectively. Also in this chapter, the approach of setting up the weight vector and the files distributed within the SNDT are described.

In the next chapter, three generic case studies will be performed to demonstrate how the SNDT system can be useful for WSN designers. Specifically, a periodic-measuring application, the notification stage of an event detection and a tracking-based application are designed and investigated. The results from the SNDT are validated through simulations.

Chapter 7

Case Studies and Evaluation

In the previous chapters, SNDT was introduced and the framework of this design tool was implemented. In this chapter, we show how SNDT will be used to provide appropriate solutions for some generic applications and we validate the design decisions by analysis. The case studies are chosen to represent unique scenarios in WSNs and we try to show the applicability of the proposed design framework for development of a complete WSN system.

7.1 Case study 1 : Periodic-measuring Application

Periodic-measuring application is one type of WSN application where a user wants to collect several sensor readings from a set of points in an environment over a period of time in order to detect trends and interdependencies. Normally, users want to collect data from hundreds of points spread throughout a given area and then analyze the data offline. In these type of applications, the topology is required to be stable and it collects data over several months or years. Hence, periodic measurement applications do not need to have strict latency requirements but lifetime needs to be maximized. Data samples can be delayed inside the network for moderate periods of time without significantly affecting application performance.

A periodic-measuring scenario is designed in this section. The physical architecture of this scenario is shown in Fig 7.1. The sensor nodes are deployed in a grid fashion. Each sensor node sends data from its sensor(of message length 512 bytes) periodically every 100 secs to the base station. It is constrained with the condition that each node just can communicate with 4 neighbours.

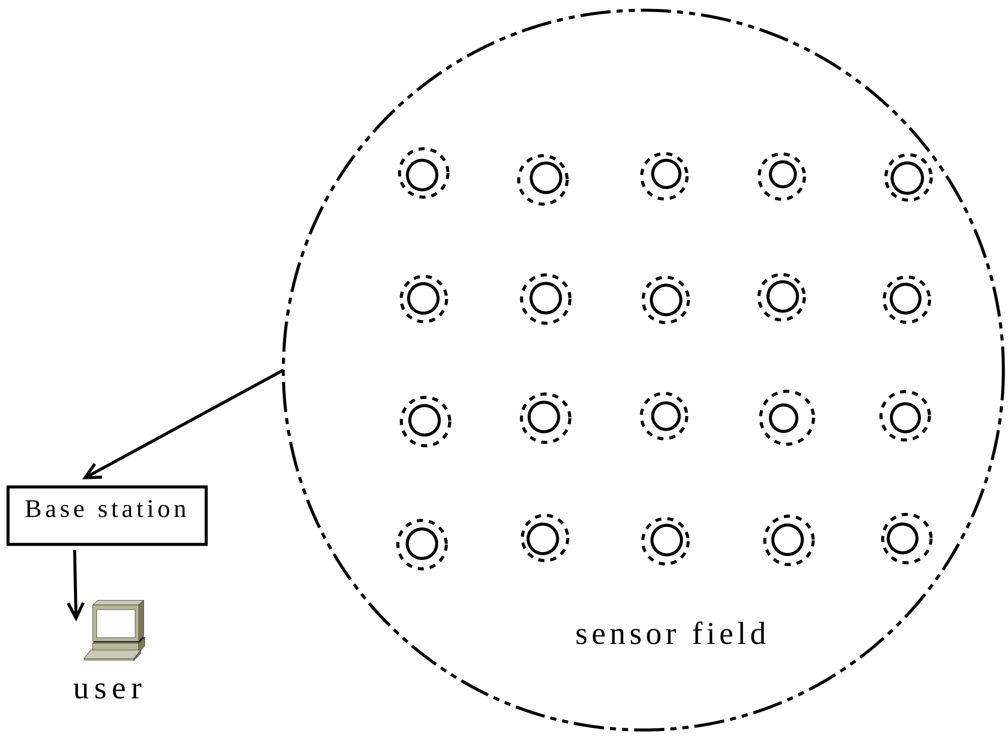


Figure 7.1: Physical architecture of a WSN for periodic-measuring application

Several other assumptions are taken into account for completeness.

- All the nodes are homogeneous in terms of battery and transmission range.

- Data are not aggregated at any point.

- Each node is a source of data to the base station.
- The network consists of up to 255 uniformly deployed nodes.
- An omnidirectional antenna is installed in each sensor node.
- Each node has a unique ID in the network.
- The data rate is sufficiently low that there will not be queuing delay in sensor nodes.
- Each node has a given uniform error model over incoming wireless channels with error rate equals to 0.1.
- The initial energy of each sensor device is configured to be $1000 Joule$. This is expected to be consumed it during the experiment. The power consumption model is installed in each sensor device, as shown in Table 7.1:

Table 7.1: Energy model of wireless sensor nodes in a periodic-measuring case

Mode of Sensor	Value ($WATT$)
Sleep Power	0.001
Idle Power	0.8
Receive Power	0.8
transmit Power	1.0
transition Power	0.2

7.1.1 Inputs of SNDT

Having described the inputs of SNDT in section 6.1, the application-related information and network topology have to be offered to the SNDT system. The maximum number of sensor nodes which can be used in this area is set to 255 . The simulation time is set at 10000sec.

We evaluate this case under Constant Bit Rate traffic generation as the sensor nodes send information regularly to the base station. The packet size is set at 512 bytes. Normally, this type of scenario expects typical reporting periods to be between 1 and 15 minutes [80]. Here, the interval between two packets is set at $100s$.

The requirements for the system performance are as follows. The network lifetime must be maximized, which implies that energy efficiency is critical concerned under this scenario. We set the importance weight w_{THD} equal to 0.3 (Note that $\sum_i^5 w_i = 1$, the average value for each metrics equals to 0.2), w_{EPUB} equal to 0.3 and w_{TFD} equal to 0.2. The network delay can be tolerated which leads to a less value for the importance weighting coefficients w_{OWD}, 0.05 is given. In addition, w_{LSS} is set to 0.15.

However, the weight chioce for a certain application scenario with implicit requirements is non-dominated. Slightly changed weights (in a range of $0 - 0.05$) lead to the same networking solution according to our simulation results. Any changes beyond this range may result in different network solution. This is obvious as the weight represents the importance of each metric.

The GA-related parameters are set according to the analysis in section 6.2.2 as follows: the population size is selected to be 10 and the crossover rate and mutation rate are identified as 0.8 and 0.08.

An example of the SNDT framework interface is displayed in Figure 7.2. The inputs of SNDT system for this periodic-measuring scenario can be summarized as in Table 7.2. The performance requirements input part represents the user-defined requirements in the framework, the application related input represents the network characteristics, and target area information relates to the nework topology.

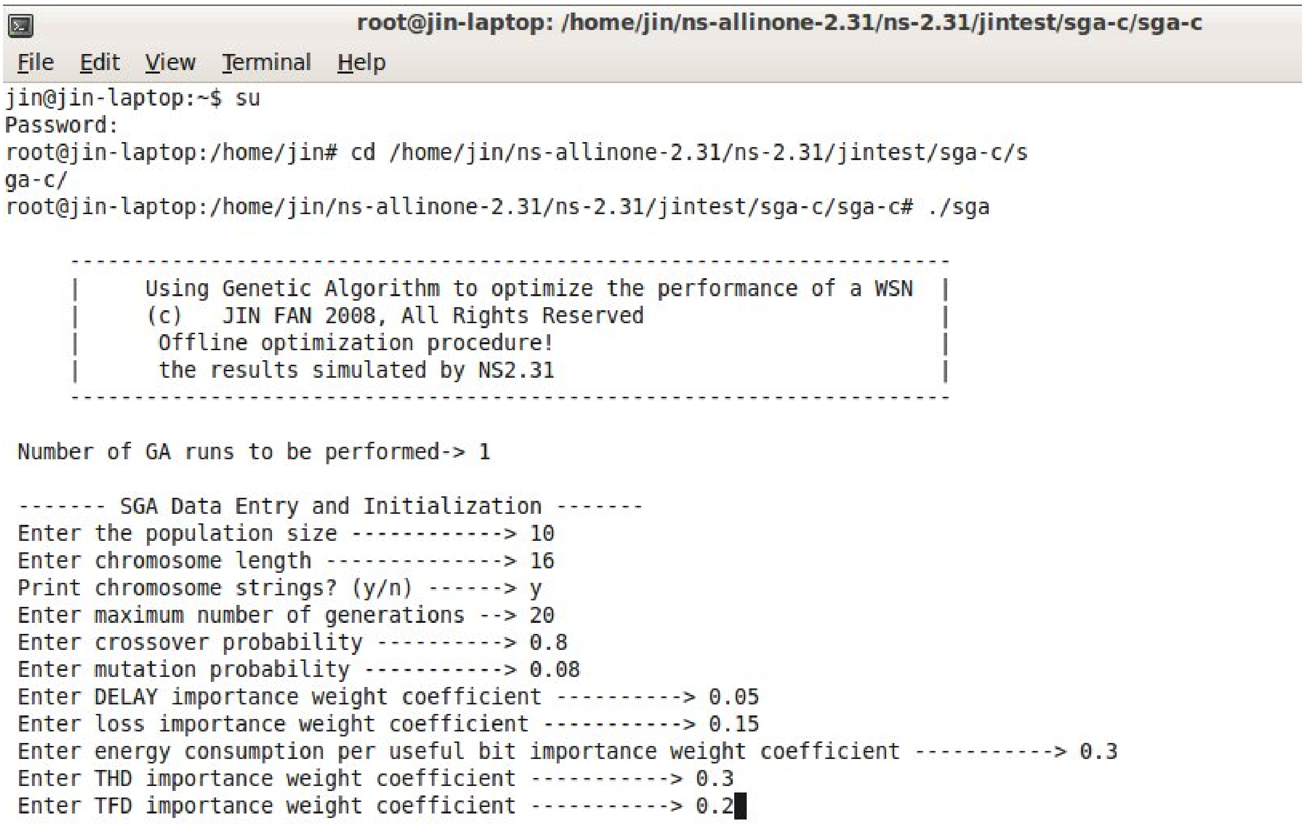


Figure 7.2: SDNT User interface

7.1.2 Outputs of SNDT

The resulting solution from SNDT system is always shown in the form of a report. This report displays both individual and statistical information of each cycle, including the fitness of each chromosome, crossover site, total crossovers and mutations occured in a generation. It also interprets the binary series into parameters for each networking solution. Due to limited space, only a part of this report is shown in Figure 7.3.

From the report, the result solution is the global best individual 1000110001100010. As mentioned in section 5.3.1, a chromosome structure consists of routing protocol, MAC layer protocol and number of nodes in the fixed area. The first four

Table 7.2: Inputs of SNDT for the Periodic-measuring scenario

Application related	Traffic type	Constant Bit Rate
	PacketSize	512bytes
	Packet Interval	$100s$
	Data Duration	$9000s$
Target area Information	Topology	Uniform distributed
	Area range	$100m * 100m$
Performance requirements	One-way Delay	w_{OWD}=0.05
	Loss	w_{LSS}=0.15
	Time of first node died	w_{TFD}= 0.2
	Time of half nodes died	w_{HFD}= 0.3
	Energy consumption per useful bit	w_{EPUB} =0.3
GA related parameters	Population size	10
	Selection Mechanism	Stochastic Sampling
	Crossover Method	One-point Crossover
	Crossover Rate	0.8
	Mutation Method	Swap bits of each string following the p_m
	Mutation Rate	0.008

digits represent the routing protocol, the following four digits represent the MAC layer protocol, last eight digits represent the number of nodes in the fixed area. The encoding mechanism of first two fields is shown in the Table 7.3 and Table 7.4.

Table 7.3: Routing protocols Binary representation in SNDT

Routing Layer Protocol	Binary series
AODV	0000
DSR	0001
DSDV	0010
TORA	0011
MFlood	0100

This shows that the best routing and MAC protocols for this scenario are DSR

```
---------------------------------------------------------------------------
                          Population Report
Generation   19                                          Generation   20
num   string            fitness    parents xsite  string           fitness
---------------------------------------------------------------------------
  1)  1111110111101000  30.701533 | ( 1, 8)   13   1001110101101001  31.442191
  2)  1001011101001001  25.796448 | ( 1, 1)   13   1110010101101000  5.710451
  3)  1001101111101000  10.701533 | ( 1, 1)    4   1110110111110000  10.598362
  4)  1001000110000111  32.177770 | ( 1, 1)    4   1111110111101001  12.330057
  5)  1000011100101001  27.186980 | ( 1, 1)   11   1111110111101001  12.330057
  6)  0111010001000110  27.186980 | ( 1, 1)   11   1111110111101000  10.701533
  7)  0100010110100101  13.394443 | ( 1, 1)    0   1111110001101000  0.669650
  8)  0100010110100100  38.203200 | ( 1, 1)    0   1011110111101000  10.701533
  9)  0110111101100000  11.691998 | ( 1, 1)    8   1111110111101000  10.701533
 10)  1111101010011101  4.054658 | ( 1, 1)    8   1111100111101000  10.701533
---------------------------------------------------------------------------
Generation 19 Accumulated Statistics:
Total Crossovers = 7, Total Mutations = 29
min = 0.669650   max = 31.442191    avg = 10.588690   sum = 115.886901
Global Best Individual so far, Generation 15:
Fitness = 53.203200: 1000110001100010
---------------------------------------------------------------------------
```

Figure 7.3: a part of a SDNT result report

Table 7.4: MAC Layer protocols Binary representation in SNDT

MAC Layer Protocol	Binary series
IEEE/802.11	0000
TDMA	0001
SMAC	0010
IEEE/802.15.4(non-beacon mode)	0011

and IEEE/802.14.5; the number of nodes in the area is 70 . This is found by interpreting the corresponding binary series from the rightmost bit to the left.

Using this case study, we have shown how designers can use SDNT to get network solution based on optimization. Although the actual performance of this framework depends on GA-related parameters, the accuracy of the assumptions made and the selection of underlying simulation tool, going through this process is extremely valuable for the designers in understanding the impact of the system parameters on the performance. The resulting solution is validated in the next sections.

7.1.3 Evaluation of Design decisions

In this section, we evaluate the design decision obtained for this periodic measurement case through simulations and compare it with all the other possible solutions in SNDT. The performance of each networking solution is assessed for energy efficiency, End-to-End Delay and loss.

The validation of the SNDT solution consists of two parts: first, the choice of node density in the area has to be examined. Then the selection of the set of communication protocols needs to be verified. Two groups of experiments were carried out. The first group of experiments were used to study the metrics for different numbers of sensor nodes during the monitoring period. The set of communication protocols was kept same as those in the SNDT solution, the number of nodes was set to the range of 10 to 255. The second group of experiments were designed to evaluate the goodness of the selected set of communication protocols. With a fixed number of nodes, we compared the protocol set(in the SNDT solution) with all the other possible protocol combinations represented in the SNDT system in terms of End-to-End delay, loss and energy efficiency. All the application-related parameters were the same as those in Table 7.2.

7.1.3.1 Verification of number of nodes

Figure 7.4 shows the energy efficiency variation with different number of nodes in this specific scenario under the same communication protocol configuration as in the SNDT solution. Energy efficiency is estimated by dividing the average energy remaining per node by the initial energy. Note that the curve changes slightly with increasing number of nodes (in the range of 0.1%). The energy efficiency starts to increase when the node number equals to 50. It reaches the local maximum

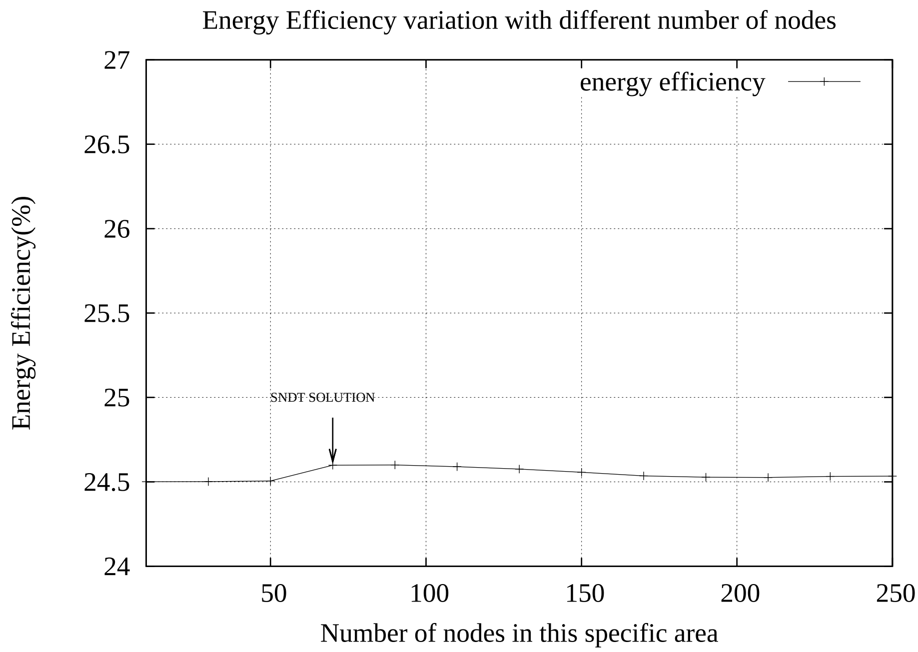


Figure 7.4: Energy Efficiency variation with different number of nodes under DSR/802.15.4 configuration

where the number of nodes equals to 70, which is also the number offered by the SNDT solution. The curve is then stable until the node number equals 90. After that, it descends gradually with increasing number of nodes. This is because only one PAN coordinate was used in this area. With many devices spread over, IEEE/802.15.4 cannot achieve the very best energy efficiency.

The network latency also depends on the routing layer protocol. We measure the average End-to -End delay for different sizes of this network. Figure 7.5 shows the End-to-End Delay variation with different numbers of nodes in this scenario under the same communication protocol configuration as the SNDT solution. It is clearly seen that the End-to-End Delay generally increases as the number of nodes rises. The increment of delay is expected, since more sensor nodes are involved when the number of nodes increase. The number of node selected by the SNDT has good performance in terms of End-to-End Delay.

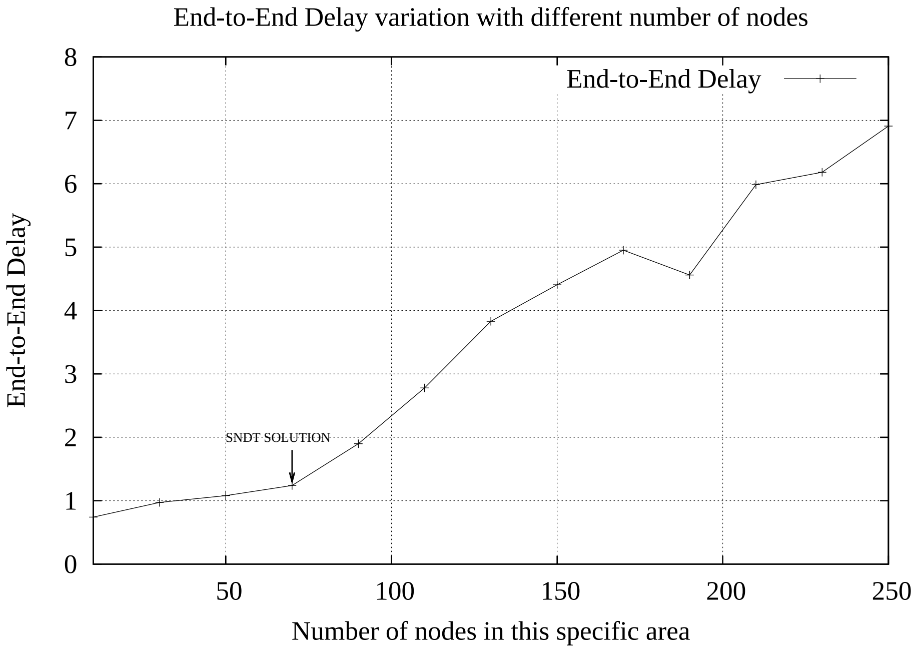


Figure 7.5: End-to-End Delay variation with different number of nodes under DSR/802.15.4 configuration

Figure 7.6 shows the Loss percentage variation with different number of nodes in this scenario under the same communication protocol configuration as the SNDT solution. It is calculated by the number of lost packets divided by the number of sent packets. In general, the loss performance degrades when the number of nodes increases. This is expected as the chosen MAC layer protocol suffers from the hidden terminal problem. The number of node selected by the SNDT does not have the best performance in terms of loss but is good enough for the requirements of this scenario.

The number of nodes selected by SNDT for this case compared favourably to the other possible node densities in terms of energy efficiency, End-to-end delay and loss. This number is used in the next group of experiments to identify the goodness of the set of communication protocols in SNDT solution.

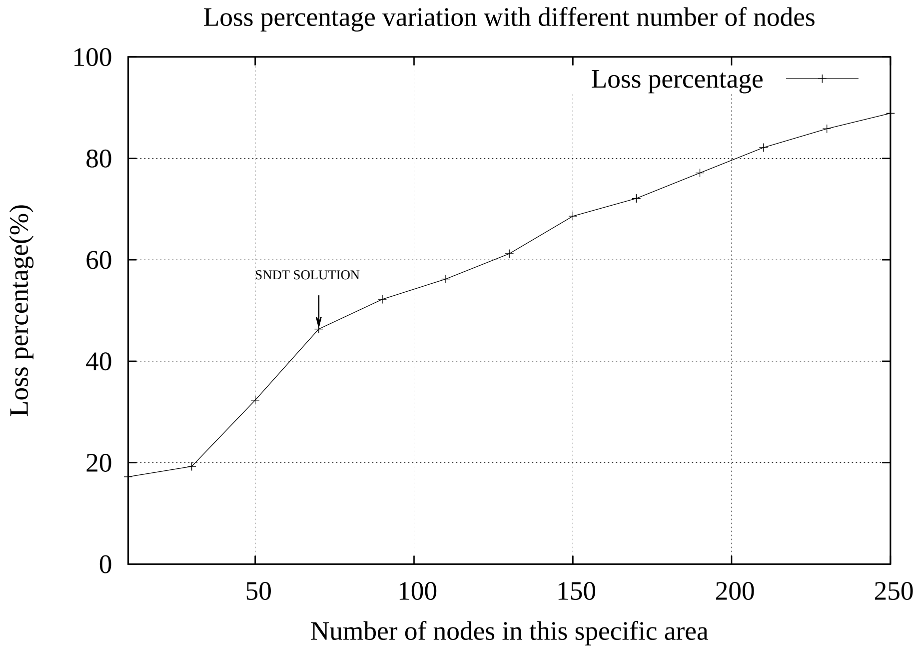


Figure 7.6: Loss variation with different number of nodes under DSR/802.15.4 configuration

7.1.3.2 Verification of the selected set of protocols

To distinguish experiment results from the SNDT solution, we use Solution 1, Solution 2,.....,Solution 19 to denote other possible protocol combinations in the SNDT system(available protocols in each layer have listed in Table 7.3 and 7.4). The detail of each solution is displayed in Table 7.5. The SNDT solution is then compared with these possible solutions in terms of energy efficiency, End-to-end delay and loss. We analyze these 20 solutions in 4 groups , every 5 solutions form a group. Solutions in each group have the same MAC layer protocol. We use Group 1 to 4 to represent them.

I. Energy efficiency

Figure 7.7 shows the energy efficiency of this scenario with 70 nodes, for different communication protocol configurations. The first result of interest

Table 7.5: Possible solutions in SNDT with a fixed number of nodes

SOLUTION	Routing Layer	MAC Layer
Solution 1	AODV	IEEE/802.11
Solution 2	DSR	IEEE/802.11
Solution 3	DSDV	IEEE/802.11
Solution 4	TORA	IEEE/802.11
Solution 5	MFlood[1]	IEEE/802.11
Solution 6	AODV	TDMA
Solution 7	DSR	TDMA
Solution 8	DSDV	TDMA
Solution 9	TORA	TDMA
Solution 10	MFlood	TDMA
Solution 11	AODV	SMAC
Solution 12	DSR	SMAC
Solution 13	DSDV	SMAC
Solution 14	TORA	SMAC
Solution 15	MFlood	SMAC
Solution 16	AODV	IEEE/802.15.4
Solution 17	DSDV	IEEE/802.15.4
Solution 18	TORA	IEEE/802.15.4
Solution 19	MFlood	IEEE/802.15.4

is that the SNDT solution outperforms all the other possible solutions. This is expected, as the MAC layer protocol IEEE/802.15.4 in the SNDT solution has low energy consumption by reducing the MAC layer control packets(no RTS/CTS mechanism). Also , the routing layer protocol DSR saves energy by limiting the routing layer control packets.

Group 1 performs the worst among others. This is because the MAC layer protocol in these solutions is IEEE/802.11. This protocol has a power management scheme that requires nodes to remain active even though the medium is idle. In this group, Solution 1, Solution 2 and solution 4 have better performance than Solution 3 and 5, since the routing protocols used in these two solutions (AODV and DSR)are reactive protocols which do not

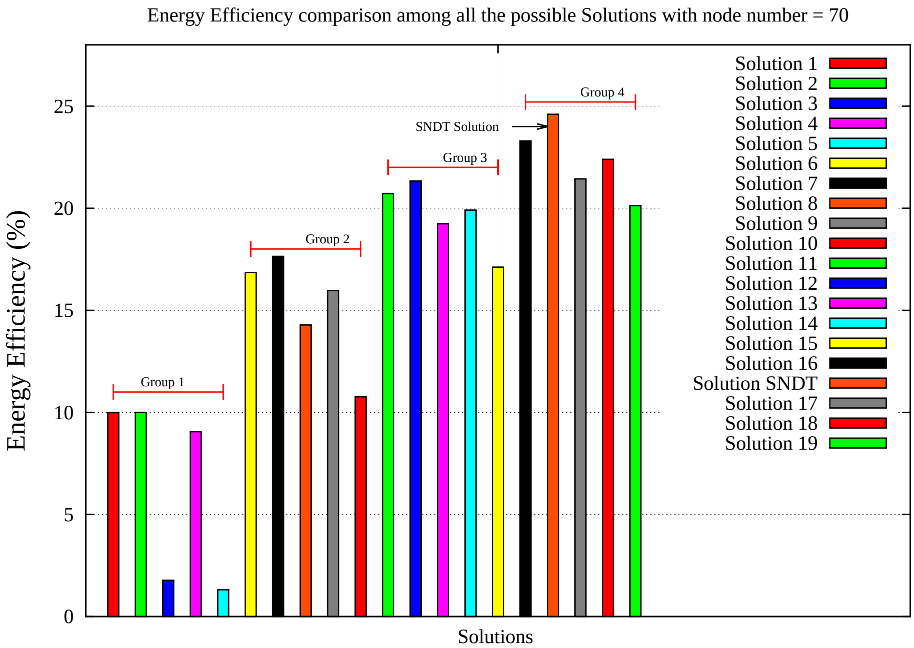


Figure 7.7: Energy Efficiency comparison among all the possible Solutions in SNDT with node number = 70

need to involve periodic exchange of information resulting in consumption of energy for each node. Solution 2 gains the best energy efficiency in Group 1. This can be explained by the small routing overhead of the DSR protocol. Solution 5 performs worst among this group. The flooding mechanism consumes large amounts of energy by packet implosion and overlapping. Solution 3, due to the time triggered routes updates , performs better than Solution 5.

Group 2 has better energy efficiency than Group 1. This is mainly because the TDMA protocol has a natural advantage of collision-free medium access which requires a relative small amount of energy consumption compared with the IEEE/802.11 protocol in Group 1. The Solution 6-10 follow the same trend in Group 1. Group 2, due to the strict synchronization requirements, performs worse than Group 3.

Group 3 enhances the energy efficiency by 100% compared to Group 1. This is expected, as the SMAC protocol employs a low duty cycle scheme to avoid overhearing(which consumes a lot of energy in the IEEE/802.11 protocol) by putting nodes into sleep mode. In this group, Solution 12 outperforms the other solutions.

Group 4 dominates over Group 3, as the MAC layer protocol IEEE/802.15.4 saves more energy than SMAC by dropping of the RTS/CTS mechanism completely. Limited control overhead in IEEE/802.15.4 leads to better energy conservation than the other MAC protocols. The SNDT solution is in this group, it has 24.5% energy efficiency, 1.53% more than Solution 16 which selects AODV as the routing protocol.

II. End-to-End Delay

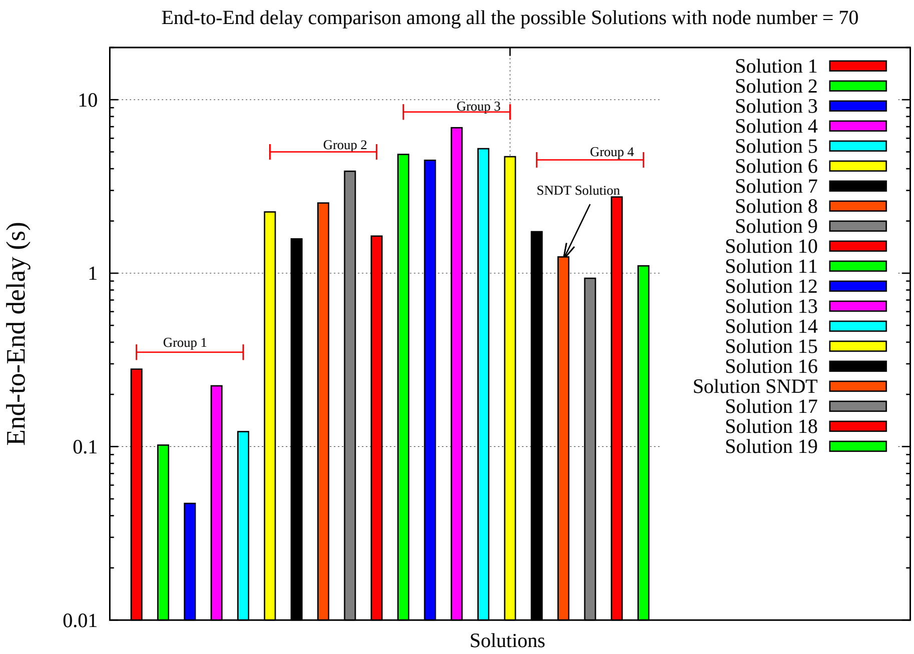


Figure 7.8: End-to-End delay comparison among all the possible Solutions in SNDT with node number = 70

Figure 7.8 displays the network latency comparison with 70 nodes in terms of End-to-End Delay, for different communication protocols configurations. It is clear that the SNDT solution does not perform the best among all the other solutions. We follow the analysis method used in the energy efficiency review by dividing the total 20 solutions into 4 groups according to different MAC layer protocols. Group 1 as shown in Figure 7.8 has the shortest End-to-End delay overall. This is expected, because IEEE/802.11 uses a scheme known as carrier-sense, multiple access, collision avoidance(CSMA/CA) which can minimize network latency by using four different types of packets: request-to-send(RTS), clear-to-send(CTS), data and acknowledge(ACK) to transmit frames, in a sequential way. In Group 1, Solution 3 outperforms the other solutions. This is because the routing protocol DSDV is a proactive routing protocol which ,in most of the cases , uses established routes and tries to get rid of packets immediately resulting in low average delay. Solution 2 has better network latency performance than Solution 1. This is can be explained because the routing protocol AODV in Solution 1 has much more routing overhead which may lead to longer delay. Solution 4 has the longest End-to-End delay in this group, this is because the TORA protocol is more suited for heavier traffic scenario.

Group 2 increases the end-to-end delay by roughly 10 times compared with Group 1. (Note that a log-axis is on the y-axis in Figure 7.8). The MAC layer protocol employed in this group is TDMA. It divides the use of the channel into fixed time slots and schedules the transmission, which could bring significant network delay. Solution 8 which shares the same routing protocol as Solution 3 does not perform best in this group. The routing protocol DSDV requires periodic advertisements, large amount of routing overhead has to be transmitted when the time slot is available. The combination of DSDV and TDMA leads to long delay. Solution 7 and 10 have the best network latency performance in Group 2.

Group 3 has the longest End-to-End delay among these four groups. This is expected, as the periodic sleeping mechanism in the SMAC protocol could cause long waiting times for the transmitted packets. When a node tries to send a packet, it must wait until the receiver node wakes up. In addition, the duty cycle in our simulation is 10%, this introduces severe delay in this scenario. In this group, Solution 13 performs worst due to the large routing overhead resulting from the DSDV protocol. Solution 12 has the best latency performance, as the DSR protocol has lower routing overhead compared with the others.

Group 4 has the second best performance in terms of End-to -End delay overall. The difference in network latency between Group 4 and the other groups comes from the fact that the MAC layer (IEEE/802.15.4) in Group 4 does not use either periodic listening or the RTS/CTS mechanism. A Small control overhead results in better network latency performance. The SNDT Solution performs second best in this group. Solution 17 has the shortest End-to-End delay among these five solutions, similarly to Solution 3 in Group 1. Solution 18 performs worst in this group.

To conclude, the SNDT solution has a good network latency performance in terms of average End-to-End delay. Solution 3 performs best among these 20 solutions in the latency aspect. Solutions with THE IEEE/802.11 protocol have smaller average End-to-End delay than those with other MAC layer protocols. At the same time, solutions with the SMAC protocol perform worst among other solutions.

III. Loss performance

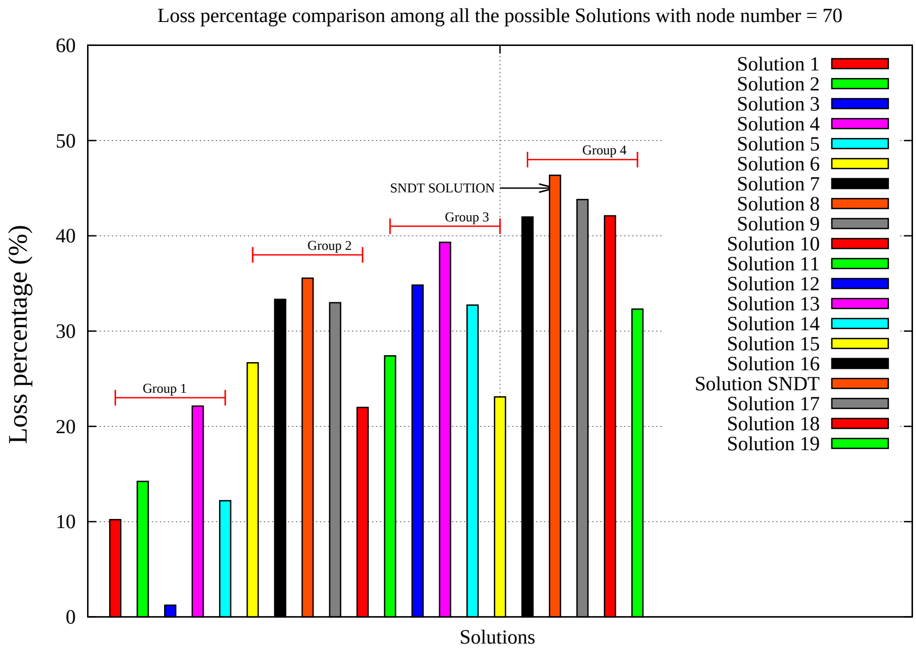


Figure 7.9: Loss performance comparison among all the possible Solutions in SNDT with node number = 70

Different communication protocol configurations also affect the loss performance. Figure 7.9 displays the loss performance comparison with 70 nodes, for different communication protocol configurations. Loss is calculated as the number of lost packets divided by the number of sent packets. The SNDT solution performs worst among all the other solutions. IEEE/802.15.4 targets low data rate and low power consumption in a small area. In this scenario, too many nodes are spread over in this fixed area. IEEE/802.15.4 suffers from the hidden terminal problem caused by the lack of RTS/CTS messages. In such cases, the SNDT solution loses 47.35% packets in this scenario. We still follow the method in the previous analysis by dividing the total 20 solutions into 4 groups according to their different MAC layer protocols.

Group 4 has the worst network accuracy overall due to the employment of IEEE/802.15.4 in this scenario. In this group, Solution 19 performs best due to the routing protocol MFlood. The flooding mechanism relays data in a WSN without the need for any routing algorithm and topology maintenance. Solution 16 loses 41.9% packets, the second best performance in this group. Solution 17 has the second worst loss performance. This can be explained because the routing protocol DSDV has large amount of routing overhead which could fail to be delivered through the network when IEEE/802.15.4 is used as the underlying protocol.

Group 1 has the smallest loss percentage overall compared with other groups. This is expected, because solutions in this group use IEEE/802.11 as the MAC layer protocol. Solution 3 performs best in this group. This is because routing protocol DSDV has an ability to maintain connection by periodic exchange of information when the MAC layer protocol is IEEE/802.11.

Group 2 performs worse than Group 1. This can be explained by the complexity of the TDMA synchronization process. Solution 10 which shares the same routing protocol MFlood as Solution 19, has the best network performance in terms of loss in this group. Solution 8 ,due to the heavy routing overhead of DSDV together with TDMA protocol, has the worst loss performance .

Solutions in Group 3 lose more packets than those in Group 1. The main reason is that the MAC layer protocol SMAC has an adaptive sleeping mechanism which could lead to more packet loss. In this group, Solution 15 performs best. Solution 11 has nearly 5% more packet loss than Solution 15. This could be explained by the routing overhead of the AODV protocol which worsens the packet loss in this scenario. Solution 13 lost 39.3% packets in this scenario because of the DSDV protocol.

It is summarised that the SNDT solution has unfavourable loss performance. Solutions with the MAC protocol IEEE/802.11 have the fewest packet losses than other solutions. Solutions with MAC protocol IEEE/802.15.4 perform worst due to scalability problems for this scenario.

The selected communication protocol set for this case is compared favourably to the other possibilities in the current SNDT system in terms of energy efficiency, End-to-end delay and loss, on the whole. The SNDT solution has the best energy efficiency, but relative poor performance in delay and loss. The user-defined application requirements are therefore successfully achieved.

7.1.4 Discussion

In section 7.1.3.1, the selected number of nodes is proven to be a fairly good choice with the given set of protocols. With this node number, the selected communication protocols are verified as the best selection among all the possible protocol combinations in section 7.1.3.2. We conclude that the solution provided by the SNDT system is favourable for this scenario. The analysis in section 7.3.2 brings out some interesting similarities among available solutions for this case.

- Solutions with MAC protocol IEEE/802.11 have good performance in terms of network latency and loss but consume a large amount of energy.

- Solutions with TDMA improves the energy consumption at the cost of introducing more delay and loss than Solutions with IEEE/802.11.
- Solutions with SMAC are energy efficient but experience long delay.
- Solutions with IEEE/802.15.4 have low power consumption but have severe packet loss.
- Solutions with routing protocol DSR enhance energy efficiency and network latency by limiting routing overhead but have the risk of losing packets.
- Solutions with routing protocol DSDV have the advantage of low network latency when the underlying MAC protocol supports this.
- Solutions with the Flooding mechanism poorly performed in terms of energy efficiency but have relatively good loss performance.
- Solutions with the AODV protocol have relative good performance in terms of end-to-end delay, loss and energy efficiency.
- Solutions with the TORA protocol have moderate performance in terms of energy efficiency, network latency but perform poorly in terms of loss.

Once the number of nodes is identified, it is easier to tell the difference among all the possible solutions in the SNDT system based on these characteristics. Given the user-defined requirements for this periodic listening scenario, we could exclude several solutions immediately, such as solutions with IEEE/802.11 which have poor performance in energy efficiency.

The SNDT solution has been shown to meet the user requirements, and to offer the best energy efficiency of any solutions, as required by the user selected parameters. In the following studies, we manually select several solutions which are believed to be reasonable solutions for each scenario and compare them with the SNDT solution. This is to reduce the considerable processing overhead required to evaluate every possible solutions. As stated above, we have shown in case study 1 that SNDT selects a good solution.

7.2 Case study 2: Emergency Detection application(Notification stage)

Emergency detection is another important type of WSN application. It is implemented by nodes that are placed at fixed or random locations throughout an environment. Such an application involves continuous monitoring of sensor nodes to detect an emergency.

A typical scenario of this kind application is comprised of two stages: a waiting(periodic listening) stage and an emergency notification stage. Each stage has its own unique characteristics and requirements. The waiting stage is similar to Case 1. It requires each node to report its status periodically to the base station. However, only a little information from each node which shows its availability needs to be transmitted in the network until the emergency is detected. Long lifetime is the first priority for this stage.

The emergency notification stage is completely different from Case 1. Successful delivery of notification packets is needed at this stage, which implies that preferably short but accurate response time is required for this stage. Energy efficiency is not really a problem.

Different preferences to the same set of performance metrics from these two stages could lead to two distinguished network solutions. The current SNDT framework can only deal with generic scenarios which have a single stage. The optimization or transition of different solutions is an issue for further research.

To emphasize the difference from Case 1, a scenario which only consider the emergency notification stage is designed for this section. The general architecture of this case is shown in Figure 7.10. In this scenario, sensor nodes are densely and randomly deployed in a surveillance area of a 100m*100m area, and a sink (base station) node is used to concentrate sensed data and forward it to a human monitoring terminal. We assume that an emergency has already been detected by a certain sensor device in this scenario. The most critical issue in this scenario is immediate response which could minimize the effects of the emergency.

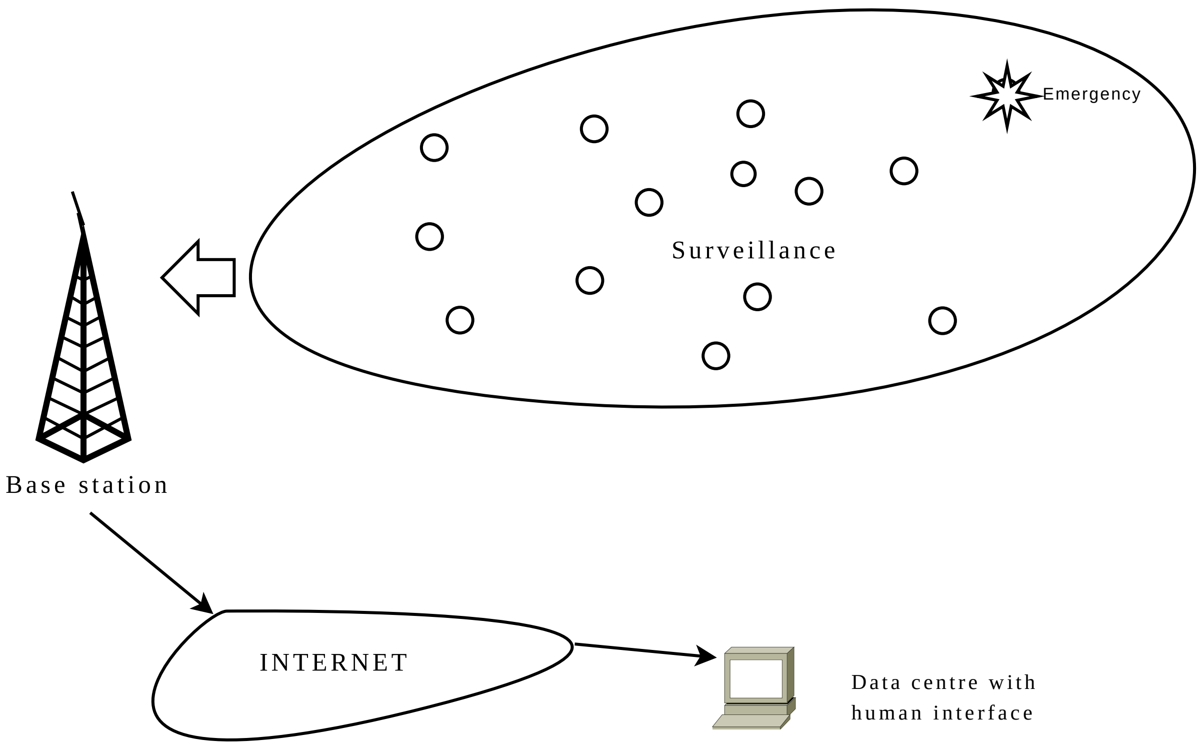


Figure 7.10: Emergency notification scenario architecture

In order to use the SNDT framework, it is necessary to take into account several assumptions for completeness.

- All the nodes are homogeneous in terms of battery capacity and transmission range.
- The network consists of up to 255 randomly deployed nodes.
- All the nodes in the network are static.
- An omnidirectional antenna is installed in each sensor node.
- Each node has a unique ID in the network.
- The data rate is low enough so that there will be no queuing delay in the sensor nodes.
- Packets communicated through the network are small and of constant size.
- Each node has a given uniform error model over incoming wireless channels with error rate equals to 0.01.
- The initial energy of each sensor device is configured to be $1000 Joule$. The power consumption model is installed in each sensor device, as shown in Table 7.6:

Table 7.6: Energy model of wireless sensor nodes

Mode of Sensor	Value ($WATT$)
Sleep Power	0.001
Idle Power	0.8
Receive Power	0.8
transmit Power	1.0
transition Power	0.2

7.2.1 Inputs and Outputs of SNDT

We use poisson traffic generation to model the traffic for this case. It is assumed that the event (packet) generation for this case is a poisson process: the time between successive events (packets) is independently generated from an exponential distribution with a given mean interval. This interval also could be interpreted as being related to the sending rate of each node. In addition, the packet size is set to 100 Bytes, since in this case it is sufficient to only transmit key information, such as the location and the emergency at the given moment. Here, the interval between two packets is set at $1s$.

The maximum number of sensor nodes that can be used in this area is 255, which means that the number of sensor nodes can be varied according to the different configuration and consequently the topology of the network. The data duration is set to $10s$, and the simulation time is set at 1000 sec.

The requirements for the system performance are as follows. The network must be able to respond to an emergency as soon as possible, which implies that message delay and loss are critical under this scenario. A short network lifetime is tolerant after an emergency is notified. This implies that energy aspect metrics are less important in this scenario. These requirements can be interpreted into a list of weight values to mark the importance of each metrics as discussed in Table 5.1 of Section 5.2.2. We set the importance weighting coefficients for end-to-end delay and Loss $w_{OWD} = 0.4$,$w_{LSS} = 0.4$ respectively. The energy consumption is the second important aspect in this scenario which leads to a lower value for the importance weighting coefficients w_{THD},w_{TFD} and w_{EPUB}. They are set to 0.05,0.05,0.1 respectively.

The GA-related parameters are set according to the analysis in section 6.2.2

as follows: the population size is selected to be 10 and the crossover rate and mutation rate are set as 0.8 and 0.08 respectively.

The inputs of the SNDT system for this case can be summarized as the Table 7.7.

Table 7.7: Inputs of SNDT for an Event Detection scenario

Application related	Traffic type	Poisson
	PacketSize	100bytes
	Packet Interval	$1s$
	Data Duration	$10s$
Target area Information	Topology	Randomly distributed
	Area range	$100m * 100m$
Performance requirements	One-way Delay	w_{OWD}=0.4
	Loss	w_{LSS}=0.4
	Time of first node died	w_{TFD}= 0.05
	Time of half nodes died	w_{HFD}= 0.05
	Energy consumption per useful bit	w_{EPUB} =0.1
GA related parameters	Population size	10
	Selection Mechanism	Stochastic Sampling
	Crossover Method	One-point Crossover
	Crossover Rate	0.8
	Mutation Method	Swap bits of each string following the p_m
	Mutation Rate	0.008

The resulting solution from the SNDT system is always shown in the form of a report. A part of this report is shown in Figure 7.11.

From the report, the resulting solution is the global best individual 01000000111000010. This shows that the best routing and MAC protocols for this scenario are DSDV and IEEE/802.11, the number of nodes in the area is 135. These are found by interpreting the corresponding binary series from the rightmost bit to their left.

```
---------------------------------------------------------------------------
                              Population Report
Generation   19                                               Generation   20
num   string            fitness    parents xsite  string            fitness
---------------------------------------------------------------------------
  1)  1101110101101000  30.701533 | ( 2, 8)   13   0100010110100100  41.442191
  2)  0100000110000111  52.177770 | ( 1, 1)    0   1111110111101001  32.330057
  3)  1110110111110000  10.598362 | ( 6, 5)   11   1011100111101000  30.701513
  4)  1111110111101001  22.330057 | ( 6, 5)   11   1111110111001010  24.343776
  5)  1000011101101000  37.186980 | ( 2, 1)   11   1111110111101001  32.330057
  6)  0111010001000110  33.126147 | ( 8, 1)   11   1111110111101000  60.701219
  7)  0100010110100101  13.394443 | ( 1, 1)    0   1111110001101000  20.669650
  8)  0101010110100100  48.203200 | ( 1, 1)    0   1011110111101000  15.401498
  9)  0110111101100000  21.691998 | ( 2, 4)    8   1111110111101000  47.201533
 10)  1111101010011101  41.054658 | ( 3, 1)    8   1111100111101000  39.401256
---------------------------------------------------------------------------
Generation 19 Accumulated Statistics:
Total Crossovers = 7, Total Mutations = 9
min = 15.401498  max = 60.701533   avg = 34.588690   sum = 345.886901
Global Best Individual so far, Generation 13:
Fitness = 73.263249: 0100000011100001 0
---------------------------------------------------------------------------
```

Figure 7.11: a part of a SDNT result report

7.2.2 Evaluation of Design decisions

As we discussed in section 7.1.4, it is unnecessary to compare all the possible solutions with the SNDT solution for a specific number of nodes. Several solutions can be easily excluded as the characteristics of the protocols contradict with the requirements of the application scenario .

In this scenario, solutions with IEEE/802.15.4 are firstly excluded as we define the operation of the network nodes such that nodes must directly communicate with the PAN coordinator. For large numbers of nodes spread over a big area, solutions with IEEE/802.15.4 experience severe packet losses which is undesired in the scenario. Besides, IEEE/802.15.4 is an energy efficient protocol favouring low data rate. In this scenario, the data rate is quite high.

Also, solutions with MAC layer protocol SMAC are not suitable for this case. The SMAC protocol employs a periodic sleeping mechanism to save energy, which imposes a serious impact on the network performance in terms of network delay .

From the routing layer aspect, solutions with TORA are excluded due to poor loss performance. Solutions with DSR protocols are not considered because it suffers from potential packet loss when the network is changing.

In the following paragraphs, we evaluate the design decisions obtained for this event detection case through simulations and compare them with several empirical solutions. The performance of each networking solution is assessed with the End-to-End delay, loss and EPUB.

Two networking solutions are selected to be compared with the SNDT result. They are shown in Table 7.8. The first one utilizes Ad-hoc On-demand Distance Vector routing(AODV) as the underlying routing protocol, Time division multiple access (TDMA) as the MAC layer protocol and 100 nodes in this $100m*100m$ area. The other solution is comprised of a Simple Flooding mechanism ,IEEE/802.11 standard and the number of nodes in this area is 150. Each of these solutions is evaluated with respect to the traffic load, here the interval between two successive packets is used to represent the traffic load. When the interval is small, the workload becomes very heavy and vice versa.

Table 7.8: SNDT solution and other two empirical solutions

Solution	Routing protocol	MAC protocol	number of nodes
Solution 1	AODV	TDMA	100
Solution 2	MFlood	IEEE/802.11	150
SNDT Solution	DSDV	IEEE/802.11	135

The first assumed solution is selected based on the following analysis: firstly, AODV [95] is the most common used routing protocol which has good performance in terms of network delay and loss in WSN simulations. It has a basic route-discovery and route-maintenance and uses hop-by-hop routing, sequence numbers and beacons. A sending node generates a route request to the destination node.

This route request is forwarded by intermediate nodes that also create a reverse route for itself from the destination node. When the route request reaches a node with a route to its destination, it will send back a route reply containing the number of hops required to reach the intended node. Similarly, all nodes that participate in forwarding this reply to the source node create a forward route to destination. TDMA [99] is a MAC layer protocol which is often used in wireless networks and has the natural advantage of collision-free medium access. It divides the use of the channel into fixed time slots and schedules the transmission of the active nodes among these time slots based on the nodes' demands and the total resources available. In the meantime, TDMA requires strict synchronization among nodes and a centralized control to coordinate the use of the channels. Consequently, it requires a large overhead in order to maintain synchronization between sensors nodes and to exchange local information, such as the network topology and the communication patterns.

Regarding to the assumed solution 2, the Flooding Mechanism was selected due to its robustness in a dense large network [100]. For the simple flooding mechanism(MFlood) considered in this work, packets are transmitted in a fashion similar to broadcasting. Every node receives and sends a packet exactly once. The MAC layer protocol IEEE/802.11 is that frequently used protocol for WSN simulations due to the good performance in terms of delay and loss. The protocol uses a scheme known as carrier-sense, multiple access, collision avoidance (CSMA/CA) which can minimize collision by using four different types of packets: request-to-send(RTS), clear-to-send(CTS), data and acknowledge(ACK) to transmit frames, in a sequential way. In CSMA, a node listens to the channel before transmitting. If it detects a busy channel, it delays access and retries later. Packets that collide are discarded and will be retransmitted later.

The simulation results bring out some important characteristic differences among the three different network solutions.

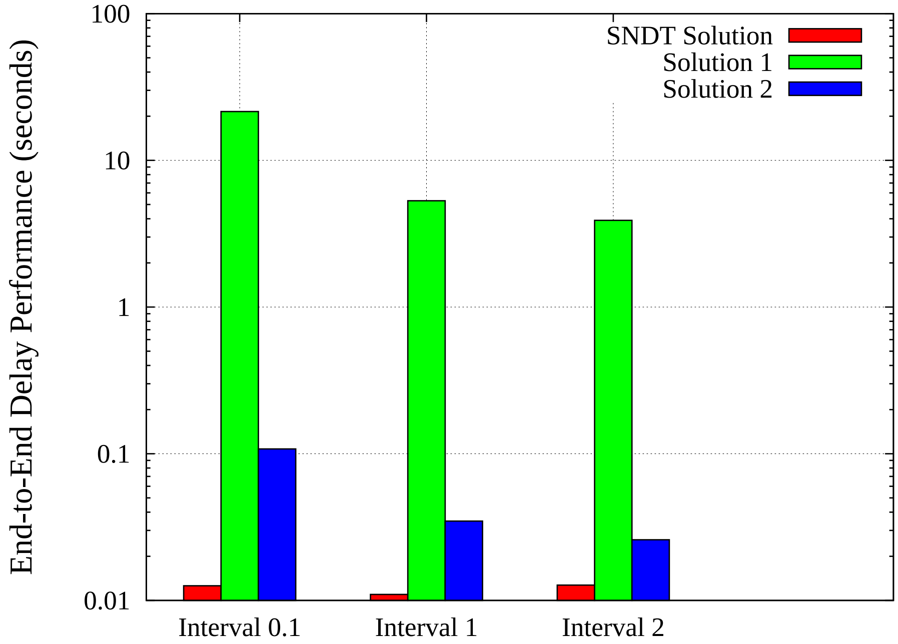


Figure 7.12: End to End Delay comparison of the SNDT solution with empirical solutions under different traffic loads

Figure 7.12 shows the simulated average End-to-End delay, for different numbers of packet interval during the monitoring period. The first result of interest is that the SNDT Solution outperforms the other two solutions in terms of end to end delay. This is because the IEEE/802.11 was selected as the MAC layer protocol in the SNDT solution. The IEEE/802.11 CSMA/CA mechanism has the ability to delivery packets very soon in all the scenarios compared with other MAC layer protocols. Also the routing layer protocol DSDV uses already established routes, which could lead to the low average end-to-end delay. Solution 1 performs worst among these three solutions for different traffic loads. This is mainly because the MAC layer protocol TDMA in Solution 1 has the problem of node synchronization. Solution 2 has much better performance compared to Solution 1 in terms of end-to-end delay, but not as good as the SNDT solution. This can be explained by the advantage of no overhead for conveying routing tables or routing information of the routing protocol-MFlood in Solution 2.

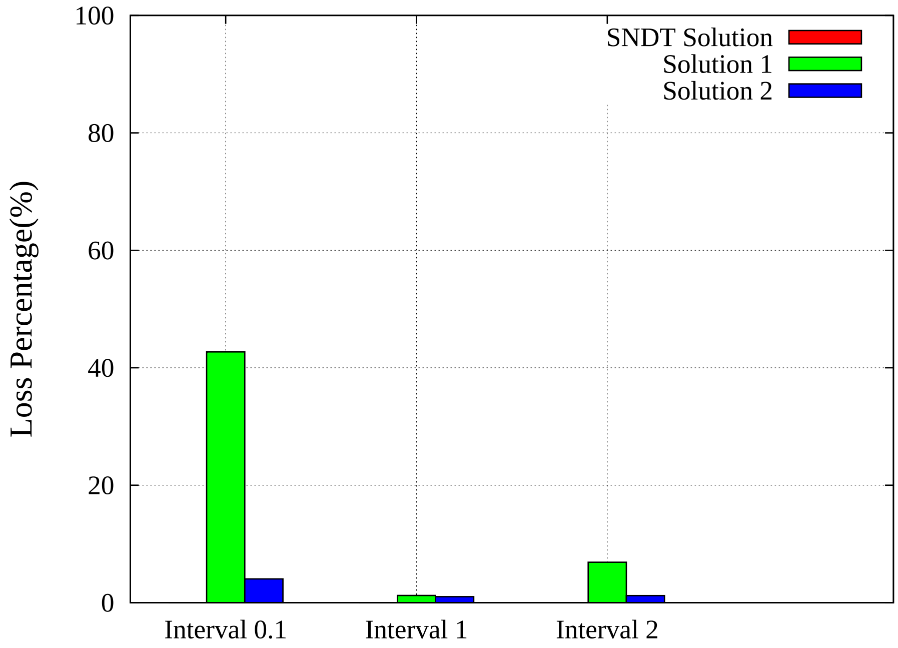


Figure 7.13: Loss percentage comparison of the SNDT solution with empirical solutions under different traffic loads

Figure 7.13 shows the variation of the loss performance for each network solution with different traffic loads. Loss is the percentage of the number of lost packets divided by the number of sent packets. It is clearly shown that the SNDT Solution has the minimum packet loss in all scenarios compared with the other two network solutions(SNDT solution has no loss). Solution 1 loses more packets than Solution 2 for all the different traffic loads. This is because of the relative high reliability of the flooding mechanism in Solution 2. On the other hand, routing protocol AODV in Solution 1 worsens the packet loss in the quite heavy traffic load scenario($interval = 0.1 packets/s$) due to the routing overhead.

Figure 7.14 illustrates the EPUB (energy consumption per useful bit) for each

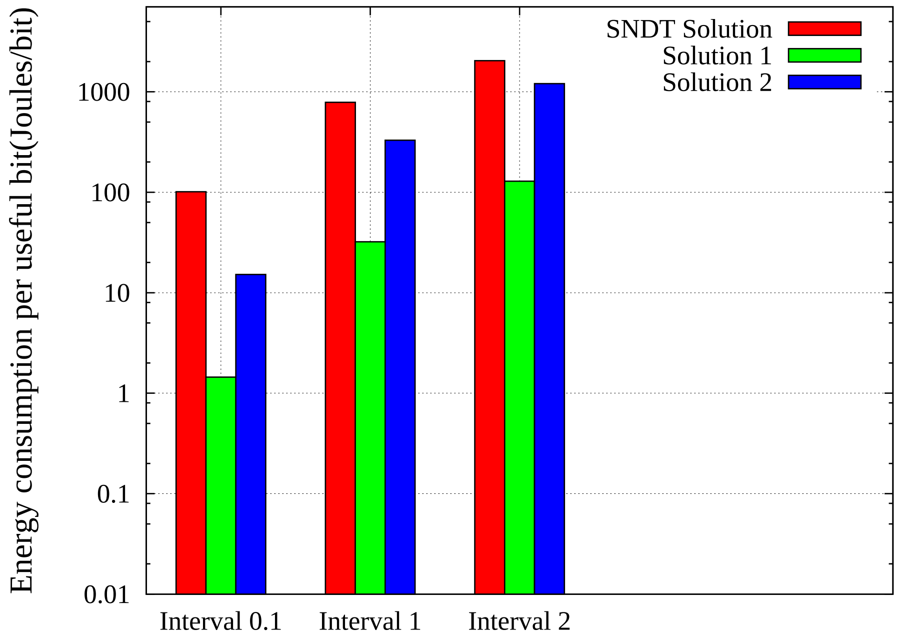


Figure 7.14: Energy consumption per useful bit comparison of the SNDT solution with empirical solutions under different traffic loads

network solution with different traffic loads. It is known that EPUB usually depends on many aspects of the network such as the routing scheme, MAC protocol, scheduling strategies, network topology etc.. The SNDT solution performs worst among these three solutions in terms of energy consumption. This is because the IEEE/802.11 protocol used in the SNDT solution has the disadvantage of energy efficiency. The overhearing avoidance mechanism requires nodes to keep active whenever an unrelated communication takes place within radio range. It wastes a huge amount of energy in idle listening when the traffic load is not heavy . Also the DSDV routing protocol uses periodic routing advertisement messages, thereby consuming more energy. Solution 1 performs best compared with other two solutions in terms of EPUB. This is mainly because the TDMA protocol has the natural advantage of collision-free medium access which requires a relative small amount of energy consumption compared with IEEE/802.11 in the SNDT solution and Solution 2. Moreover, the flooding mechanism used in Solution 2 has the problem of a large number of retransmission packets, which impose a serious impact on the EPUB.

Considering the initial requirements of this case, the solution provided by SNDT is seen to compare favourably to empirical design solutions of the WSN for this scenario. Good performance in network latency is achieved, packet loss is minimized and poor performance in energy consumption is tolerated.

7.3 Case study 3: A tracking-based application

Tracking mobile objects is one of the important application issues for wireless sensor networks. In such scenarios, the sensor network may be deployed for military use (tracking enemy vehicles, detecting illegal border crossings) and civilian purposes(tracking the movement of wild animals or monitoring a distant patient). In this section, we design a target tracking application. The scenario is the tracking of a tagged mobile object through a region of space monitored by a sensor network. The physical architecture of this scenario is shown in Fig 7.15. Node tracking applications will continually have topology changes as nodes move through the network. While the connectivity between the nodes at fixed locations will remain relatively stable, the connectivity to mobile nodes will be continually changing.

We assume that the sink node is more powerful than other sensor nodes and is always located at the border of the surveillance area. The sensor nodes are deployed for the surveillance of a $100m * 100m$ area in a lattice fashion. A mobile event is moving randomly in the WSN field with average speed equals to $20m/s$. Several assumptions are made in this scenario.

- Only one message stream is generated per target.

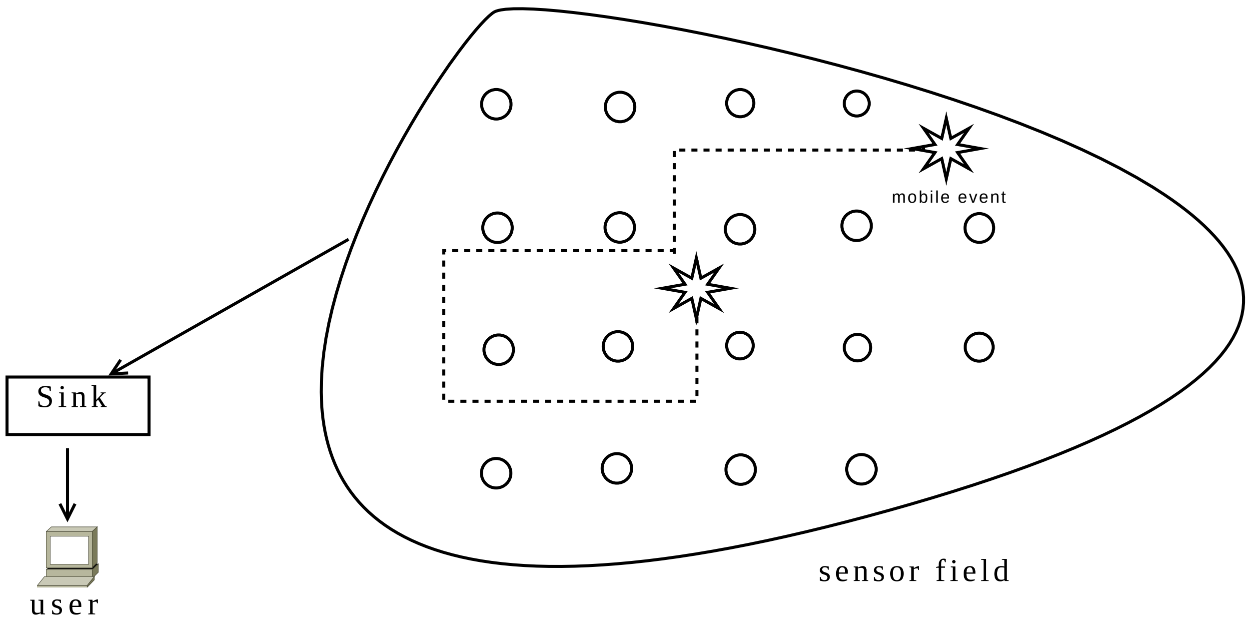


Figure 7.15: Physical architecture of a WSN for tracking-based application

- The transmission range of every node is set at the distance between two rows(or columns) of the grid. In another words, each node can only talk to the surrounding four nodes.

- The mobile target is already in the surveillance area and moving within this area without pause.

- The transport model used in this case is a UDP-like transport [101], i.e.there is not guarantee on the delivery of the data.

- The mobile target moves according to the "Random waypoint" model [102] and stops at 200s.

- In a simulation run, the source keeps monitoring the mobile target until the simulation time expired.

- Each node has a given uniform error model over incoming wireless channels with error rate equal to 0.01.

- The initial energy of each sensor device is configured to be 1000$Joules$,The power consumption model installed in each sensor device is same as it was shown in Table 7.9

Table 7.9: Energy model of wireless sensor nodes for Case 3

Mode of Sensor	Value ($WATT$)
Sleep Power	0.001
Idle Power	0.8
Receive Power	0.8
transmit Power	1.0
transition Power	0.2

7.3.1 Inputs and Outputs of SNDT

We use the executable file *setdest* in NS2 to generate an object movement file . It is assumed that the tracking object keeps moving following a path for 200s and remains stationary until the simulation terminates. So the pause time in *setdest* is set to 0s. CBR traffic is chosen to simulate the traffic generated by the target node. In addition, the packet size is set at 512 bytes. The interval between two successive packets is set to 10s.

The maximum number of sensor nodes which can be used in this area is set to 255. This means that the number of sensor nodes is varied according to the different configuration and consequently the topology of the network. The simulation time is set at 1000sec.

The requirements for the system performance are as follows. The network must be able to correctly track the mobile target, which implies that message delay is critical under this scenario. We set the importance weighting coefficient w_{OWD}

as 0.35 (in the range of $0-1$). Loss performance is also important in this case, the corresponding weighting coefficient w_{LSS} is let to 0.2. The network lifetime is better to be maximized, which leads to a great value for the importance weighting coefficients w_{THD} and w_{EPUB}. We let w_{THD} equal to 0.25, w_{EPUB} equal to 0.15 respectively. Weight coefficient w_{TFD} is set to 0.05

The inputs of the SNDT system for this tracking-based scenario can be summarized as in Table 7.10.

Table 7.10: Inputs of SNDT for the tracking-based scenario

Application related	Traffic type	Constant Byte Rate
	PacketSize	512bytes
	Packet Interval	$10s$
	Data Duration	$900s$
Movement Information	Speed	$20m/s$
	Moving range	$100m * 100m$
	Moving time	$200m$
	Pause time	$0s$
Target area Information	Topology	Uniform distributed
	Area range	$100m * 100m$
Performance requirements	One-way Delay	w_{OWD}=0.35
	Loss	w_{LSS}=0.2
	Time of first node died	w_{TFD}= 0.05
	Time of half nodes died	w_{HFD}= 0.25
	Energy consumption per useful bit	w_{EPUB} =0.15
GA related parameters	Population size	10
	Selection Mechanism	Stochastic Sampling
	Crossover Method	One-point Crossover
	Crossover Rate	0.8
	Mutation Method	Swap bits of each string following the p_m
	Mutation Rate	0.008

The resulting solution from the SNDT system is shown partly in Figure 7.16. From the report, the resulting solution is the global best individual 0000010011011110.

```
-------------------------------------------------------------------------------
                              Population Report
Generation   19                                                   Generation   20
num   string            fitness    parents xsite  string           fitness
-------------------------------------------------------------------------------
  1)  1000001110000011  23.177770 | ( 4, 5)   3   1111000111101000  41.701534
  2)  0110010110100101  31.394443 | ( 4, 5)   3   1111011101001001  25.792138
  3)  1011110010100110  25.325143 | ( 5, 1)   0   1000100111101000  33.719533
  4)  0011011101001001  34.932348 | ( 5, 1)   0   1001000110000111  37.167770
  5)  1001110101101000  22.101233 | ( 3, 7)   3   1000011100101001  17.289602
  6)  0001111001001011  27.579816 | ( 3, 7)   3   0111010001000110  23.782137
  7)  1000011101101001  11.442191 | ( 2, 2)   0   0100010110100101  22.394443
  8)  1100101100000110  26.614792 | ( 2, 2)   0   0100010110100100  38.203200
  9)  0011010001011011  21.218305 | ( 7,10)  12   0110111101100000  26.691998
 10)  1011000010011001  30.701604 | ( 7,10)  12   1001101010011101  24.054658
-------------------------------------------------------------------------------
Generation 19 Accumulated Statistics:
Total Crossovers = 7, Total Mutations = 5
min = 17.28960   max = 41.701533   avg = 29.079701   sum = 290.797010
Global Best Individual so far, Generation 17:
Fitness = 62.473042: 0000010011011110
```

Figure 7.16: A part of a SDNT result report for case 3

It shows that the best routing and MAC protocols for this scenario are AODV and SMAC, the number of nodes in this area is 123 . These are found by interpreting the corresponding binary series from the rightmost bit to the left.

7.3.2 Evaluation of Design decisions

In the following paragraphs, we evaluate the design decision obtained for this tracking-based scenario through simulations and compare them with several empirical solutions. Each solution is evaluated with respect to the traffic load and target speed. The performance of each networking solution is assessed with regards to the End-to-End delay, loss and EPUB.

Similar to Case 2, two guestimate solutions are compared with the SNDT solution. They are shown in Table 7.11. Solution 1 consists of the routing protocol DSDV, the MAC layer protocol TDMA and 150 nodes in the area. Solution 2 uses DSR as the routing protocol, IEEE/802.11 as the MAC layer protocol and 50 nodes in the area. They are selected based on the following analysis:

Table 7.11: SNDT solution and other two empirical solutions for a tracking-based scenario

Solution	Routing protocol	MAC protocol	number of nodes
Solution 1	DSDV	TDMA	150
Solution 2	DSR	IEEE/802.11	50
SNDT Solution	AODV	SMAC	123

For Solution 1, the DSDV protocol is selected as it is a proactive routing mechanism. In most cases it uses already established routes and tries to delivery packets immediately, which could lead to low average delay and loss. The underlying MAC protocol TDMA is chosen to support the periodic advertisements characteristic of the selected routing protocol. In the meantime, TDMA could prolong the network lifetime to an extent. The number of nodes in Solution 1 is set to be 150, 27 more nodes than those in the SNDT solution.

In Solution 2, by contrast to Solution 1, a smaller number of nodes are used in the area to compare the node number choice in the SNDT solution. The routing protocol DSR is chosen because of its good performance for network latency. The DSR protocol has a unique advantage by virtue of source routing. For the MAC layer protocol, IEEE/802.11 is selected to meet the network requirements in terms of network delay and loss.

We ran simulations with three different target average moving speed: slow $10m/s$, medium $20m/s$, fast $30m/s$ (Note the Case 3 target speed is medium $20m/s$). This means to examine the compatibility of SNDT solution for other scenarios with different target speeds(slow and fast speed). The performance comparison in terms of End-to-End delay, loss and EPUB was carried out with each target moving speed. The simulation results are analyzed in the following paragraphs.

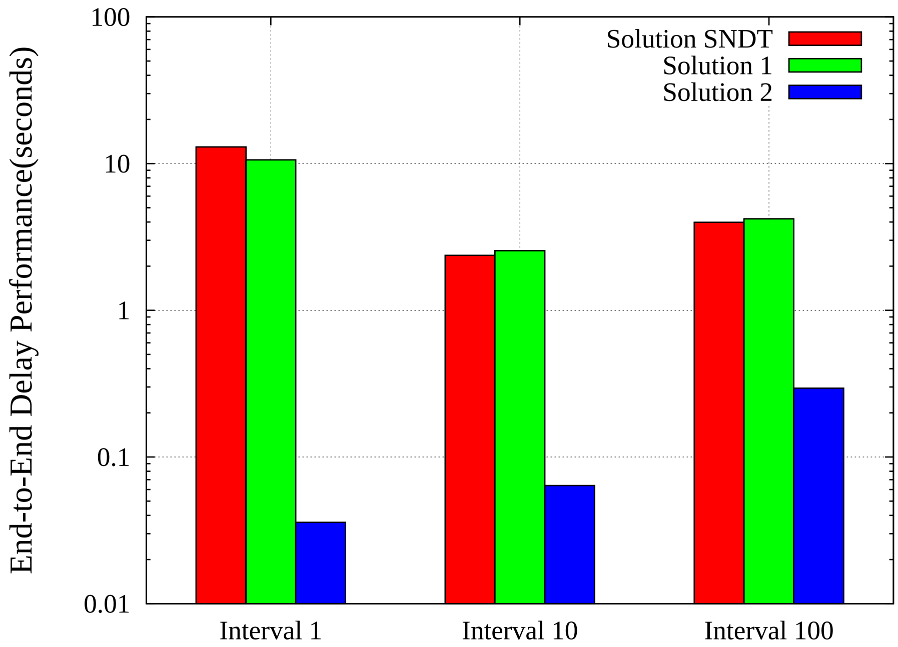


Figure 7.17: End to End Delay comparison of the SNDT solution with two empirical solutions under different traffic loads with mobile target speed $10m/s$

Figure 7.17, Figure 7.18 and Figure 7.19 show the comparison of average end-to-end delay for the two empirical solutions and the SNDT solution under different traffic loads for the slow, medium and fast target moving speed respectively.

For all the different target speed scenarios, the Solution 2 has the smallest end-to-end delay under different traffic loads. The result matches the theoretic estimation that the MAC layer protocol IEEE/802.11 has the best network latency performance. For the slow speed scenarios, the SNDT Solution performs slightly worse than Solution 1 under heavy traffic(i.e. $interval = 1packet/s$). This is expected, as the SMAC protocol in SNDT solution could lead to longer delay due to the scheduled sleeping mechanism. In the case of less heavier traffic scenarios (i.e. $interval = 10packets/s$ or $interval = 100packets/s$), the SNDT solution has a small advantage in network latency compared with Solution 1. This is because

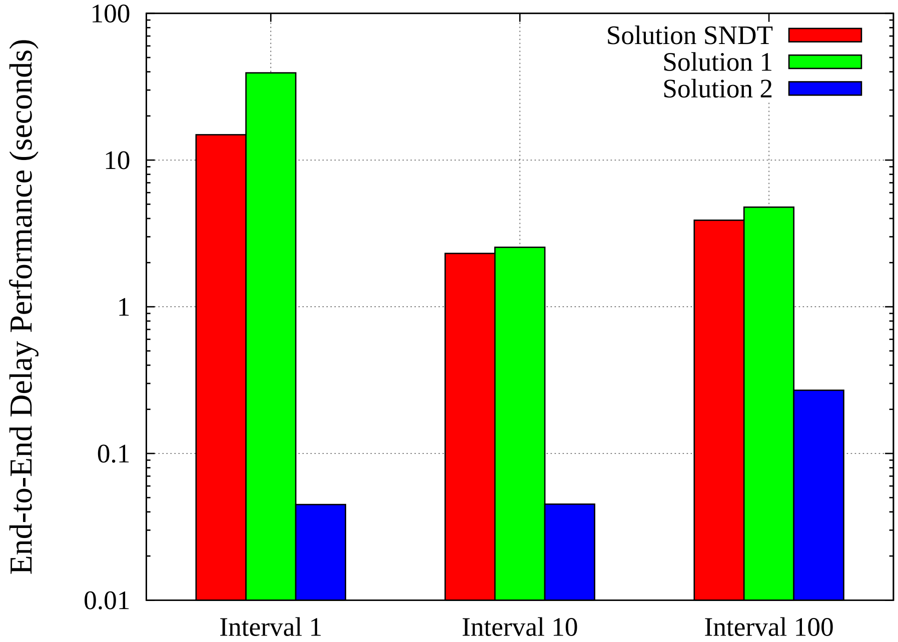


Figure 7.18: End to End Delay comparison of the SNDT solution with two empirical solutions under different traffic loads with mobile target speed $20m/s$

the synchronization of a large number of nodes of Solution 1 is required by the MAC protocol TDMA. The huge amount of overhead introduced by adaptation to the changing topology in the TDMA protocol worsens the performance of the Solution 1 in terms of network latency .

For Case 3 scenarios(medium speed scenarios), the SNDT Solution performs better than Solution 1 under all different traffic loads in terms of end-to-end delay as indicated in Figure 7.18. This is can be explained by the good routing efficiency of the AODV protocol in the SNDT Solution in the case of a quite mobile scenario. On the other hand, the heavily changed network topology challenges the TDMA protocol in Solution 1. Routing overhead introduced by the DSDV protocol is another factor which leads to poor network latency performance for Solution 1. However, the advantage becomes smaller when the traffic loads lessen. In the case of fast moving target scenarios(as shown in Figure 7.19), the variation of

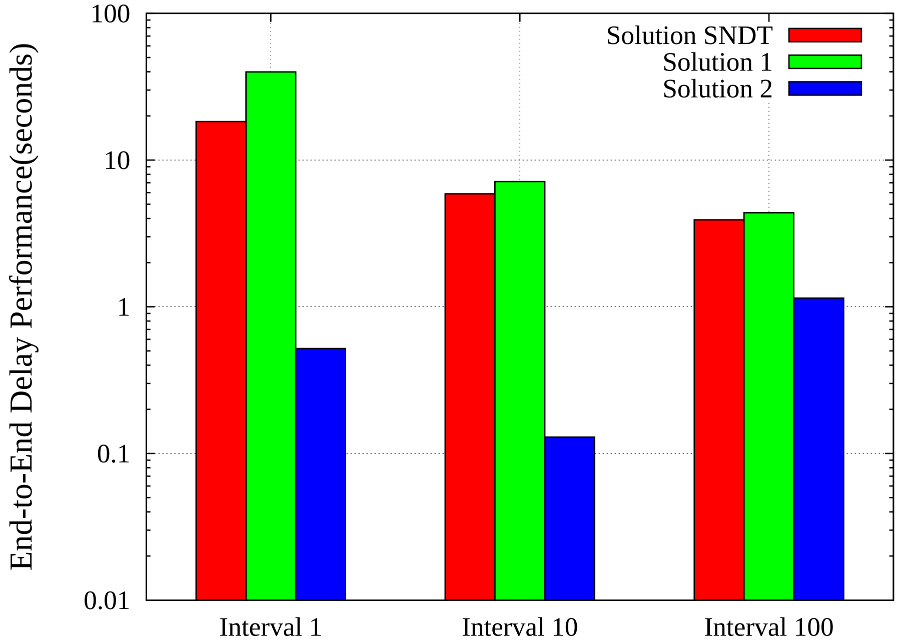


Figure 7.19: End to End Delay comparison of the SNDT solution with two empirical solutions under different traffic loads with mobile target speed $30m/s$

performance of end-to-end delay with traffic load is similar to the variation in performance in the medium moving target scenarios.

It can be summarized that the SNDT solution has a relatively good performance in network latency for case 3 as indicated in Figure 7.18. As shown in Figure 7.17 and 7.19, the SNDT solution also performs well when the traffic load changes in the medium and fast moving target scenarios in terms of end-to-end delay.

Figure 7.20, Figure 7.21 and Figure 7.22 show the loss comparison of the SNDT solution and two empirical solutions under different traffic loads for the slow ,medium and fast target moving speeds respectively.

Similar to the end-to-end delay performance, Solution 2 outperforms the other two solutions in terms of loss percentage for all the different target speed scenarios. This is expected, as the MAC layer protocol in Solution 2 is IEEE/802.11.

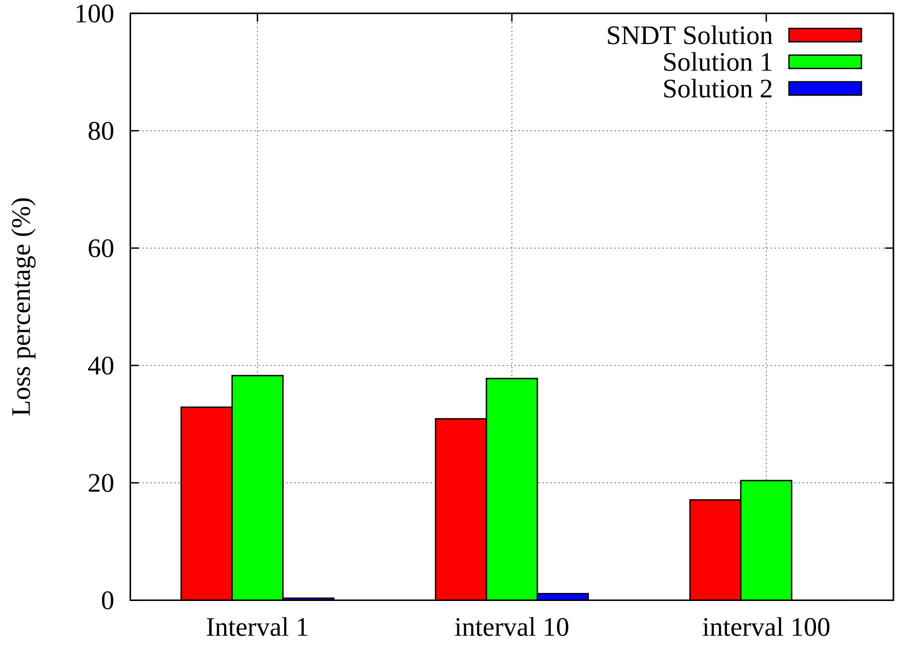


Figure 7.20: Loss comparison of the SNDT solution with two empirical solutions under different traffic loads with mobile target speed $10m/s$

For the slow speed scenarios, the SNDT Solution has better performance for loss percentage than the Solution 1 under the heavy traffic loads circumstance (i.e. $interval = 1packet/s$). This is can be inferred from Figure 7.20. This is mainly because of the synchronization drawback of the TDMA protocol in the Solution 1 for the changing topology. As the number of sensor nodes is increased, the behaviour of the physical layer and its interactions with the MAC layer are more complex. When the traffic loads decreases, all the solutions lose less packets. The SNDT Solution still loses fewer packets than Solution 1.

For the Case 3 scenarios(medium speed scenarios), the SNDT Solution loses a bigger number of packets compared to Solution 1 under heavy traffic loads as indicated in Figure 7.21. When the traffic loads become medium(i.e. $interval = 10packets/s$), the SNDT Solution lost 28.94% packets, slightly better than Solution 1. The advantage is then kept when the traffic loads turns light (i.e.

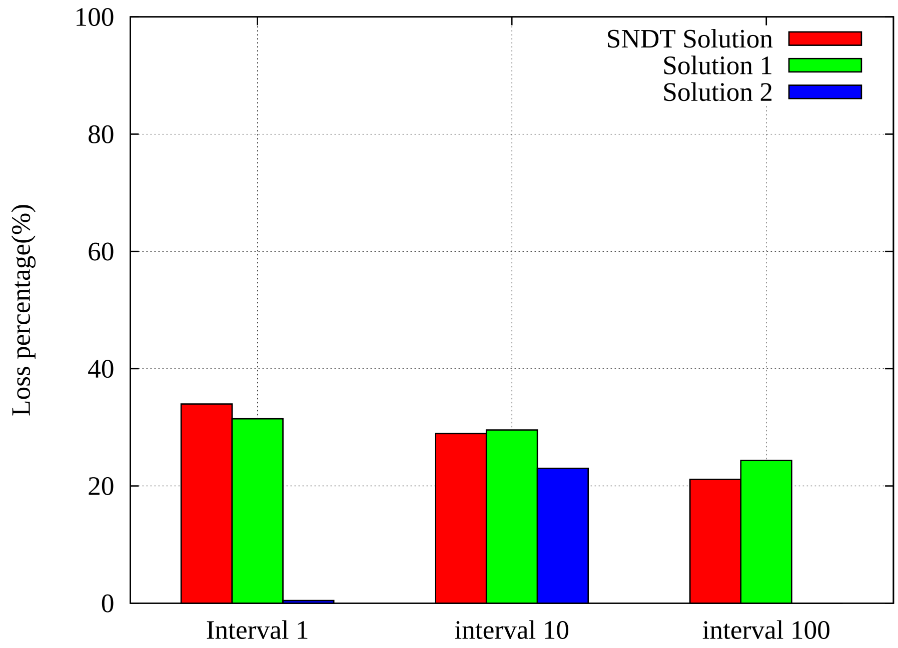


Figure 7.21: Loss comparison of the SNDT solution with two empirical solutions under different traffic loads with mobile target speed $20m/s$

$interval = 100packets/s$). Although the SMAC protocol puts nodes into sleep mode periodically, the reactive routing scheme AODV cooperates well by sending small routing overheads.

The SNDT Solution performs worst among these three solutions under different traffic loads with the mobile target speed equals to $30m/s$ as shown in Figure 7.22. A large number of overheads have to be transferred in the network to deal with the fast changing topology. The SMAC protocol in SNDT Solution begins to lose more packets under this circumstance. The dynamic topology also presents big challenges for the other two solutions.

It is clear that SNDT solution has a relatively good loss performance for Case 3. As indicated in Figure 7.20- 7.22, the SNDT solution is no longer favourable to scenarios with a fast moving target.

The energy performance comparison of the SNDT solution and other two em-

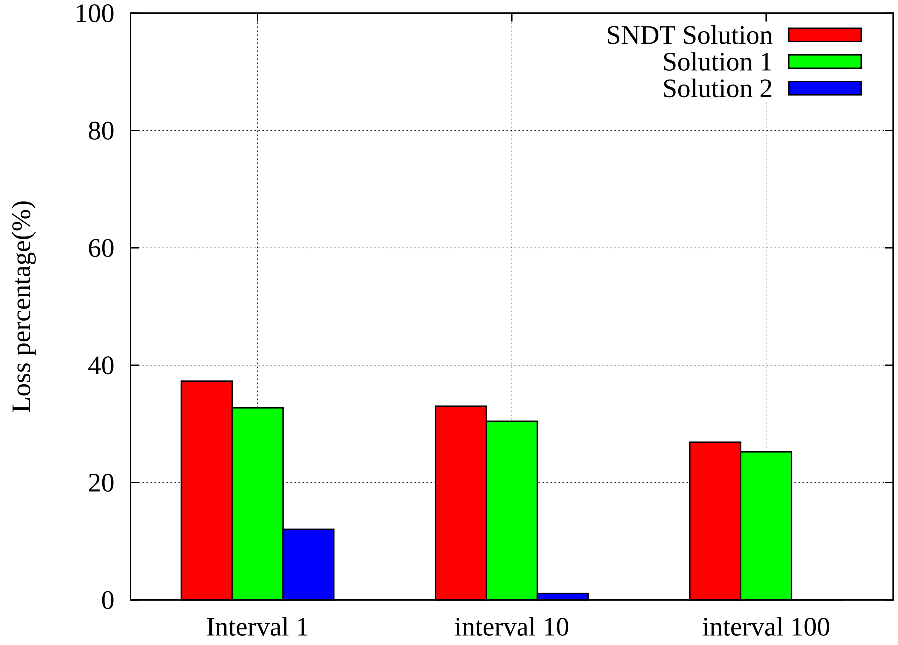


Figure 7.22: Loss comparison of the SNDT solution with two empirical solutions under different traffic loads with mobile target speed $30m/s$

pirical solutions under different traffic loads in the slow, medium and fast target moving speed are displayed in Figure 7.23, Figure 7.24 and Figure 7.25 respectively. The variation of energy consumption per useful bit to different traffic loads is plotted in each figure.

For all the different target speed scenarios, the SNDT Solution has the smallest energy consumption per useful bit under different traffic loads. This is can be explained by the good energy efficiency of the SMAC protocol in the SNDT Solution. The energy consumption caused by idle-listening is reduced and time synchronization overhead is prevented by sleep schedules messages.

Solution 2 has the worst performance in terms of EPUB for all the different target speed scenarios with different traffic loads. This is expected, as the underlying MAC protocol IEEE/802.11 wastes a huge amount of energy in idle listening. Solution 1 performs better than Solution 2, but not as good as the SNDT Solution.

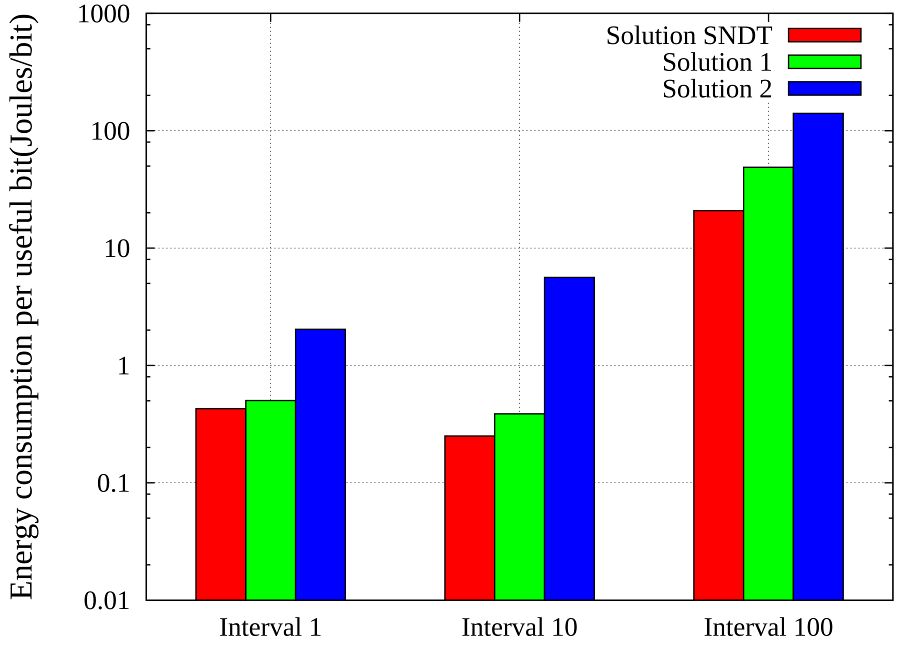


Figure 7.23: Energy consumption per useful bit comparison of the SNDT solution with two empirical solutions under different traffic loads with mobile target speed $10m/s$

This result matches the estimate that TDMA due to the collision-free medium access consumes less than IEEE/802.11 protocol.

It is concluded that SNDT solution meets the energy requirements of Case 3. It has the smallest energy consumption per useful bit for all the different target speed scenarios. Solution 2 outperforms the SNDT solution in terms of network delay and loss, but it fails to meet the energy requirements.

7.4 Summary

In this chapter, we performed three detailed case studies to demonstrate and assess how the SNDT framework operated. In order to show the effect of varying performance requirements to the choice of a certain network solutions, three specific

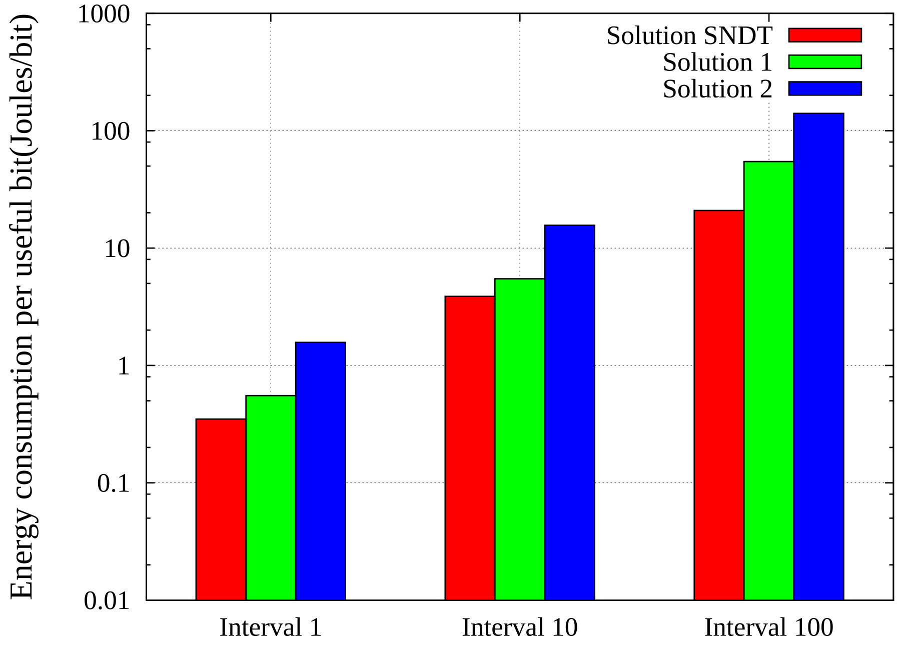


Figure 7.24: Energy consumption per useful bit comparison of the SNDT solution with two empirical solutions under different traffic loads with mobile target speed $20m/s$

scenarios : a periodic-measuring application, an event detection type of application and a tracking-based application were designed. The results from SNDT were validated through simulations.

The result produced by SNDT consists of a set of communication protocols and the number of nodes in a given area. It is impossible to compare the SNDT solution with all the others available in the SNDT framework, as it has 2^{16} possibilities. For Case 1, we verified the choice of number of nodes by varying node number with the same communication protocols. This number was then used to help identifying the goodness of the set of communication protocols in the SNDT solution. The SNDT solution had been shown to meet the user requirements, and to offer the best energy efficiency of any solution, as required by the user selected parameters.

Several interesting similarities among available solutions are observed in case

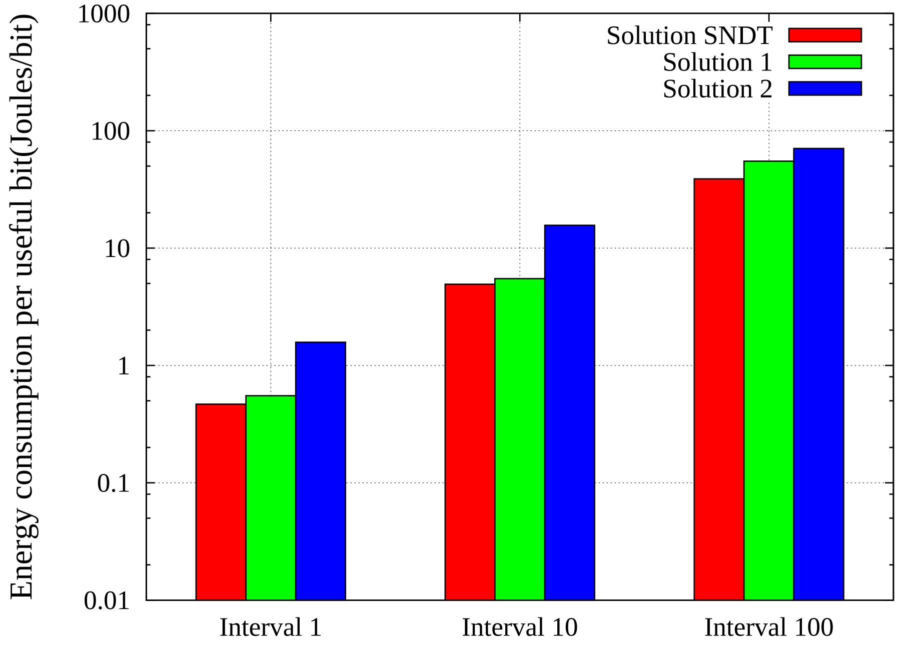


Figure 7.25: Energy consumption per useful bit comparison of the SNDT solution with two empirical solutions under different traffic loads with mobile target speed $30m/s$

study 1. These characteristics were used in the next two case studies to reduce the considerable processing overhead required to evaluate every possible solutions. We then deduced two different empirical solutions to compare with the SNDT solution in Case 2 and Case 3.

Considering the initial requirements of each case, the solutions provided by SNDT were proven to be favorable in terms of energy consumption, end-to-end delay and loss. User-defined application requirements were successfully achieved.

Chapter 8

Conclusion and Future Work

8.1 Conclusion

The emerging field of wireless sensor networks combines sensing, computation, and wireless communication into a single tiny device. They collaborate with each other to perform sensing tasks in a given environment. WSNs have a huge range of applications starting from personal area to environment monitoring. These highly diverse scenarios impose different requirements on WSN design decisions. With strict energy constraints and application-specific characteristics, WSNs distinguish themselves from other traditional networks. And these two aspects present major configuration challenges for a WSN designer.

A huge number of new communication protocols considering the WSNs' features are designed to meet the heterogeneous application requirements. It leads to the fact that a common set of protocols that will suit all applications do not exist. Properly designing the WSN before deployment is crucial and involves trades-off between many competing goals.

This thesis has presented a WSN design framework which could provide the whole pack solution based on expert knowledge. By utilizing a GA as the optimization tool, the framework can automately select an appropriate set of protocoals out of hundreds of options and optimize the number of nodes in a fixed area given a certain application scenario.

Our approach that optimizes the WSN design at the system level was motivated by current WSN design optimization methods. Optimization in the communication layers, node hardware optimization and cross-layer optimization had been discussed in a progressive way. Most of these methods only optimize a certain objective or a few objectives. Our optimization approach which takes into account the princels, characteristics and requirements of an application-specific WSN at the system level has been developed in this thesis.

The optimization choice of Genetic Algorithms has been discussed as well. Taking the user-defined requirements as out starting point, the approach has to deal with many nonlinear design criteria simultaneously. Thus, the focus of the problem is how to find a near-optimal non-dominated solution in a practically acceptable time. GAs try to imitate natural evolution by assigning a fitness value to each candidate solution of the problem and by applying the principle of survival of the fittest. Genetic algorithms have shown their capability for solving complex searching problems in many research areas and has been successfully applied to WSNs to meet a certain design objective.

The work described in this thesis proposes a GA-based design framework which aids sensor network designers in system performance tuning before a network is practically deployed. This framework enables users to find a good set of the protocols out of hundreds of configuration possibilities as well as the number of nodes in a fixed area for a certain application scenario before physical deployment.

The proposed framework automates the WSN design procedure by using a genetic algorithm. The offline procedure is achieved by identifying the acceptable performance of a WSN under different configurations for a certain scenario. A well-informed performance function considering network connectivity, application specific requirements and energy conservation, has been derived to represent a WSN operation. This performance function could cover most of the fundamental characteristics of WSNs' representative scenarios which have been investigated in Chapter 5.

The proposed framework takes into consideration the application-specific requirements, scenario informations and GA configurations to automate the design procedure for a good network solution. To design and build this framework, the foundation provided by the simulation engine must be configured at the system level. Considering the simulation constraints in WSNs, the underlying simulator was NS2 as it is not only easily extended but is also capable of analyzing network performance by capturing performance measures.

We demonstrate the usage of proposed framework through detailed case studies. Three specific cases : a periodic-measuring application, the notification stage of an event detection type of application and a tracking-based application were analysed. The results from the proposed system compare favourably to possible solutions.

The SNDT framework is targeted as a design system which can provide optimal solutions based on expert knowledge and can be easily used by less experienced users. Although it suffers limitations from many aspects, the proposal of this framework automates the design optimization procedure of a WSN and allows non-expect users to design their WSNs according to different application requirements.

The runtime of this framework depends on several factors: GA parameters(number of population size and generations) and the target application itself. For a simple periodic-measuring application of 10 generations with population 10, it took about 14 hours running on a Low performance machine. The optimization process for current applications may last ranging from hours to several days, which is bearable at moment. When optimizing a scenario involving a long simulation time, the system could be changed by selecting protocols from empirical ones. It could reduce the size of iterations of each measure cycle, also save the GA runtime.

8.2 Future Work

In the future, we would like to extend the proposed framework in many aspects:

- In chapter 7, only three simple single-phase wireless sensor network scenarios are considered. They are a periodic measurement scenario, the notification stage of an emergency detection and a tracking scenario. Many real-life applications may contain two or more generic phases. Using these three different scenarios for evaluating the proposed framework can be expanded in future work to support other complex scenarios. Research topics such as optimization of two or three sets of networking solutions for a certain application, development of new protocols based on existing protocols for a certain multi-phases applications, would be worthy of investigating.

- Adding features that make the resulting solution more comprehensive: more configuration attributes such as error handling mechanisms, environmental effects,topology etc. are worthy of consideration in the future.

- Optimizing the performance of the GA: Current SNDT framework employs routine roulette wheel selection mechanism. Other selection mechanisms such as stochastic remainder and tournament selection can be considered in the future for the best GA performance.

- Protocol constraints: NS2 is not designed for wireless sensor network analysis. The limited number of WSN protocol modules in NS2 challenges the design of this work, although many protocol modules are introduced in the NS code. More WSN protocols which are well programmed and compatible with existing protocols need to be added in the future. If a new simulator which could be configured at the system level with more available WSN protocols became available, the framework could change its underlying simulator to the new one.

8.3 Publications based on part of this thesis

- J Fan, D.J.Parish,"SNDT: a genetic algorithm-based Protocol Selection Tool for Wireless Network Design", Proceedings of the IEEE UKIWCWS on Cognitive Wireless Systems, IIT Delhi, India.(To be published in Dec of 2009)

- J Fan, D.J.Parish, "Optimization of Wireless Sensor Networks Design using a Genetic Algorithm", Proceedings of the IASTED International Symposium on Parallel and Distributed Computing and Systems, PDCS 2008, Orlando, Florida, USA November 2008, pp 399-405, ISBN 978-0-88986-773-4

- J Fan, D.J.Parish,"Using a Genetic Algorithm to Optimize the Performance of a Wireless Sensor Network", Proceedings of Post Graduate Network, PG Net2007, Liverpool John Moors University,Liverpool, June 2007, pp 1-6, ISBN 1-9025-6016-7

- J Fan, D.J.Parish, "SNDT: A genetic algorithm based Sensor Network Design Tool", Multiservice Network, MSN 2009 Coseners, July 2009

References

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *Communications Magazine, IEEE*, 40(8):102–114, Aug 2002. ISSN 0163-6804.

[2] F. Silva, J. Heidemann, R. Govindan, and D. Estrin. Directed diffusion. *Frontiers in Distributed Sensor Networks*, pages 573–596, 2003.

[3] A.Cerpa, J. Elson, D. Estrin, L. Girod, M. Hamilton, and J.Zhao. Habitat monitoring: application driver for wireless communications technology. *SIGCOMM Comput. Commun. Rev.*, 31(2 supplement):20–41, 2001. ISSN 0146-4833.

[4] R. Szewczyk, E. Osterweil, J. Polastre, M. Hamilton, A. Mainwaring, and D. Estrin. Application driven systems research: Habitat monitoring with sensor networks. *Communications of the ACM Special Issue on Sensor Networks*, 2004.

[5] E. Biagioni and K. Bridges. The application of remote sensor technology to assist the recovery of rare and endangered species. *International Journal of High Performance Computing Applications*, 16(3).

[6] D.M. Doolin and N. Sitar. Wireless sensors for wildfire monitoring. In *Proceedings of SPIE*, volume 5765, page 477. SPIE, 2005.

[7] E.M. Petriu, N.D. Georganas, D.C. Petriu, D. Makrakis, and V.Z. Groza. Sensor-based information appliances. *Instrumentation & Measurement Magazine, IEEE*, 3(4):31–35, Dec 2000. ISSN 1094-6969.

[8] University of Massachusetts at Amherst. The intelligent home project. `http://dis.cs.umass.edu/research/ihome/`, March 2008.

[9] A. Milenkovic, C. Otto, and E. Jovanov. Wireless sensor networks for personal health monitoring: Issues and an implementation. *Computer Communications*, 29(13-14):2521–2533, 2006.

[10] C.Y. Chong and S.P. Kumar. Sensor networks: evolution, opportunities, and challenges. *Proceedings of the IEEE*, 91(8):1247–1256, 2003.

[11] Intel. Intel sensor networks research. `http://techresearch.intel.com/articles/Exploratory/1501.htm`, March 2008.

[12] S.Y. Cheung, S. Coleri, B. Dundar, S. Ganesh, C.W. Tan, and P. Varaiya. Traffic measurement and vehicle classification with single magnetic sensor. *Transportation Research Record*, 1917(-1):173–181, 2005.

[13] S. Bapat, V. Kulathumani, and A. Arora. Analyzing the Yield of ExScal, a Large-Scale Wireless Sensor Network Experiment. In *Thirteenth IEEE International Conference on Network Protocols (ICNP 2005)*, 2005.

[14] Crossbow Corp. Crossbow's products overview. `http://www.xbow.com/`, October 2008.

[15] Crossbow mica2 datasheet. `http://www.xbow.com/Products/productdetails.aspx?sid=174`, October 2009.

[16] D. Cook and S. Das. *Smart Environments: Technology, Protocols and Applications (Wiley Series on Parallel and Distributed Computing)*. Wiley-Interscience, 2004.

[17] S Ghiasi, Srivastava A., X. Yang, and M. Sarrafzadeh. Optimal energy aware clustering in sensor networks, 2002.

[18] C. Zhou and B. Krishnamachari. Localized topology generation mechanisms for wireless sensor networks. In *Global Telecommunications Conference, 2003. GLOBECOM'03. IEEE*, volume 3, 2003.

[19] J. Elson and A. Parker. Tinker: A tool for designing data-centric sensor networks. In *Information Processing in Sensor Networks, 2006. IPSN 2006. The Fifth International Conference on*, pages 350–357, 2006.

[20] R. Ramadan, K. Abdelghany, and H. El-Rewini. SensDep: a design tool for the deployment of heterogeneous sensing devices. In *Second IEEE Workshop on Dependability and Security in Sensor Networks and Systems, 2006. DSSNS 2006*, page 10, 2006.

[21] V. Prasad, T. Yan, P. Jayachandran, Z. Li, S.H. Son, J.A. Stankovic, J. Hansson, and T. Abdelzaher. ANDES: an ANalysis-based DEsign tool for wireless Sensor networks. In *Proceedings of the 28th IEEE International Real-Time Systems Symposium (RTSS)*, 2007.

[22] W. Ye, J. Heidemann, and D. Estrin. An energy-efficient MAC protocol for wireless sensor networks. In *INFOCOM 2002. Twenty-First Annual Joint*

Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, volume 3, 2002.

[23] W. Ye, J. Heidemann, and D. Estrin. Medium access control with coordinated adaptive sleeping for wireless sensor networks. *Networking, IEEE/ACM Transactions on*, 12(3):493–506, 2004.

[24] T. van Dam and K. Langendoen. An adaptive energy-efficient MAC protocol for wireless sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 171–180. ACM Press New York, NY, USA, 2003.

[25] M.I. Brownfield, A.S. Fayez, T.M. Nelson, and N. Davis IV. Cross-layer wireless sensor network radio power management. In *Proceedings of the International Wireless Communications and Networking Conference (WCNC*, volume 2, pages 1160–1165.

[26] E. Callaway, P. Gorday, L. Hester, JA Gutierrez, M. Naeve, B. Heile, and V. Bahl. Home networking with IEEE 802.15.4: a developing standard for low-rate wireless personal area networks. *Communications Magazine, IEEE*, 40(8):70–77, 2002.

[27] Wiki. Zigbee standard. `http://en.wikipedia.org/wiki/Zigbee`, October 2008.

[28] G. Lu, B. Krishnamachari, and C.S. Raghavendra. An Adaptive Energy-Efficient and Low-Latency MAC for Data Gathering in Wireless Sensor Networks. In *Proceedings of 18th International Parallel and Distributed Processing Symposium, Pages*, volume 224, pages 26–30, 2004.

[29] JN Al-Karaki and AE Kamal. Routing techniques in wireless sensor networks: a survey. IEEE Wireless Communications Magazine, Volume 11, Pages 6–28, 2004.

[30] K. Akkaya and M. Younis. A survey on routing protocols for wireless sensor networks. *Ad Hoc Networks*, 3(3):325–349, 2005.

[31] W.R. Heinzelman, J. Kulik, and H. Balakrishnan. Adaptive protocols for information dissemination in wireless sensor networks. In *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, pages 174–185. ACM New York, NY, USA, 1999.

[32] J. Kulik, W. Heinzelman, and H. Balakrishnan. Negotiation-Based Protocols for Disseminating Information in Wireless Sensor Networks. *Wireless Networks*, 8(2):169–185, 2002.

[33] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed diffusion: a scalable and robust communication paradigm for sensor networks. In *Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 56–67. ACM New York, NY, USA, 2000.

[34] J. Heidemann, F. Silva, and D. Estrin. Matching data dissemination algorithms to application requirements. In *Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 218–229. ACM New York, NY, USA, 2003.

[35] Y. Yu, R. Govindan, and D. Estrin. Geographical and energy aware routing: A recursive data dissemination protocol for wireless sensor networks. *UCLA Computer Science Department Technical Report, UCLA-CSD TR-01-0023, May*, 2001.

[36] B. Karp and HT Kung. GPSR: greedy perimeter stateless routing for wireless networks. In *Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 243–254. ACM Press New York, NY, USA, 2000.

[37] WR Heinzelman, A. Chandrakasan, H. Balakrishnan, and C. MIT. Energy-efficient communication protocol for wireless microsensor networks. In *System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on*, page 10, 2000.

[38] S. Lindsey and CS Raghavendra. PEGASIS: Power-efficient gathering in sensor information systems. In *Aerospace Conference Proceedings, 2002. IEEE*, volume 3, 2002.

[39] K. Sohrabi, J. Gao, V. Ailawadhi, and GJ Pottie. Protocols for self-organization of a wireless sensor network. *Personal Communications, IEEE [see also IEEE Wireless Communications]*, 7(5):16–27, 2000.

[40] K. Akkaya and M. Younis. An Energy-Aware QoS Routing Protocol for Wireless Sensor Networks. In *the Proceedings of the IEEE Workshop on Mobile and Wireless Networks (MWN 2003), Providence, Rhode Island, May*, 2003.

[41] M. Younis, M. Youssef, and K. Arisha. Energy-Aware Routing in Cluster-Based Sensor Networks. In *Proceedings of the 10th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems (MASCOTS'02)*, page 129. IEEE Computer Society Washington, DC, USA, 2002.

[42] T. He, J.A. Stankovic, C. Lu, and T. Abdelzaher. SPEED: A Stateless Protocol for Real-Time Communication in Sensor Networks. VIRGINIA UNIV CHARLOTTESVILLE DEPT OF COMPUTER SCIENCE, 2005.

[43] A. Perrig, R. Szewczyk, JD Tygar, V. Wen, and D.E. Culler. SPINS: Security Protocols for Sensor Networks. *Wireless Networks*, 8(5):521–534, 2002.

[44] C. Karlof and D. Wagner. Secure routing in wireless sensor networks: attacks and countermeasures. *Ad Hoc Networks*, 1(2-3):293–315, 2003.

[45] KP Ferentinos and TA Tsiligiridis. Heuristic dynamic clustering in wireless sensor networks towards uniform sensing. 15th IST Mobile & Wireless Communications Summit, Myconos, Greece, June, 2006.

[46] Eduardo F. Nakamura, Antonio A. F. Loureiro, and Alejandro C. Frery. Information fusion for wireless sensor networks: Methods, models, and classifications. *ACM Comput. Surv.*, 39(3):9, 2007. ISSN 0360-0300.

[47] C. Alippi and G. Vanini. Application-based routing optimization in static/semi-staticWireless Sensor Networks. In *Pervasive Computing and Communications, 2006. PerCom 2006. Fourth Annual IEEE International Conference on*, pages 47–51, 2006.

[48] S. Okdem and D. Karaboga. Routing in Wireless Sensor Networks Using Ant Colony Optimization. In *the Proceedings of the First NASA/ESA Conference on Adaptive Hardware and Systems*, volume 404. IEEE Press, 2006.

[49] F. Bouhafs, M. Merabti, and H. Mokhtar. A node recovery scheme for data dissemination in wireless sensor networks. *Communications, 2007. ICC '07. IEEE International Conference on*, pages 3876–3881, June 2007.

[50] M. Ali and Z. A. Uzmi. An energy-efficient node address naming scheme for wireless sensor networks. In *IEEE International Networking and Communications Conference (INCC'04)*, 2004.

[51] I. Marín, E. Arceredillo, A. Zuloaga, and J. Arias. Wireless Sensor Networks: A Survey on Ultra-Low Power-Aware Design. *Transactions on Engineering, Computing and Technology*, 8:44–49, 2005.

[52] L. Benini, A. Bogliolo, and G. De Micheli. A survey of design techniques for system-level dynamic powermanagement. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 8(3):299–316, 2000.

[53] J. Pouwelse, K. Langendoen, and H. Sips. Dynamic voltage scaling on a low-power microprocessor. In *Proceedings of the 7th annual international conference on Mobile computing and networking*, pages 251–259. ACM Press New York, NY, USA, 2001.

[54] S.P. Mohanty, N. Ranganathan, and V. Krishna. Datapath Scheduling using Dynamic Frequency Clocking. In *Proceedings of the IEEE Computer Society Annual Symposium on VLSI*, pages 65–70, 2002.

[55] M.T. Schmitz, B.M. Al-Hashimi, and P. Eles. Iterative schedule optimization for voltage scalable distributed embedded systems. *ACM Transactions on Embedded Computing Systems (TECS)*, 3(1):182–217, 2004.

[56] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin. Highly-resilient, energy-efficient multipath routing in wireless sensor networks, 2001.

[57] B. Krishnamachari, Y. Mourtada, and S. Wicker. The energy-robustness tradeoff for routing in wireless sensor networks. In *Communications, 2003. ICC'03. IEEE International Conference on*, volume 3, 2003.

[58] W.R. Heinzelman, J. Kulik, and H. Balakrishnan. Adaptive protocols for information dissemination in wireless sensor networks. In *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, pages 174–185. ACM New York, NY, USA, 1999.

[59] M. Gen and R. Cheng. *Genetic Algorithms and Engineering Optimization*. Wiley-Interscience, 1999.

[60] DE GOLDBERG. Genetic and evolutionary algorithms come of age. *Communications of the ACM*, 37(3):113–119, 1994.

[61] Genetic algorithms overview. `http:http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/tcw2/report.html`, October 2008.

[62] Genetic algorithms introduction. `http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol1/hmw/article1.html`, October 2008.

[63] D.E. Goldberg and K. Deb. A comparative analysis of selection schemes used in genetic algorithms. *Foundations of Genetic Algorithms*, 1:69–93, 1991.

[64] A. Kenneth. De Jong. An Analysis of the Behavior of a Class of Genetic Adaptive Systems. *Ph. D. Dissertation, University of Michigan, USA*, 1975.

[65] L.B. Booker. Intelligent behavior as an adaptation to the task environment. 1982.

[66] J.E. Baker. Adaptive Selection Methods for Genetic Algorithms. In *Proceedings of the 1st International Conference on Genetic Algorithms table of contents*, pages 101–111. Lawrence Erlbaum Associates, Inc. Mahwah, NJ, USA, 1985.

[67] D. E. Knuth. *The Art of Computer Programming*. Addison-Wesley, 1981.

[68] A. Brindle. *Genetic algorithms for function optimization (Doctoral dissertation and Technical Report TR81-2)*. PhD thesis, University of Alberta, Department of Computer Science., 1981.

[69] D.E. Goldberg. Genetic Algorithms in Search, Optimization, and Machine Learning. Addision-Wesley. *Reading, MA*, pages 170–174, 1989.

[70] B.L. Miller and D.E. Goldberg. Genetic Algorithms, Tournament Selection, and the Effects of Noise. *Urbana*, 51:61801.

[71] Crossover in genetic algorithms. `http://en.wikipedia.org/wiki/Crossover_(genetic_algorithm)`, October 2008.

[72] S.Y. Ho and J.H. Chen. A GA-based systematic reasoning approach for solving travelingsalesman problems using an orthogonal array crossover. In *High Performance Computing in the Asia-Pacific Region, 2000. Proceedings. The Fourth International Conference/Exhibition on*, volume 2, 2000.

[73] S. Sen, S. Narasimhan, and K. Deb. Sensor network design of linear processes using genetic algorithms. *Computers and Chemical Engineering*, 22(3):385–390, 1998.

[74] D. Turgut, B. Turgut, R. Elmasri, and TV Le. Optimizing clustering algorithm in mobile ad hoc networks using simulated annealing. *Wireless Communications and Networking, 2003. WCNC 2003. 2003 IEEE*, 3, 2003.

[75] S.A. Aldosari and J.M.F. Moura. Fusion in sensor networks with communication constraints. In *Proceedings of the third international symposium on Information processing in sensor networks*, pages 108–115. ACM New York, NY, USA, 2004.

[76] S. Jin, M. Zhou, and A.S. Wu. Sensor network optimization using a genetic algorithm. In *Proceedings of the 7th World Multiconference on Systemics, Cybernetics and Informatics*. Citeseer, 2003.

[77] Qinru Qiu, Qing Wu, Daniel Burns, and Douglas Holzhauer. Distributed genetic algorithm for energy-efficient resource management in sensor networks. In *GECCO '06: Proceedings of the 8th annual conference on Genetic and evolutionary computation*, pages 1425–1426. ACM, New York, NY, USA, 2006. ISBN 1-59593-186-4.

[78] K.P. Ferentinos and T.A. Tsiligiridis. Adaptive design optimization of wireless sensor networks using genetic algorithms. *Computer Networks*, 51(4): 1031–1051, 2007.

[79] D.B. Jourdan and O.L. de Weck. Multi-objective genetic algorithm for the automated planning of a wireless sensor network to monitor a critical facility. In *Proc. of SPIE Defense and Security Symposium*, 2004.

[80] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson. Wireless sensor networks for habitat monitoring. In *Proceedings of the 1st ACM*

international workshop on Wireless sensor networks and applications, pages 88–97. ACM New York, NY, USA, 2002.

[81] A. Woo and D.E. Culler. *Evaluation of Efficient Link Reliability Estimators for Low-power Wireless Networks*. Computer Science Division, University of California, 2003.

[82] J.L. Hill. *System Architecture for Wireless Sensor Networks*. PhD thesis, University of California, 2003.

[83] K. Whitehouse. The Design of Calamari: an Ad-hoc Localization System for Sensor Networks. *University of California at Berkeley*, 2002.

[84] H. Karl and A. Willig. *Protocols and Architectures for Wireless Sensor Networks*. Wiley, 2005.

[85] AY Wang and CG Sodini. A simple energy model for wireless microsensor transceivers. In *Global Telecommunications Conference, 2004. GLOBECOM'04. IEEE*, volume 5.

[86] K. Baoqiang, C. Li, Z. Hongsong, and X. Yongjun. Accurate energy model for WSN node and its optimal design. *Journal of Systems Engineering and Electronics*, 19(3):427–433, 2008.

[87] J. Ammer and J. Rabaey. The Energy-per-Useful-Bit Metric for Evaluating and Optimizing Sensor Network Physical Layers. *log*, 20(10):10.

[88] Elias Ekonomou and Kate Booth. Simulating Sensor Networks. In *the 6th Annual PostGraduate Symposium on the convergence of Telecommunications, Networks, and Broadcasting*. Liverpool John Moores University, 2005.

[89] S. Bajaj, L. Breslau, D. Estrin, K. Fall, S. Floyd, P. Haldar, M. Handley, A. Helmy, J. Heidemann, P. Huang, et al. Improving simulation for network research. 1999.

[90] The network simulator - ns-2. `http://www.isi.edu/nsnam/ns/`, October 2008.

[91] I.T. Downard and NAVAL RESEARCH LAB WASHINGTON DC. Simulating Sensor Networks in NS-2, 2004.

[92] D. Gay, P. Levis, R. von Behren, M. Welsh, E. Brewer, and D. Culler. The nesC language: A holistic approach to networked embedded systems. *ACM SIGPLAN Notices*, 38(5):1–11, 2003.

[93] P. Levis and N. Lee. TOSSIM: A Simulator for TinyOS Networks. *Retrieved Jan*, 5:2007, 2003.

[94] E. Altman and T. Jimenez. NS Simulator Course for Beginners. *Univ. De Los Andes Merida, Venezuela, February*, 11, 2003.

[95] K. Fall and K. Varadhan. The ns Manual (formerly ns Notes and Documentation). *The VINT Project*, 1, 2007.

[96] T. Issariyakul and E. Hossain. Developing New Modules for NS2.

[97] F.J. Ros and P.M. Ruiz. Implementing a New Manet Unicast Routing Protocol in NS2. *Dept. of Information and Communications Engineering University of Murcia, December*, 2004.

[98] D.E. Knuth. The art of computer programming, vol. 2. *Reading*, 1981.

[99] K. Sohraby, D. Minoli, and T. Znati. *Wireless sensor networks: technology, protocols, and applications*. Wiley-Interscience, 2007.

[100] P. Downey and R. Cardell-Oliver. Evaluating the impact of limited resource on the performance of flooding in wireless sensor networks. In *Dependable Systems and Networks, 2004 International Conference on*, pages 785–794.

[101] O.B. Akan and I.F. Akyildiz. Event-to-sink reliable transport in wireless sensor networks. *IEEE/ACM Transactions on Networking (TON)*, 13(5): 1003–1016, 2005.

[102] D.B. Johnson and D.A. Maltz. Dynamic source routing in ad hoc wireless networks. *Klumwer International Series in Engineering and Computer Science*, pages 153–179, 1996.