TDMA BASED WIRELESS SENSOR NETWORK FOR MILITARY MONITORING (MILMON)

by

İlker BEKMEZCİ

B.S., Computer Engineering, Boğaziçi University, 1994
M.S., Computer Engineering, Boğaziçi University, 1998

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy

Graduate Program in Computer Engineering
Boğaziçi University
2008

TDMA BASED WIRELESS SENSOR NETWORK FOR MILITARY MONITORING (MILMON)

APPROVED BY:

Assoc. Prof. Fatih Alagöz (Thesis Supervisor)	……………………
Prof. Emin Anarım	……………………
Prof. M.Ufuk Çağlayan	……………………
Assoc. Prof. Hacı Ali Mantar	……………………
Assist. Prof. Tuna Tuğcu	……………………

DATE OF APPROVAL: 19.09.2008

ACKNOWLEDGEMENTS

The people most responsible for my success as a graduate student are my advisor, Associate Professor Dr. Fatih ALAGÖZ and my wife Mukaddes BEKMEZCİ. First of all, many thanks to Associate Professor Dr. ALAGÖZ for his academic support which made this research possible. His enthusiasm, experience and collaborative spirit helped me to study for this long period. My wife’s support always encouraged me through the inevitable times when one’s graduate career looks in doubt. I would also like to thank Professor Tuna TUĞCU for providing help any time I needed.

I also thank to Prof. Emin ANARIM, Prof. M.Ufuk ÇAĞLAYAN, Assoc. Prof. Hacı Ali MANTAR, and Assist. Prof. Tuna TUĞCU for their valuable contributions and revisions.

Thanks also to my commander Major Ünal ARI who always support me to continue on my PhD work. I also have to remember the contributions of İlknur BEKMEZCI, my sister, Major Gürsev PİRGE, and Captain Özgür Koray Şahingöz who helped me in especially on writings.

At the end, I thank to my one year old daughter, Gülçin, for giving me the strength to live.

ABSTRACT

TDMA BASED WIRELESS SENSOR NETWORK FOR MILITARY MONITORING (MILMON)

Wireless sensor network (WSN) is a new network family that enables to create smart environments. Although WSN has many application areas, military applications of WSN are very interesting. In this thesis, a new TDMA based sensor network for military monitoring (MILMON) is proposed. MILMON is developed to operate in large areas for acceptable lifetime periods. Design considerations of MILMON are energy consumption, delay, and fault tolerance. The main problems of TDMA based systems are time synchronization and time slot distribution. In order to realize MILMON, a new time synchronization mechanism (SyncHRT), a new distributed time scheduling mechanism (ft_DTSM) and data indicator slot mechanism (DISM) are proposed. Time synchronization with high range transmitter, SyncHRT is designed to minimize energy consumption and maximize precision. It assumes the existence of high range transmitter, so that many of the nodes can receive the broadcast signal of the high range transmitter. In this way, sensor nodes can be synchronized to a central point. Simulation model shows that SyncHRT can reduce energy consumption and increase precision. Another mechanism proposed for MILMON is delay sensitive, energy efficient and fault tolerant distributed slot assignment algorithm (ft_DTSM). It uses much less energy than the existing slot assignment mechanisms and it reduces delay with the convergecast traffic assumption which is very common traffic type for WSNs. Its fault tolerant structure helps to survive against single point of sensor node failures. Another mechanism proposed for MILMON is data indicator slot mechanism (DISM) which reduces energy consumption especially on low traffic requirements, as in most of the military monitoring systems.

Analysis and simulation show that although there are WSN systems that perform better than our system for only energy consumption or for only delay, MILMON realizes an optimization on energy, delay and fault tolerance.

ÖZET

ASKERİ GÖZETLEME AMAÇLI ZAMAN BÖLÜMLÜ ÇOKLU ERİŞİM TABANLI KABLOSUZ ALGILAYICI AĞ SİSTEMİ

Kablosuz algılayıcı ağlar (KAA) akıllı çevreler yaratılmasını sağlayan yeni bir ağ ailesidir. KAA'ların birçok kullanım alanı olmasına rağmen, askeri amaçlı olanları en ilginçleridir. Bu tezde, askeri gözetleme amaçlı zaman bölümlü çoklu erişim tabanlı yeni bir KAA (MILMON) önerilmiştir. MILMON geniş alanlarda makul bir sure dahilinde çalışmak üzere geliştirilmiştir. MILMON'un tasarım hedefleri enerji harcamasını azaltmak, gecikmeyi azaltmak, hata toleransını artırmaktır. Zaman bölümlü çoklu erişim tabanlı sistemlerin ana problemleri zaman senkronizasyonu ve zaman bölme dağıtımıdır. MILMON'u gerçekleyebilmek için, yeni bir zaman senkronizasyon mekanizması (SyncHRT), yeni bir zaman bölme dağıtım mekanizması (ft_DTSM) ve veri gösterge bölmesi mekanizması (DISM) önerilmiştir.

Çıkış düğümünde bulunan uzun menzilli verici ile zaman senkronizasyonu, SyncHRT, enerji harcamasını en aza indirmek ve kesinliği artırabilmek amacı ile tasarlanmıştır. Uzun menzilli vericinin yolladığı sinyalleri birçok algılayıcı düğümün alabildiği kabul edilerek bu mekanizma geliştirilmiştir. Böylece çıkış düğümünün sinyalleri ile düğümler kendilerini doğrudan çıkış düğümüne senkronize edebilmektedir. Simülasyon modelleri SyncHRT'nin enerji harcamasını azaltırken kesinliği de arttırdığını göstermiştir. MILMON için önerilen bir başka mekanizma ise gecikmeye hassas, enerji verimli ve hata toleranslı dağıtık zaman bölümü dağıtım algoritmasıdır (ft_DTSM). Halen mevcut olan zaman bölümü dağıtma algoritmalarından çok daha az enerji harcar ve KAA'larda çok sık görülen yaklaşımlı veri trafiği esas alarak gecikmeyi de azaltır. ft_DTSM'in hata toleranslı yapısı, tek nokta hatalarına karşın ağın çalışabilmesine yardımcı olur. Askeri gözetleme sistemlerinin çoğunda olduğu gibi düşük veri yüklerine sahip sistemlerde, enerji harcamasını azaltan bir diğer mekanizma olarak veri gösterge bölmesi mekanizması (DISM) geliştirilmiştir.

Analiz ve simülasyon sonuçları, sadece gecikme yada sadece enerji harcaması için MILMON'dan daha iyi performans gösteren KAA'lar olmasına karşın, MILMON'un enerji, gecikme ve hata toleransı konularında bir eniyileme gerçekleştirebildiğini göstermektedir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS / ABBREVIATIONS

a_i	Clock drift
B	Bit rate
b_i	Clock offset
d	Distance between communicating sensor nodes.
e_{amp}	Energy consumption of amplifier
E_{elect}	Energy consumption of radio circuit
$E_r(l,d)$	Energy consumption of receiving l bit from the distance d
$E_t(l,d)$	Energy consumption of transmitting l bit to the distance d
ftp	Fault tolerance parameter
h	Hop number
k	The expected number of neighbors
l	Number of received bits.
L	Average number of relaying packets in one second
m	The number of broadcast signal used for linear regression
n	The number of nodes in a cluster
N_b	The number of the nodes that has no valid slot and can not contend due to a success within one hop
N_c	The number of contender nodes in one hop neighborhood

N_c	Number of children for a sensor node
N_{maxh}	Maximum hop number
N_p	Number of alternative parent for a sensor node
N_T/N_R	The number of times transmitter/receiver is switched on per unit time
P_c	Power consumption of radio circuit
P_{DMAC}	Power consumption of DMAC
P_{FLAMA}	Power consumption of FLAMA
P_{ns}	Power consumption of the new system (MILMON)
P_{out}	The output power of the transmitter
$P_{T/R}$	Power consumed by the transmitter/ receiver
R_1	The first parameter of ft_DTSM
R_2	The second parameter of ft_DTSM
R_{DTSM}	Running time of DTSM
R_{FPRP}	Running time of FPRP
R_{ft_DTSM}	Running time of ft_DTSM
R_r	Receive energy for one bit
R_s	Transmit energy for one bit
s	Total number of slots in one time frame
S_d	Number of bits in one data package

S_{dism}	Number of bits in one DISM signal for sending
S_{pr}	Preamble signal
t	Time
T_{as}	Time for advertisement slots
$t_i(t)$	Clock of sensor node i at time t
T_{on}/R_{on}	The transmitter/receiver on time
T_{one_as}	Time for one advertisement slot
T_{rc}	Total number of reservation cycles in a reservation frame
T_{se}	Time for one signal exchange slot
T_{st}/R_{st}	The transmitter/receiver start-up time
T_w	Data time frame
u	The number of sub time frame
ACK	Acknowledgement
AS	Advertisement slot
C4ISRT	Command, Control, Communications, Computing, Intelligence, Surveillance, Reconnaissance and Targeting
CSMA	Carrier Sense Multiple Access
CSMA/CD	Carrier Sense Multiple Access with Collision Detection
CTS	Clear to Send
DARPA	Defense advanced research project agency

DIS	Data indicator slot
DISM	Data indicator slot mechanism
DSN	Distributed sensor networks
ft_DTSM	Fault tolerant distributed time slot assignment mechanism
GPS	Global positioning systems
LSM	Local site masters
MAC	Medium access control
MEMS	Micro-electro-mechanical systems
MILMON	Newly Proposed TDMA Based Wireless Sensor Network for Military Monitoring
MTE	Minimum transmission energy routing
QoS	Quality of Service
RTS	Request to Send
SensIT	Sensor information technology
SNR	Signal to Noise Ratio
SyncHRT	Time Synchronization Mechanism with High Range Transmitter
TDMA	Time division multiple access
WSAN	Wireless sensor and actor network
WSN	Wireless sensor network
WMSN	Wireless multimedia sensor network

1. INTRODUCTION

1.1. Wireless Sensor Networks (WSN)

In recent years, with the pace of the developing micro-electro-mechanical systems (MEMS) technology, it has been possible to integrate battery operated sensor, computing power and low power wireless communication components into one small size device. Typical sensor node architecture is presented in Figure 1.1 [1].

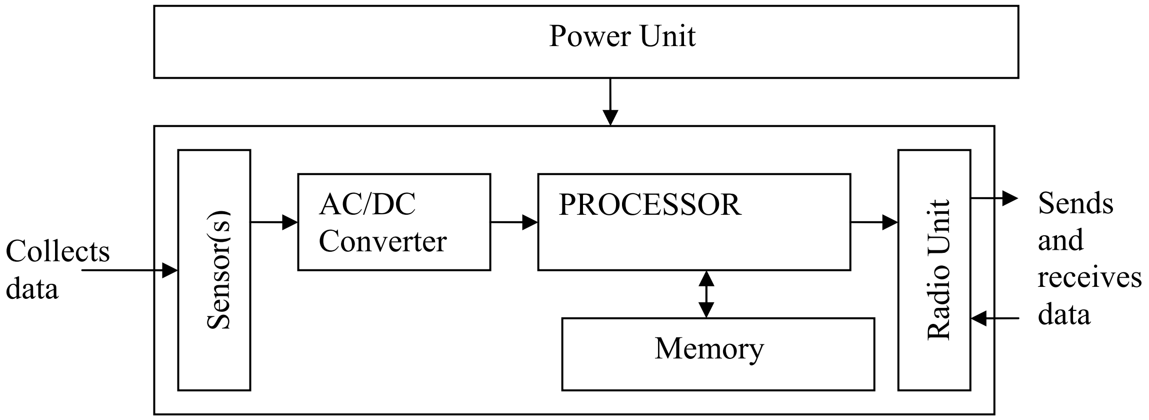


Figure 1.1. Typical sensor node architecture of a sensor node

A sensor node collects data from the environment continuously. It converts the data by the help of its converter. Processor processes the collected data. If node wants to store the data, it uses its very limited memory. If node wants to send data to another node, it sends from radio unit. It can receive the data from other nodes from its radio unit.

The data collected by only one node are nearly useless, but the collaborative work of thousands of these nodes can be used to collect process and send the data about the environment. In other words, smart environments can be designed. The network that is composed of these wireless sensors is called as wireless sensor network (WSN). Wireless sensor networks are also known as distributed sensor networks [2].

This kind of network has many advantages over traditional sensors which are deployed in the following two ways [3]:

- Sensors can be positioned far from the actual phenomenon. These kinds of systems are called as remote sensors. Remote sensing is the science and art of obtaining information about an object, area, or phenomenon through the analysis of data acquired by a device that is not contact with the object under investigation [4]. In most of the systems, the object and the sensor very distant. Radar or satellite photogrametry are very successful examples of remote sensing systems. However, the distance between the sensor and the sensed entity is a serious disadvantage. Obviously, it increases the signal to noise ratio (SNR) and the perception of traditional sensors may not be clear enough to get healthy data. For example, up to now atmospheric movements and data are gathered from the satellite. The distance between atmosphere and satellite distorts the actual data, so it is always possible to get noisy data. On the other hand, if WSN is used for this purpose, sensors will be in the environment and can get almost exact data. By the help of their wireless communication component, WSN can send the sensed data to the desired centers.

- Several sensors that perform only sensing can be deployed. Because of the lack of communication capability of traditional sensors, they can’t send the collected data. This can be an important problem. In WSN case, sensors can sense and send data at the same time, in the same device. While sensors sense the environment, they can send the data to the central nodes through the network.

- Small sizes of wireless sensor nodes easies the deployment. Small size is also important for hiding from physical attacks. Size of a sensor node is so small that sometimes it is called as smart dust or motes [5]. Figure 1.2 can give an idea about the size of the nodes. In recent years, sensor nodes sizes are getting smaller and smaller. The node proposed in [6] is 31x17 mm^2.

A classic application scenario can be very helpful to understand the WSN concept. Assume that military units clearing urban terrain must clear a building but cannot afford to leave people behind to ensure that it stays cleared. Military unit must be notified, if anyone enters the cleared portion of the building after they have left. Sensor networks can be very helpful in this situation. Soldiers would carry something like a dispenser, possibly attached to their weapons, filled with sensor nodes that could be shot or emplaced by hand on a wall, stairway or doorway. The sensors, using some combination of acoustic, IR, visual or vibration cues would pass information about intruders to the appropriate people [7].

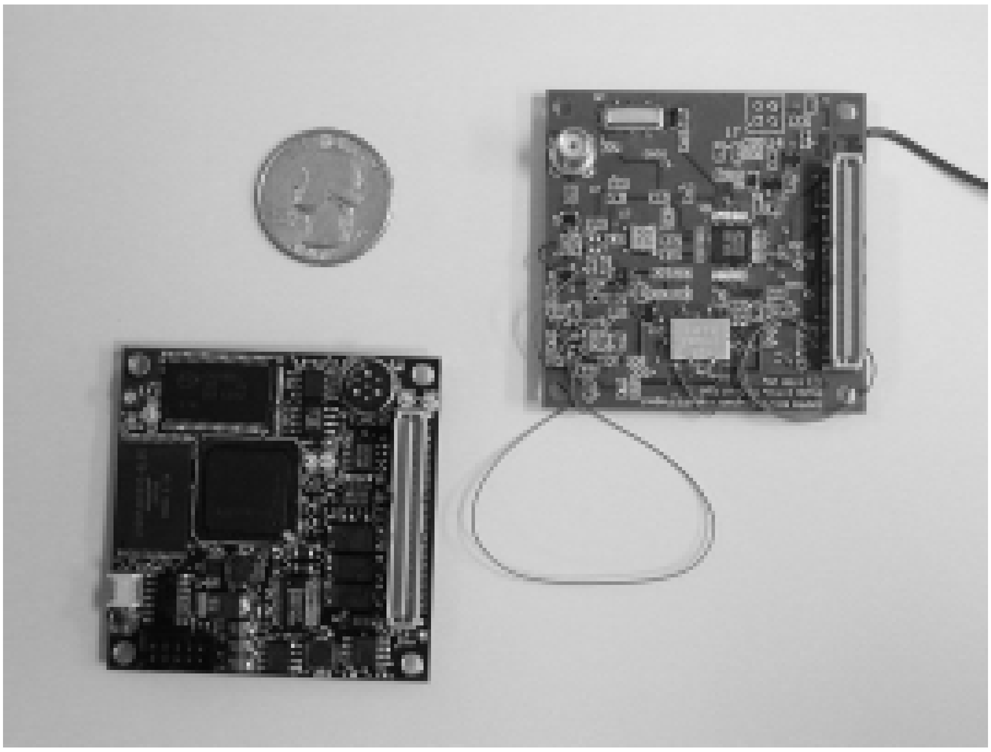


Figure 1.2. Implementation of wireless sensors compared to a US quarter [8]

In this thesis, a new wireless sensor network for military monitoring (MILMON) is developed. Wireless sensor networks are application specific and all design considerations are different for each application. Especially, military application requirements are very special. MILMON monitors its environment and reports the sensed events as soon as possible to the sink. Design considerations of MILMON are energy consumption, delay and fault tolerance.

1.2. Application Areas of WSN

The potential applications of wireless sensor networks are highly varied. Environmental monitoring, condition based maintenance, smart spaces, military, precision agriculture, transportation, factory instrumentation, inventory tracking are just some of the sample application areas [9]. Nodes of WSN can be configured to carry different types of sensors so that it can monitor different types of events. Different sensors can sense thermal, visual, acoustic or infrared data. So one WSN can be used for fire detection, another can be used to detect targets for building security. However sensor networking applications can be classified into two main branches: Military applications and civilian applications.

1.2.1. Military Applications

After the Gulf War, a new concept has been very popular among military specialists, revolution in military affairs (RMA) [10]. Sophisticated Conventional War (SCW) which is based on the combination and integration of high quality target intelligence and acquisition methods, on effective and rapid command and control, and on high kill probability precision fire-power capable of destroying most targets, on land, at sea or in the air, either by day or at night is a part of RMA [11]. One of the most important parts of SCW is sensor networks. Especially, two important programs the Distributed Sensor Networks (DSN) and the Sensor Information Technology (SensIT) form the Defense Advanced Research Project Agency (DARPA), sensor networks are applied very successfully in the military sensing [12]. Wireless sensor networks can be an integral part of military command, control, communications, computing, intelligence, surveillance, reconnaissance and targeting (C4ISRT). In [13], it is shown that scalable, affordable, real-time wireless network can contribute to the common military operation picture.

One of the most important military application families for WSN is intruder detection. The intrusion detection is defined as a mechanism for a WSN to detect the existence of inappropriate, incorrect, or anomalous moving attackers [14]. The main aim of WSN based intruder detection systems is to identify the intruder and to report the intruder to a base station. Basically, this thesis also proposes a new WSN based military intrusion detection system.

REMBASS, the Remotely Monitored Battlefield Sensor System, Integrated Sniper Location System and Environmental Systems Management, Analysis and Reporting Network (E-SMART) are important military application examples.

REMBASS, the Remotely Monitored Battlefield Sensor System, is an unattended ground sensor (UGS) system that detects, classifies, and determines direction of movement of personnel, wheeled vehicles, and tracked vehicles. It provides world wide deployable, day/night, all-weather, early warning surveillance and target classification. The sensors are placed along likely avenues of approach or intrusion and respond to seismic and acoustic disturbances, infrared energy, and magnetic field changes. The most serious problem of REMBASS is the false alarms. So each sensor runs a complex signal processing algorithm which aims to result in a high probability of detection. The sensor information is incorporated into short, digital messages and communicated by VHF radio burst transmission [15].

Gunfire has a very particular acoustic signature that can provide the location of the gun when processed by an acoustic sensor designed to exploit its particular characteristics. DARPA funded a program to develop such a sensor during the years of 1995 to 1997. Time and direction of arrival of the muzzle blast signature. This signature will unambiguously point in the direction of the shooter. The sensors in the system acquire various data from the environment and process very complicated algorithm to find out the range and the angle of the sniper. Typically, the system could locate the sniper’s location within one to two degrees in bearing and within 20 per cent in range. When it locates the sniper, it sends the location data to the center [16].

Since 1991 the U.S. Air Force Research Laboratory, Air Expeditionary Forces (AEF) Technologies Division (AFRL/MLQ) has been developing the Environmental Systems Management, Analysis and Reporting Network (E-SMART) which is deployed at Tinker AFB in Oklahoma City. The E-SMART system has served as a network to connect multiple sensors and other input and output devices deployed over a wide area to provide comprehensive, accurate, and near real-time information on environmental conditions.

A networked system of chemical and biological detectors can provide adequate warning of chemical and biological attack that can save many lives by rapidly evacuating affected areas and providing timely, effective medical treatment to exposed personnel. That is the goal of the E-SMART system, which will consist of a network of sensors and communication links [17].

1.2.2. Civilian Applications

Most of the civilian applications are about detection and tracking. Sensors detect the desired event and send the data to sinks or sensors track the movements of the desired objects.

Habitat monitoring is a driver application for wireless sensor network. In August 2002, researchers from UCB/Intel Research Laboratory deployed a mote-based tiered sensor network on Great Duck Island, Maine, to monitor the behavior of storm petrel. Wireless sensor nodes were placed at area of interest (e.g., inside a burrows). Those nodes, grouped into sensor patches, transmit sensor reading to a gateway, which is responsible for forwarding the data from the sensor patch to a remote base station through a local transmit network [18].

One of the most known application areas of WSN is fire detection. The most important factor to handle fire is time, because fire can spread very quickly. So, fire station must get the fire data precisely and quickly. Sensor nodes can be deployed randomly to detect fire in a forest. The number of nodes can be thousands to millions. They can be equipped with effective power scavenging methods, such as solar cells [19]. In this way, nodes can be functional for years. When nodes sense a noticeable change in the level of temperature, they send the data to sinks.

Another detection application is flood detection [20]. ALERT is an example for flood detection. Several types of sensors are used in ALERT system. These sensors supply information about water level, weather condition and rainfalls. A typical ALERT installation consists of several types of sensors in the field: rainfall sensors, water level sensors, weather sensors, etc. A predefined set of data is regularly extracted from each sensor, transferred to a central site and stored in a database system. Users can also query the environment through a graphical user interface. Flood detection systems are useful for search and rescue teams especially.

Wireless sensor networks also open new avenues for the implementation of wireless networks for telemedicine. The envisioned concept provides the individual patient with greatly improved mobility and allows him/her to roam freely outside of treatment centers, thus facilitating a higher quality of life. This capability (independence of wired monitoring/diagnosis equipment) is achieved by the patient carrying a sensor network that communicates with a central/supervisory processor which would typically be located at a treatment center [21]. In [22], a complete design of a patient monitoring solution for intensive care unit environments has been presented. At the bedside end, the system is a plug-and-play sensor network communicating with a gateway that collects medical information and sends the data to a monitoring server.

Active volcanoes are another application area for wireless sensor networks. Fang and Kedar have proposed a system design for monitoring active volcanoes [23]. They combine sensor network system engineering with systems-on-chip implementation to develop an integrated surveillance system called Sensor Networks for Active Volcanoes (SNAV). SNAV can identify final stages of eruptive process; assess the hazard by measuring the volume of magma. When SNAV detects an event it can send the data to the base station which is far from the dangerous area.

Efficient water management is a major concern in many cropping systems in semiarid and arid areas. WSN-based irrigation systems offer a potential solution to support site-specific irrigation management that allows producers to maximize their productivity while saving water. In [24], a new variable rate irrigation control system is proposed based on a wireless sensor network.

WSN may be deployed over a region to record data for meteorological, geophysical or planetary research. It may be employed to perform measurements in environments where wired sensors are unusable or lead to measurement errors. Examples include instrumentation of semiconductor processing chambers, rotating machinery, wind tunnels, and anechoic chambers. In biological research, WSN may be used to monitor the movements and internal processes of insects or other small animals [25].

Industrial control is a newly developed application area for wireless sensor networks. In a traditional process control system, devices are connected to controllers via some types of buses. A sensor collects certain status information of a process and feeds it to a controller. Based on the readings from sensors, the controller determines whether the actuators should act to maintain some physical property of the process, e.g., the flow in a pipe must be kept at a fixed speed. The main problem for WSN based industrial control is real time control over wireless network systems. The latest studies have shown that a near real-time system can be set up with the WSNs and WSNs can be used for industrial control securely [26].

1.3. Design Issues in WSN

Wireless sensor networks bring unique solutions as well as unique design challenges. Most of the realization of sensor network applications requires ad hoc networking. Ad hoc networking design issues have been investigated and discussed for many years. However, no ad hoc network protocol can solve all design problems of wireless sensor networks. The basic differences between traditional ad hoc networks and wireless sensor networks can be outlined as follows:

- Sensor nodes are prone to failures.
- Event arrival rates or environmental conditions may change very frequently. Thus, sensor network should adapt itself instantly.
- Most ad hoc network protocols assume that the traffic pattern of each node is independent. However, the data traffic in sensor networks is highly synchronized and correlated.
- Unlike conventional wireless networks, a WNS network has to support large numbers of sensors in a local area with short range and low average bit-rate communication. In WSN, node density can be up to 20 nodes/m^3 [8], and typical bit rate is about 1–100 Kbps [27].
- In WSN, sensor to sensor communication is unimportant. The main aim is to collect the data into sinks. So, the most important objective is to produce sensor to sink communication. Nodes sense the environment and produce data. It forwards the data through other nodes to a sink node. Sink nodes are connected to task management

node. User of sensor network can assign the tasks or can get the data from task manager node. Most of the sensor network produces convergecast traffic. This structure is more or less the same in all WSNs.

- Another important feature is their unattended nature. Because of their compact form factor and potential low cost, nodes might be autonomously deployed in an unplanned fashion.
- Sensor nodes in different WSNs can be very different. Behaviors of one specific sensor node can be totally different from the others even under the same conditions. Energy consumption models can be a good example of this situation. In this case, it is clear that protocols which are developed for a WSN may not be optimal for another WSN [8].
- The application areas for WSN are also very different. One application may require very low bandwidth and long network lifetime, and the other may require quick response and high quality. Sensor network systems should be application specific [28]. It is not only valid for communication protocols and algorithms of sensor networks. Even the sensor node hardware should be developed in application specific manner [29].
- Although in most of the mobile and ad hoc networks, the main objective is to provision of high QoS [30-32], it is mainly energy efficiency in WSN. Many nodes in the emerging sensor systems will be unethered, having only finite energy reserves from a battery. When battery goes off, node will simply die. The requirement for energy efficiency pervades all aspects of the system design. In the next section, general power consumption models of sensor networks will be outlined.
- Computational powers and memory resources are also very limited in sensor nodes.

These new challenges bring new performance issues and metrics. Here is some of the important performance metrics for sensor networks:

- Overall network lifetime,
- Power consumption,
- Delay,
- Scalability,
- Fault tolerance,

1.3.1. Network Lifetime and Power Consumption

Many researches have concentrated on maximizing the lifetime of wireless sensor networks [33]. In the literature, many different lifetime definitions are used, such as duration of time until the first sensor failure due to battery depletion in [34], fraction of surviving nodes in a network in [35] and [36], and mean expiration time in [37]. In [38], the network is considered functional if it can produce an estimation satisfying a given distortion requirement; otherwise, it is nonfunctional. Whatever the definition of network lifetime is, it is closely related with the lifetime of sensor nodes. Sensor node has limited power. When the power of a sensor node goes off, the sensor node also goes off. Energy consumption is the most important limitation for the lifetime of sensor nodes.

In this section, the most important aspect of WSN, power consumption is investigated in detail. After the investigation of power consumption models, taxonomy about energy efficient strategies in WSN designs is presented.

The scarcest resource of a sensor node is energy, so the most important design aspect of sensor networks is energy saving. All protocols and algorithms for sensor networks should be designed to avoid energy waste. In order to avoid energy waste, power consumption model of sensor nodes must be stated clearly.

The main task of a sensor node in a sensor is to detect events, perform quick local data processing, and then transmit the data. Power consumption can hence be divided into three domains: sensing, communication, and data processing [3]. Sensing task is a continuous task and energy consumption of sensing is related with the hardware of sensor node. In this proposal, network architectures of sensor networks will be investigated. In this case, it is better to get into details of communicational and computational energy consumption rather than sensing.

Dominant factor in energy consumption for sensor nodes is communication. Not only transmission but also receiving is the source of energy waste. In [8], an energy model of radio circuit is formulated in Equation (1.1).

$$P_c = N_T*[P_T*(T_{on} + T_{st}) + P_{out}*T_{on}] + N_R*[P_r*(R_{on} + R_{st})] \tag{1.1}$$

where *PT/R* is the power consumed by the transmitter/ receiver; P_{out}, the output power of the transmitter; T_{on}/R_{on} the transmitter/receiver on time; T_{st}/R_{st}, the transmitter/receiver start-up time and N_T/N_R, the number of times transmitter/receiver is switched on per unit time, which depends on the task and medium access control (MAC) scheme used. Figure 1.3 shows the block diagram of radio circuit which is used in Equation (1.1).

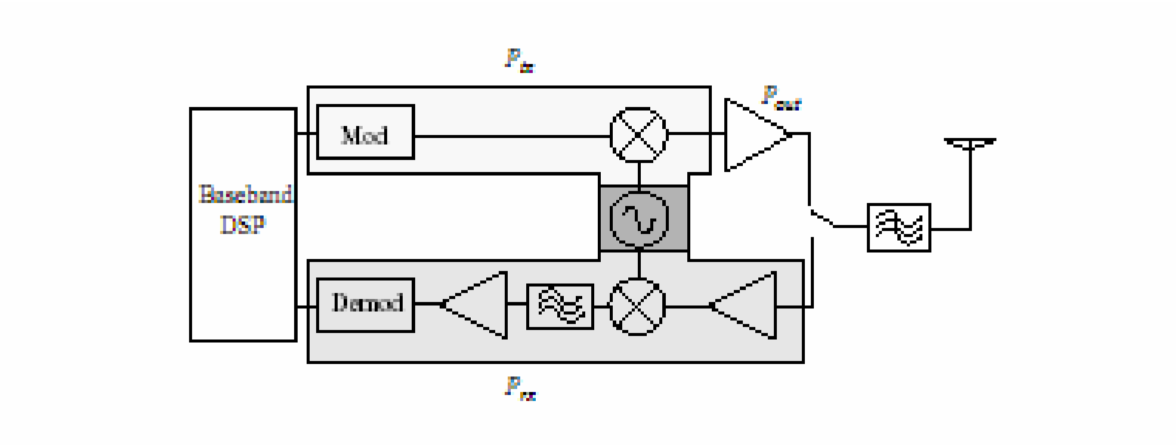


Figure 1.3. A block diagram of radio circuit.

Another model is proposed by Heinzelman in [39]. According to this model, transmitter dissipates energy to run radio electronics and amplifier and receiver dissipates energy to run radio electronics [40]. Power attenuation is dependent on the distance between transmitter and receiver. In general, the minimum output power required to transmit a signal over a distance *d* is proportional to d^n, where n is between 2 and 4 [41]. For relatively short distance, n is near to 2 and for longer distances n is equal to 4. Thus to transmit l bit message to a distance d, the radio expends:

$$E_t(l,d) = l*E_{elect} + l*e_{\mathrm{amp}}*d^2 \text{ for } d<86.2 \text{ m.} \tag{1.2}$$

$$E_t(l,d) = l*E_{elect} + l*e_{amp}*d^2 \text{ for } d>86.2 \text{ m.} \tag{1.3}$$

and to receive *l* bit message to a distance *d*, the radio expends:

$$E_r(l,d) = l* E_{elect} \tag{1.4}$$

The block diagram of this power model is shown in Figure 1.4.

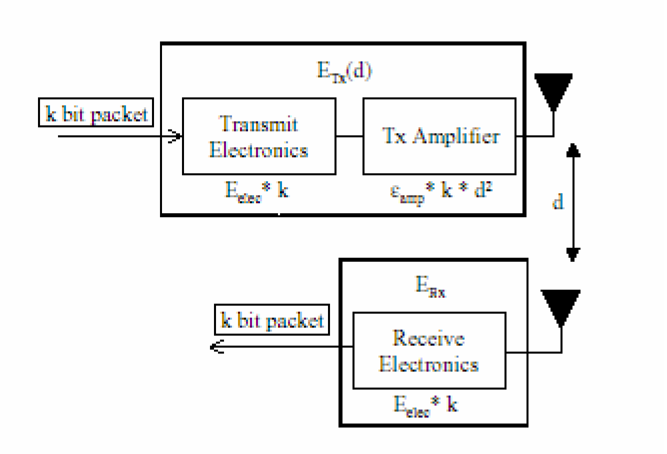


Figure 1.4. A power model for sensor nodes.

In [42], a routing strategy analysis is discussed according to this power model. Direct transmission and minimum transmission energy routing is compared with respect to their energy consumption in sensor networks. Minimum transmission energy routing (MTE) selects the route that uses the least amount of energy to transport a packet from source to destination. [43]. Direct transmission is to transmit directly from source to destination. As conclusion of this study, it can be said that when transmission energy is on the same order as receive energy, which occurs when transmission distance is short and/or the radio electronics energy is high, direct transmission is more energy-efficient on a global scale than MTE routing. Another study about the relationship between routing strategies and energy consumption is about the tradeoff between multi-hop strategy and direct transmission [44]. This study is stated that MTE is not the optimal strategy for sensor nodes. If a sensor node is around 30m diameter, direct transmission may be the best strategy. Multi-hop strategy may be used if the diameter of network is larger then 30m.

Another energy consumption source of sensor nodes is data processing. In [8], Pottie has stated that the energy cost of transmitting 1 KB a distance of 100 m is approximately the same as that for executing 3 million instructions by a 100 million instructions per second. It concludes that sensor nodes should process the data first and then they should transmit the data. It also shows the importance of data aggregation and beam forming algorithms for sensor networks.

Energy constrained nature of sensor nodes requires the use of energy efficient strategies to maximize network lifetime. There are many studies about energy consumption efficiency of wireless sensor networks. There are four main classes for energy efficient strategies [45]. These are energy efficient routing, scheduling the node's sleeping state, topology control by tuning node transmission power and reducing the volume of information transferred by the nodes.

The goal of energy efficient routing algorithms is to minimize the energy consumed by the end-to-end transmission of a packet, to avoid nodes with a low residual energy and to reduce the number of unsuccessful transmissions.

Scheduling the node's sleeping state is another technique to reduce energy consumption. It allows nodes to sleep in order to spare energy, provided that the network and application functionalities are still ensured. In this thesis, a new time slot distribution mechanism is proposed to schedule the node's sleeping states.

Topology control by tuning node transmission power is an important strategy to maximize energy efficiency. These strategies find the optimum node transmission power that minimizes energy consumption, while keeping network connectivity.

Energy consumption can be reduced by decreasing the volume of information transferred. The information volume can be reduced by aggregating information, decreasing the frequency of information refreshment with distance, avoiding information transfer to uninterested nodes.

1.3.2. Delay

While traditional network systems focus on quality of service (QoS) issues, wireless sensor networks primarily concentrate on energy efficiency. They must have inbuilt tradeoff mechanisms that give the end user the option of prolonging network lifetime at the cost of lower throughput or higher transmission delay [3].

Energy efficiency and delay are always closely related with each other. When an energy efficient system is designed, most of the time, delay is increased. Many wireless sensor network designs try to optimize energy efficiency and delay performances.

In many WSN systems, delay is not the most important performance metric. For a habitat monitoring, even tens of seconds delay is acceptable. Another example for delay insensitive wireless sensor network is the wireless vineyard project [46]. In this project, information gathered by sensor networks can be used for irrigation or harvesting to improve quality.

There are some applications in which delay can be important. Especially, military applications can be more delay sensitive than civilian applications. In MILMON scenario, the intruder should be reported as soon as possible. If the report of an incoming intruder is too late, it can be a disadvantage for the user of MILMON.

Although delay is more important in military applications, a couple of seconds is also acceptable in most of the scenarios. Delay requirement of a typical WSN is much more relaxed than a traditional network. In this case, delay and energy efficiency optimality problem is still a concern for military WSNs, like MILMON.

1.3.3. Fault Tolerance

Sensor nodes are prone to failures. They may have physical damage or they may be affected by the environmental conditions. Power drainage is another cause for sensor node failures. The failure of sensor node should not affect the functionality of the whole network. This topic is called as fault tolerance in WSN. Fault tolerance is the ability to sustain sensor network functionalities without any interruption due to sensor node failures [47].

The fault tolerance requirement level of WSN is addressed by the environment, sensor nodes, and the application. If the environment is very harsh and sensor nodes are very sensitive for the environmental changes, higher level of fault tolerance is needed. If environment is steady and sensor nodes can work in the environment most of the time, fault tolerance requirement can be relaxed. Another point that should be considered for fault tolerance requirement is the application of the WSN. If the sensor nodes are deployed in a secure area, and the maintenance of the sensor nodes is available, fault tolerance may be less important. However, if WSN is designed for military applications, sensor nodes are being deployed in a battlefield for surveillance and detection. So, the fault tolerance has to be high because the sensed data are critical and sensor nodes can be destroyed.

In many applications, sensor nodes are deployed randomly. the randomness may cause a node far from others so as to it may be out of the validate transmitting range, or a node's signal may be jammed or interfered by the bad landform where it locates, make the information that emits form it or passes through it could not reach signal target, or may even result in died zone [48]. If the nodes are blindly densified, the flux of signals would increase, which makes the energy in nodes exhaust earlier, and shortens the life-span of the network.

For MILMON, fault tolerance is a very important design issue. MILMON is developed for military monitoring. Physical attacks may take place. The environment may be very harsh. If the fault of a single sensor node may lead any performance degradation, MILMON can not be reliable.

2. CURRENT DESIGNS FOR WSN

Wireless sensor networks are basically ad hoc wireless networks and there are many algorithms for wireless networks. However, because of the differences that are mentioned in Chapter 1, traditional wireless network communication solutions, such as Bluetooth or MANET can not be applied to WSN directly.

The power consumption behavior of a Bluetooth node is very different from a sensor node. Sensor nodes try to save energy by shutting down radio circuits when it is not needed. However, the cost of turn on and off the radio circuit is so high in terms of power that, Bluetooth node can not turn off its radio circuit when it is not needed [49]. Moreover, Bluetooth tries to form a star topology. Master node gives different time slots to each slave node. This structure never changes master node can support only seven slave node. The main aim of Bluetooth is to replace the cables between electronic user terminals with RF links. For example, it may be useful to replace the cardiograph machine’s cables with some tiny and cheap chips. However in harsh environment, to form such a structure for densely and unattended deployed sensor nodes is not easy. MANET solutions can be thought as another alternative for mobile ad hoc structures. However, the primary goal of MANET is to form a network structure for maximum QoS under mobile conditions [3]. In most of the wireless sensor networks, QoS is not the first concern. The highest priority is for the energy consumption. When a battery of a sensor node is exhausted, that node simply dies. There is no opportunity to maintain so many nodes. In addition to this factor, wireless sensor networks are more mobile and denser than the MANET can support.

There are two basic approaches for wireless sensor networks: Random access (CSMA) based and fixed allocation [3]. Each of them has its own advantages and disadvantages.

2.1. Random Access (CSMA) Based WSN

Random Access Protocols address the situation where the traffic is burst. With burst traffic a station can be busy for short amounts of time and idle for the remaining stretches.

There are many different types for traditional networks, like ALOHA, slotted ALOHA or CSMA, CSMA/CD (collision detection). However, sensor networks are different from traditional networks. In this case, different CSMA regimes must be developed for sensor networks. The most known examples for wireless networks are IEEE 802.11 [50] and MACAW [51]. Unfortunately, these are not completely suitable for sensor networks. In this section SMACS will be presented as an example of contention based sensor network.

According to the contention based architecture, when a node tries to transmit data and the medium is idle, it must send RTS (request to send) control frame. If the second node is idle for transmission and receives RTS, it replies the request by a CTS (clear to send) frame. After this handshaking, the first node transmits its data. When the data is transmitted the second node sends an ACK (acknowledgment) frame.

This handshaking aims to solve hidden terminal problem. This protocol ensures that both of the nodes are idle and want to communicate. Although this protocol solves important problems in wireless communication, there should be some revisions to use it in WSN. The following WSN MAC layer is based on some revision on contention based mechanisms.

2.1.1. Sensor MAC (S-MAC)

Sensor-MAC (S-MAC) is a random access based MAC protocol designed explicitly for wireless sensor networks. As reducing energy consumption is the primary goal in S-MAC, it also has good scalability and collision avoidance capability. It achieves good scalability and collision avoidance by utilizing a combined scheduling and contention scheme [52].

The main sources of energy waste in WSN are collision, overhearing, control packet overhead and idle listening. S-MAC aims to reduce energy consumption from all of the sources. Four different new mechanisms are developed to realize the design goal. These are periodic listen and sleep, collision avoidance, overhearing voidance, and message passing. After discussions about the new protocols, the performance of S-MAC will be presented.

In WSN, if nothing is sensed, nodes should be in idle mode for most of the time. However, IEEE 802.11 must listen to the medium for all time to receive possible traffic. This idle listening consumes 50-100 per cent of the energy required for receiving [53]. This algorithm reduces the listen time by letting node go into periodic sleep mode. Each node works for half of the second and at the second half it sleeps. This brings 50 per cent energy saving from idle listening. Choosing and maintaining schedules of the periods is the main problem of this mechanism. Each node maintains a schedule table that stores the schedules of all its known neighbors. There are three main rules to choose the schedule. Before the first step, there is no synchronization and there is no schedule. At the first step, a node listens to receive its neighbor's schedule. If it does not hear any schedule, it chooses a random time to sleep and broadcasts its schedule. If a node hears a schedule from its neighbor, it sets its schedule as the same. If a node hears another schedule after it selects a schedule, it adopts its according to both schedules.

Collision avoidance mechanism is the same as IEEE 802.11 that can address hidden terminal problem. Broadcast packets do not use RTS/CTS. Unicast packets follow the sequence of RTS/CTS/DATA/ACK. Collision mechanism uses not only physical carrier sense but also virtual carrier sense. According to virtual carrier sense protocol, each transmitted packet includes a number that shows how long the remaining transmission will be. In this way, a node that receives a packet knows how long it should be silent. Only at the end of the conversation, it tries to transmit data.

In 802.11 each node listens to all transmissions from its neighbors in order to perform more effective virtual carrier sense. That node hears lots of packets that are not directed to it. A protocol that can solve this problem can prevent lots of energy waste.

The protocol that is used in this research is simple and useful. Every node that hears any RTS or CTS (not directed to it) goes to sleep. The following example will help to understand the protocol.

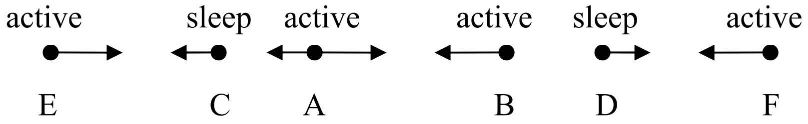


Figure 2.1. Overhearing avoidance protocol example.

Assume that each node can hear the transmissions only from its neighbors, and A is transmitting data to B. The situation is illustrated in Figure 2.1. It is clear that D must sleep until the end of A-B conversation, because D may interfere with B. E and F can not be heard from A or B. So they do not need to sleep. The discussion is about C. If C can communicate with E, it should not sleep. However, all the packets that are coming from E will collide with the transmissions from A. So C can not receive any packet including RTS, CTS. In summary, C can not communicate with any other node, while A is transmitting data. .In this case, C should sleep until the end of the A-B conversation.

There is a classic dilemma on the packet length in communication. If packet length is long, control packet overhead will be lower and cost of retransmission will be higher. On the other hand, if packet length is small, control packet overhead will be higher, and cost of retransmission will be lower.

The approach used in S-MAC is to fragment the long message onto smaller fragments and send them in burst mode. In this approach, only one RTS/CTS packet is used. After RTS/CTS handshaking, when a smaller fragment is transmitted, the sender waits for an ACK packet. At the end of all fragments, medium is assumed as idle. If sender can not get the ACK packet, it resends the fragment. The ACK packets prevent hidden terminal problem. For example there is a node (C) that can hear the receiver node (B) but can not hear the sender (A), and it tries to communicate with another node. If receiver does not send ACK at the end of each fragment, it will test the medium whether it is idle or not. If medium is idle, it will send data immediately. In this way, the communication between A and B will be corrupted. However, when B sends ACK packets frequently, C hears that ACK packets and does not initiate any communication. This mechanism lowers complete transmission time, energy waste that arises from control packets and the loss of resending the long fragment.

The performance of S-MAC is researched with respect to 802.11. Basically, the performance criterion is the average energy consumption. According to [52], on a source node, S-MAC consumes 2-6 times less energy than 802.11 like MAC. Especially in heavy traffic, energy saving rate is increased dramatically. However, S-MAC may not be so successful in delay performances. In some time critical applications, the delay of sending data from source to sink is very important. In these cases, S-MAC should be investigated very carefully.

2.1.2. DMAC

In WSN systems, sensor nodes collect data from the environment and relays to the sink. This concludes that data traffic of most of the WSN systems flows from the nodes to the sink. DMAC [54] leverages the convergecast traffic pattern assumption for low delay and high energy efficiency. DMAC could be summarized as an improved Slotted Aloha algorithm where slots are assigned to the sets of nodes based on a data gathering tree. It arranges the slots of the sensor nodes, so that while energy saving is still realized, delay is minimized. Data gathering tree of DMAC is presented in Figure 2.2.

In DMAC, the activity schedule of nodes on the multihop path is designed for waking up sequentially like a chain reaction [54]. In receive state, sensor node listens to the medium to be able to receive the data and if it receives data, it sends an ACK packet. In transmission state, a node will try to send a packet to its next hop and receive an ACK packet. In sleep state, nodes will turn off radio to save energy. DMAC do not use RTS/CTS mechanism which waste considerable amount of energy. Instead of RTS/CTS, to reduce collision during the transmission period of nodes on the same tree level, every node backs off for a period plus a random time within a contention window before packet transmission.

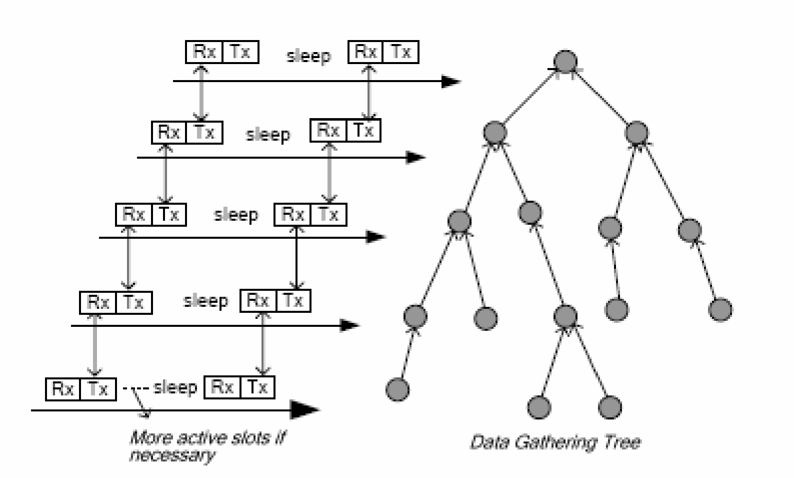


Figure 2.2. Data gathering tree of DMAC [55].

The main advantage of DMAC is to reduce delay with convergecast traffic assumption. By the help of its data gathering tree, delay can be effectively reduced. However, its energy efficiency performance is poor. Collision avoidance methods are not utilized, hence when a number of nodes that has the same schedule (same level in the tree) try to send to the same node, collisions will occur. This is a possible scenario in event-triggered sensor networks [55]. Collision results in energy waste. Another disadvantage of DMAC is the need for time synchronization. Time synchronization means another energy consumption source and the energy consumption increase.

2.1.3. Preamble Sampling Based MAC Layers (B-MAC/WiseMAC/SyncWUF)

Preamble sampling is an effective technique to reduce energy consumption. According to preamble sampling, sensor nodes listen to the medium for a very short period of time. If a node receives a signal during receive period, it continues to listen, otherwise it sleeps. When a node wants to send data, it sends a long preamble signal. In this way, the node that wakes up for a short period of time can receive preamble signal and synchronize itself to the sending node.

Berkeley MAC, B-MAC [56], is one of the most known preamble sampling based MAC layer. To reliably receive data, the preamble length is matched to the interval that the channel is checked for activity. If the channel is checked every 100 ms, the preamble must be at least 100 ms long for a node to wake up, detect activity on the channel, receive the preamble, and then receive the message. B-MAC data transfer structure is presented in Figure 2.3.

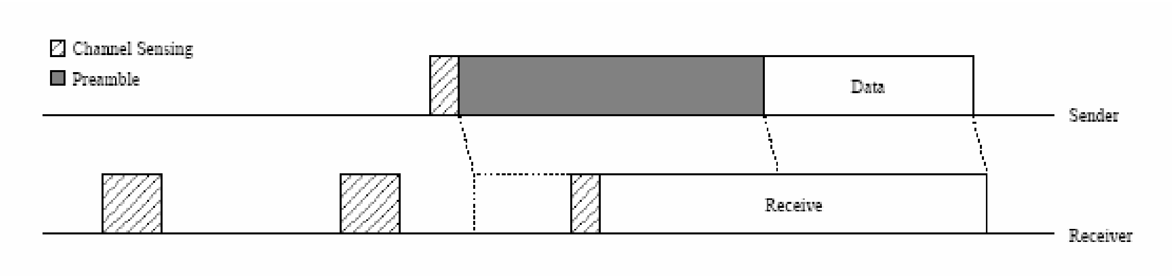


Figure 2.3. Data transfer structure of B-MAC [56].

WiseMAC [57] is another preamble sampling based MAC layer for WSN. The difference between WiseMAC and B-MAC is the length of preamble signal. Unlike B-MAC, WiseMAC remembers the schedule of its neighbors. According to the event inter arrival time and maximum clock drift; it calculates the maximum clock drift between itself and the destination node. While B-MAC sends maximum preamble signal for every data transfer, WiseMAC sends preamble signal with the length of the maximum clock drift difference. Shorter preamble signal reduce energy consumption.

SyncWUF [58] is an improvement on WiseMAC. WiseMAC is very successful for high system loads, but it may waste energy consumption for lower system loads. Low system load requires longer preamble signal and it increases energy consumption. For high system load, SyncWUF behaves like WiseMAC. However, for lower system loads, it sends multiple small packets that include the address of the destination node and a sequence number. These small packets are called as wake up frame (WUF). In this way, receiving node receive only one small WUF, instead of receiving the whole long preamble signal. By the help of the sequence number of the received WUF, receiving node can know when the data will be sent. After receiving a WUF, node sleeps if the data will be sent for it. When the transmitting node sends the data, receiving node wakes up and receives the data. It reduces received bits and saves energy. Simulation studies also show that SyncWUF can save more energy than WiseMAC [58].

The common disadvantage of preamble sampling methods is delay problem. Delay from one to another is the half of one cycle on the average. If one cycle is assumed as 1 second, delay may be more than multiple seconds for large WSNs. Delay is very important for military applications and new techniques must be developed for MILMON to reduce energy consumption and delay at the same time.

2.2. Fixed Allocation Based WSN

Reduction of energy consumption is the most important objective for most of the wireless sensor networks. The easiest way to reduce energy consumption is to turn the radio off, when it is not used [8]. Fixed allocation methods, TDMA or FDMA is extremely suitable for this kind of network. Contention based approach needs some control packets to transmit data. For a successful transmission, it must consume its energy for transmission of data as well as control packets and idle listening. In CSMA based systems, nodes transmit the data without any control. In that kind of systems, collision is always possible. Collision means energy waste. In addition to these, another problem is the listening period. In these methods, nodes listen to the medium even if there is no signal. However, in TDMA or FDMA, links between nodes are established by fixed allocation of frequency or time. If two nodes want to communicate, they open their radio channel for a certain time and for certain frequency. After that period, their radio may be turned off. This is why fixed allocation based MAC layers are the natural way for sensor networks.

TDMA also has some disadvantages. The main problem of TDMA based systems is global time synchronization. In fact, due to precision, efficiency, cost and form factory, the energy or the costs needed to carry out the synchronization algorithm could be higher than the energy saved. On the other hand, many applications or scenarios, if adequately configured, may receive concrete advantages, in terms of energy consumption, that remarkably increase lifetime of each node [59]. In this thesis, global time synchronization disadvantage of TDMA based systems is minimized by minimizing the energy consumption of time synchronization mechanism.

Another disadvantage of TDMA based systems is delay. If a node can transmit its data with a certain schedule, it must wait and it increases delay. Another advantage of TDMA based systems is the need for time synchronization. In this section, some examples of fixed allocation based sensor MAC layer will be presented.

2.2.1. Self Organizing MAC for Sensor Networks (SMACS)

SMACS is an infrastructure building protocol that forms a flat topology for sensor networks. It is a distributed protocol which enables a collection of nodes to discover their neighbors and establish transmission/reception schedules, without any need for master node [60].

After deployment of sensor nodes, they wake up at some random times and it starts to listen to a certain frequency. This frequency can be called as establishment frequency. After listening to the establishment frequency for a random period of time, if it can not receive any invitation signal from other nodes, it transmits a short invitation packet that contains some basic data about the node. This packet called as TYPE1 message. If a node can receive a TYPE1 message from other nodes and if it wants to establish a link, it responds TYPE1 that it will be an invitee. This response is TYPE2 message. There may be more than one TYPE2 message for any TYPE1 message. At that particular time, the node that transmits TYPE1 message has to decide which node it should choose. After its decision, it sends a response to TYPE2 that includes data about the node that is selected. This is TYPE3 message. At the end, TYPE4 is sent by the invitee. In TYPE3 and TYPE4 phases, nodes agree on a certain schedule to communicate.

The main problem is the contention of the communication schedules. For example, nodes A, B, C, D can hear each other; A and B establish a link for ts_1 schedule. In the same way, C and D establish a link for ts_2 schedule. If ts_1 and ts_2 collide, communication is impossible. The practical solution of this problem is to use different frequencies in schedules. If it is assumed that nodes can tune carrier frequency to different bands, schedules can be defined as a pair of time and frequency. When time periods of a certain schedule collide with another schedule, because of the different frequencies, there will be no collision.

As times goes on, there will be some subnets in the wireless sensor networks. These subnets unite with each other, when new links are established. At the end, all sensor nodes construct a connected network.

SMACS is an infrastructure building protocol that forms a flat topology for sensor networks. It is a distributed protocol which enables a collection of nodes to discover their neighbors and establish transmission/reception schedules, without any need for master node [60]. Although flat topology has some advantages, it has also some difficulties. Flat topology requires a separate network layer, while cluster based approach has implicit network layer in itself. However, the most serious handicap of SMACS is that there is no sensor node that can support the requirements of SMACS. According to SMACS, node must support frequency multiplexer and there is no such a node up to now. A sensor network should be applicable on the existing node models.

2.2.2. LEACH: Low-Energy Adaptive Clustering Hierarchy

LEACH is a self-organizing, adaptive clustering protocol that uses randomization to distribute the energy load evenly among the sensors in the network [28]. In LEACH, the nodes organize themselves into local clusters, with one node acting as the local base station or cluster-head. If the cluster heads were chosen a priori and fixed throughout the system lifetime, as in conventional clustering algorithms, it is easy to see that the unlucky sensors chosen to be cluster-heads would die quickly, ending the useful lifetime of all nodes belonging to those clusters. Thus LEACH includes randomized rotation of the high-energy cluster-head position such that it rotates among the various sensors in order to not drain the battery of a single sensor. In addition, LEACH performs local data fusion to decrease the amount of data, further reducing energy dissipation and enhancing system lifetime.

Sensors elect themselves to be local cluster-heads at any given time with a certain probability. These cluster head nodes broadcast their status to the other sensors in the network. Each sensor node determines to which cluster it wants to belong by choosing the cluster-head that requires the minimum communication energy. Once all the nodes are organized into clusters, each cluster-head creates a time schedule for the nodes in its cluster. This allows the radio components of each non-cluster-head node to be turned off at all times except during its transmit time, thus minimizing the energy dissipated in the individual sensors.

Once the cluster-head has all the data from the nodes in its cluster, the cluster-head node aggregates the data and then transmits the compressed data to the base station. Since the base station is far away in the scenario we are examining, this is a high energy transmission. However, since there are only a few cluster-heads, this only affects a small number of nodes [39]. As discussed previously, being a cluster-head drain the battery of that node. In order to spread this energy usage over multiple nodes, the cluster-head nodes are not fixed; rather, this position is self-elected at different time intervals. An example cluster head reselection scenario is presented in Figure 2.4 and Figure 2.5 [42].

The decision to become a cluster-head depends on the amount of energy left at the node. In this way, nodes with more energy remaining will perform the energy-intensive functions of the network. The lifetime of LEACH can be thought in terms of rounds. In each round, there are certain phases that must be done. These phases are as advertisement, cluster set up, schedule creation, and data creation.

In advertisement phase, a distributed cluster head selection algorithm is run in every node. At the end of the algorithm, cluster heads advertise themselves to the other nodes. In cluster set up phase, non-head nodes decide which head node will be their cluster head and send an acknowledgement for it.

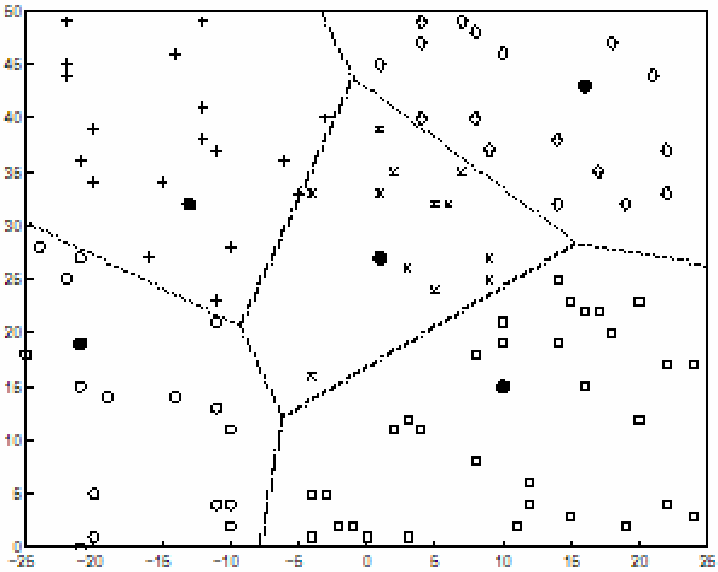


Figure 2.4. Cluster head selections before reselection in LEACH

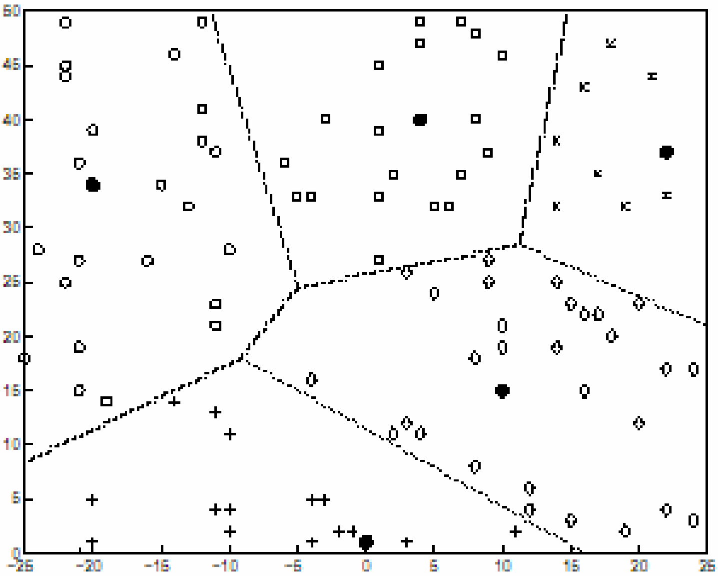


Figure 2.5. Cluster head selections after reselection in LEACH

In schedule creation, cluster head gets all acknowledgments and creates a time schedule. LEACH uses centralized approach for time slot scheduling. This schedule is broadcasted to all nodes in the cluster. Data Transmission is steady state position of LEACH. All data are sent in this phase. After a certain time, which is determined a priori, the next round begins with each node determining if it should be a cluster-head for this round and advertising this information, as described in advertisement phase.

All performance comparisons of LEACH are based on minimum transmission energy (MTE) routing and direct communication. In MTE routing, each node sends a message to the closest node on the way to the base station [61]. Comparison study shows that LEACH achieves between 7 times and 8 times reduction in energy compared with direct communication and between 4 times and 8 times reduction in energy compared with MTE routing.

Although LEACH achieves power consumption decrease dramatically, the assumption of LEACH system is not scalable. According to LEACH, each node can receive signals from all the other nodes. However, it is not possible in every environment, especially in larger areas. If sensor network is designed to operate for relatively large areas as in MILMON, every node cannot receive all signals. Another unrealistic assumption of LEACH is about the existence of data. LEACH assumes that every node has data to send every time. However, in most of the time, there is no data to send in sensor network. For example, MILMON is designed for intrusion detection and the existing of an intruder is not likely to occur most of the time.

Because of the extra duties of cluster heads, cluster heads have to spend more power than regular nodes. The solution of this problem is to change cluster heads periodically. Unfortunately, the solution of quick power drain of cluster heads is a cause for spending power. Periodic cluster head selection and reorganization of network for the new cluster heads may consume considerable energy.

In order to maintain balanced energy consumption, LEACH suggests that each node probabilistically become a cluster head. The probabilistic approach may lead to dramatic energy waste. Unless each node selects its probability of becoming a cluster head wisely, the performance of the network may be far from optimal. This disadvantage is investigated and an optimal cluster head algorithm is proposed in [62]. Using this optimal probability can reduce energy consumption and extend network lifetime, when it is compared with the original LEACH.

2.2.3. Two Tiered Architecture

Two tiered architecture is another sensor network model that uses TDMA technique [63]. This sensor network architecture is designed for monitoring structural health of buildings. It is based on clustered approach. Sensor nodes are clustered and cluster heads are special nodes called local site masters (LSM). LSMs are deployed one by one manually and the power source of LSMs is the power source of the building. So, there is no power limitation on LSM. LSMs collect the data from their sensor nodes. LSMs construct a higher level of network and send the collected data to a center. According to analysis of two tiered sensor network, the expected lifetime of network is about 18 months. It is very impressive and acceptable. However, because of LSM limitations, the application of this network in open areas is extremely difficult. There must be lots of LSM units and these LSM units should be deployed manually. Moreover, LSMs must have unlimited source power. Because of these limitations, it is not suitable to use this system as military monitoring system.

2.2.4. TRAMA / FLAMA

The traffic-adaptive medium access protocol (TRAMA) [64] is introduced for energy efficient collision-free channel access in wireless sensor networks. TRAMA reduces energy consumption by ensuring that unicast and broadcast transmissions incur no collisions, and by allowing nodes to assume a low-power, idle state whenever they are not transmitting or receiving. Its election algorithm that uses traffic information decides on which node can send data at a certain slot. If a node is unlikely to send data, according to current traffic information, it sleeps.

Accordingly, TRAMA consists of three components: the Neighbor Protocol and the Schedule Exchange Protocol, which allow nodes to exchange two-hop neighbor information and their schedules; and the Adaptive Election Algorithm, which uses neighborhood and schedule information to select the transmitters and receivers for the current time slot, leaving all other nodes in liberty to switch to low-power mode.

The main advantage of TRAMA is its collision free structure. If there is no collision, energy waste is lower. Another advantage of TRAMA is its traffic adaptive mechanism. It can adapt with its traffic information and it can save considerable amount of energy. However, TRAMA needs random access based period for synchronization and neighbor discovery. Random access based period must be repeated to maintain the topology of WSN. During random access based period, a sensor node must receive or transmit. Random access based periods is an important energy waste source for TRAMA.

Flow Aware Medium Access, FLAMA [65] is an improvement over TRAMA for energy saving. It also uses random access based period. However, instead of listening to whole time slot, it only listens the beginning of the slot. If it can receive a signal, it continues to receive. If it can not get ant signal, it sleeps. Simulation results indicate that FLAMA outperforms TRAMA and S-MAC in terms of reliability, and energy savings.

FLAMA achieves significant improvement in delay performance for scheduling based protocols.

3. A NEW TDMA BASED SENSOR NETWORK FOR MILITARY MONITORING (MILMON)

3.1. Overview

Because of the unattended structure of the sensor nodes, the scarcest resource in sensor networks is power. Power consumption can be divided into three domains, as sensing, communication, and data processing domains and dominant factor in energy consumption for sensor nodes is communication [28]. Not only transmission but also receiving is the main cause of energy waste. The easiest way to reduce energy consumption is to turn the radio off, when it is not used. Fixed allocation methods, TDMA or FDMA is extremely suitable for this kind of network.

In this section, a new TDMA based sensor network system, which can be used for military monitoring, MILMON, is presented. In order to realize MILMON, time synchronization and time slot assignment mechanisms are developed. In order to enhance the network in terms of power usage, data indicator slot mechanism (DISM) is also proposed. The main design considerations of MILMON are energy consumption, delay, fault tolerance. Before introducing the mechanisms of MILMON, basic assumptions and sample scenarios of MILMON are presented.

3.2. Basic Assumptions and Sample Scenarios

The potential applications of wireless sensor networks are highly varied. Environmental monitoring, condition based maintenance, smart spaces, military, precision agriculture, transportation, factory instrumentation, inventory tracking are just some of the sample application areas. Behaviors of one specific sensor node can be totally different from the others even under the same conditions. In this case, it is clear that protocols that are developed for a WSN may not be optimal for another WSN. Sensor network systems should be application specific [39]. Firstly, the assumptions of MILMON are presented. After the assumptions, sample application scenarios help to understand the assumptions and mechanisms of MILMON.

The assumptions of the sensor network system are as follows. Sensor nodes will be immobile. Mobile cases can be investigated for further analysis. Power consumption model of sensor node is the same as described in [66]. This model is one of the mostly used models in sensor network simulation analysis researches. Radio channel is symmetric. If Node A can receive a signal of Node B, Node B can also receive the signal of Node A. Sink node's power source theoretically infinite. The sink is most probably a laptop or PDA. It can get its energy from the wires or it can be charged with solar energy.

The assumptions, which have been mentioned above, are valid for most of the sensor networks. There are some additional assumptions specifically for MILMON. MIMON uses the beacons of a high range transmitter. This can be realized by the existence of a high range transmitter in the application area of MILMON. The location of the high range transmitter can be anywhere. The only restriction is that all sensor nodes must receive the synchronization beacons. In this thesis, for the simplicity, the sink is assumed to have a high range transmitter.

The sink node has high range transmitter as well as low range transmitter. In this way, sink can use its low range transmitter to communicate with its neighbor nodes and it can send broadcasts for all nodes. The node that holds the high range transmitter has a GPS. In this way, clock drift of sink is near zero [67]. The application area of MILMON is restricted with the coverage area of the sink. In order to increase the application area of MILMON, using more than one sink is a must. Although MILMON architecture is open to use more than one sink, this thesis assumes the existence of only one sink.

A typical application scenario for MILMON is military monitoring against terrorists. As it is known, PKK terrorists use certain paths from Iraq to Turkey. The simplest way of preventing PKK terrorist actions is to stop them cross the border. However, because of the geographical structure of the region, to detect and defend the region is not an easy mission. Today, Turkish troops try to control the regions with patrol activities. Sometimes, it may cause causalities. If a military monitoring sensor network is deployed in critical regions of Iraq and Turkey border, it can report the arrivals of the terrorist as soon as possible with a cost effective manner. In MILMON, the environment must be covered by the beacons of a powerful sink. The power of the sink is theoretically infinite. The sink can be assumed as a PDA or a laptop with replaceable battery. The sink sends its beacon to the sensor nodes, Sensor nodes synchronizes to the sink directly and organize themselves, so that, when sensor nodes detect an intruder, they send their data to a sink. In this way, soldiers can defend against the intruders.

3.3. Basic Mechanisms

3.3.1. Time Synchronization with a High Range Transmitter

According to assumption of this sensor network, every node in the system can receive the signals of the high range transmitter. It transmits a broadcast signal to sensor nodes at the beginning of each time frame. These broadcast signals synchronize the network.

Packet latency error is an important source for time synchronization errors. In [68], Kopetz and Schwabl have decomposed message latency into four distinct components: send time, access time, propagation time and receive time. This synchronization schema and TDMA based MAC layer eliminates send time and access time with the periodic broadcast of the sink. Because of the close sensor nodes, propagation time is near zero. Receive error can be minimized with MAC-layer time stamping as in Flooding Time Synchronization Protocol, FTSP [69]. Another time synchronization error is clock drift and clock drift is minimized with linear regression as in [70]. The last error source is clock instability. The solution for clock instability is to use the most recent broadcast receives times. In this way, old receive times cannot affect the result.

Sensor nodes that can receive the broadcast can synchronize directly. If the node is not in the range of the sink, it synchronizes with other synchronized nodes as in the flooding technique used in FTSP [69]. When a node is synchronized, it broadcasts a beacon. The node that can receive this beacon can synchronize with the synchronized sender node. In this way, the entire network is globally synchronized to the sink.

This global time synchronization scheme that is based on the periodic broadcast of the high range transmitter is called as SyncHRT.

3.3.2. Fault Tolerant Distributed Time Scheduling Mechanism

TDMA based system has a natural advantage over the other methods for WSN. However, TDMA based systems has also special problems and one of the main problem of TDMA based MAC layers is time slot distribution. Slots must be distributed so that no two sensor node collide each other.

Almost all sensor network architectures that use TDMA produce its time schedule centrally. Cluster head collects the data about its sensor nodes and produces the time schedule of its cluster. Time schedule is sent to nodes by the cluster head. In most of the centralized time scheduling algorithms, the protocol to collect data about the nodes is contention based. This can be a serious problem for power sensitive systems. However, in distributed time scheduling algorithms, there is no need for communication between nodes and the sink directly. This leads to power saving.

A new delay sensitive, energy efficient, fault tolerant distributed time slot assignment algorithm, ft_DTSM, is proposed to realize MILMON. The design considerations of ft_DTSM are delay, energy consumption, reliability.

3.3.3. Data Indicator Slot Mechanism

Data indicator mechanism (DISM) is proposed to reduce energy consumption of MILMON. A sensor node has to check its children whether they have data or not. According to DISM, one special slot of every sub time frames is reserved for data indication. When a sensor node senses an event or when it has data to send, it sends a signal in the data indicator slot (DIS). All the other nodes listen to DIS. If a node receives a signal in DIS, it continues to listen. Otherwise, it stops and sleeps.

All the details about basic mechanisms proposed for MILMON are presented in Chapter 4.

4. BASIC MECHANISMS OF MILMON

4.1. Using High Range Transmitter on Wireless Sensor Network Time Synchronization (SyncHRT)

Time synchronization service is an important part of any distributed system including WSN. Local time synchronization may be sufficient for many applications, like data fusion. However, there are some sensor network systems that require global synchronization, like TDMA based WSNs. Although there are many WSN time synchronization schemas, most of them are focused on optimizing scalability and precision. However, the scarcest resource in most of the WSNs is energy. SyncHRT is proposed to reduce energy consumption and to increase precision. In SyncHRT, a node, typically the sink with no energy constraint, can send broadcast signal to all sensor nodes. Sensor nodes receive the global reference broadcast and synchronize themselves to sink. Although the existence of high ranger transmitter is not a common assumption for WSN, there are WSN systems that are based on the sink with high range transmitter [71].

Wireless sensor network time synchronization methods can be classified in two main methods: receiver to receiver synchronization and sender to receiver synchronization [72]. In receiver to receiver method, receivers receive the beacon of a common sender. However, receivers exchange time data with other receivers not to the sender. Reference Broadcast Synchronization, RBS, is one of the most known examples of receiver to receiver synchronization for WSN [70]. In sender to receiver method, the receiver synchronizes itself with the sender by the help of the sender’s message. Time Sync Protocol for Sensor Networks, TPSN [73], is an example of sender to receiver WSN time synchronization. The first step of TPSN is to create a hierarchical topology. After two way message exchange between the root and its neighbors, the neighbors synchronize themselves with the root and start a new message exchange with other unsynchronized nodes. Flooding Time Synchronization Protocol, FTSP [69], is another sender to receiver WSN time synchronization method. While TPSN uses two way message exchange between sender and receiver, FTSP needs only one message transmission from sender to receiver. FTSP also uses a MAC-layer time stamping that reduces most of the synchronization errors.

If the sink transmits its signals to a higher range, sensor nodes can synchronize themselves with the sink directly. This idea helps to produce higher precision, save more energy and reduce run time. In this section, a global time synchronization schema for WSNs, SyncHRT is proposed with the assumption of using high range transmitter at the sink.

4.1.1. Literature Survey about Time Synchronization

Time synchronization is a fundamental service for distributed systems. There are many different synchronization mechanisms for different systems [74-77]. NTP [74] is the most famous synchronization mechanisms used in the Internet. NTP has a multi layer structure and the nodes in the highest layer are synchronized externally, for instance with Global Positioning Systems (GPS). A node synchronizes itself to the one level higher node. NTP requires exchanging messages frequently and consumes high amount of energy. Moreover, NTP does not account for the frequent changes in the topology. However, sensor nodes are also prone to failures and frequent changes in the topology [3].

Clock Sampling Mutual Network Synchronization (CS-MNS) is able to achieve microsecond global synchronization accuracy for single-hop or multiple-hop network topologies in mobile or static wireless ad hoc and sensor networks [78]. Different from existing mutual network synchronization approaches, the timing information is exchanged explicitly by using periodic time stamp packets. These packets can be, for instance, the same beacons used in the IEEE 802.11, which makes CS-MNS compatible to popular standards.

Reference Broadcast Synchronization technique uses a broadcast signal to synchronize the nodes [69]. This broadcast signal is received by one hop neighbors of the sender. The receivers of the same broadcast form a cluster. The nodes in the same cluster, exchange signals to learn relative offsets. This procedure is repeated several times. The data collected from the signal exchanges are used in linear regression. In this way, clock drifts between the nodes are also extracted. In order to extend this strategy for multihop synchronization, gateways are defined. Gateway is a node that belongs to more than one cluster. Gateway nodes can transform time stamps from one cluster to another. The most important drawback of RBS is its complexity. If the number of broadcast signal used for linear regression is m, and the number of nodes in a cluster is n, RBS requires $mn(n-1)$ signal exchanges between nodes. RBS should be considered for every cluster separately. High complexity means high signal exchange as well as high energy consumption. In addition to energy consumption problems of RBS, its implementation is another handicap. RBS requires senders to synchronize the receivers and the selection of the senders is also very problematic.

Tiny-Sync and Mini-Sync [79] is a synchronization technique that assumes clock drifts are linear. Nodes exchange signals with their neighbors. After collecting enough data, they calculate linear clock drift parameters with linear regression. When every node learns the parameters about its neighbor, global time synchronization can be achieved. However the number of signal exchange is still high. If m is the number of broadcast to calculate linear regression, n is the number of nodes in the network, and k is the expected number of neighbors, total number of signal exchanges is mnk.

Lightweight Time Synchronization LTS [80] requires lower complexity at the expense of lower precision. This technique is based on construction of a tree. Child synchronizes itself to its parent with the exchange signals. In this case, signal exchange number decreases to $2(n-1)$, where n is the number of nodes. Time-Sync Protocol TPSN [73] is very similar to LTS. It also constructs a tree and parents synchronize its children as in LTS.

FTSP is based on flooding of the time stamp of the master node [69]. It has a unique mechanism called as MAC-layer time stamping that can eliminate most of the packet latency errors very effectively. In MAC-layer time stamping, sender node sends multiple time stampings, instead of single time stamp. Each send time of the time stamps are recorded. These records are used to create final time stamp. The sending message is composed of all time stamps and the final time stamp. Sensor nodes collect the latest six messages to perform linear regression. The result of this linear regression gives clock drift and offset to the master node. Although precision performance of FTSP is very successful when it is compared to the other existing methods, flooding technique used in FTSP may result in overhearing of unnecessary data and energy waste.

4.1.2. Time Synchronization Errors

Before giving the details of SyncHRT, time synchronization errors are discussed. Time synchronization algorithms aim at minimizing time synchronization errors. The most important error sources of time synchronization are packet latency, clock skew and clock instability.

In [68], Kopetz and Schwabl have decomposed message latency into four distinct components. The first one is the time spent at the sender to construct the message. It includes operating system kernel delays, context switches and transfer time from host to network interface. The second is access time. It is the time spent to access to the media at MAC layer. In contention based methods, message must wait until the media is clear. In 802.11 [50], RTS/CTS schemes take time.

In TDMA, host must wait for its slot to send the message. The third component of packet latency is propagation time. It is the time required for a signal to travel from one point to another across the medium. The last latency component is receiver error. This is typically the time for constructing the message in the receiver. If the message is labeled in very low level in the operating system, receive error can be minimized. Receiver error is characterized in [70] for Berkeley Mote [52] sensor nodes. Motes use a minimal event-based operating system developed by Hill et al. specifically for that hardware platform called TinyOS [81]. Receiver’s jitter is maximum 53.4 μs. The distribution of offset between sensor nodes is Gaussian ($\mu=0$, $\sigma=11.1$).

Clock skew is one of the main problems with the behavior of the oscillators [82]. Time of a node is calculated as the integration of the node’s oscillator. The change or error in the oscillator results in error of the clock. Clock skew is related with the agreement between expected and actual frequencies of the oscillator. The maximum difference is specified by vendors. In most of the cases, crystal oscillators used in economical sensor nodes are accurate to 1-100 μs in one second [82].

Clock instability is about the changes in frequency of the oscillator over time. Frequency may change due to environmental factors. This kind of changes is classified as short term changes. Long term stability changes are mostly related with the aging problems of the oscillators.

4.1.3. SyncHRT System Architecture

SyncHRT is a new global synchronization system for TDMA based WSN under the assumption of the sink's global broadcast signal. The main objective of SyncHRT is to develop a synchronization technique with low energy consumption and high precision. According to SyncHRT, the sink sends a periodic broadcast signal so that most of the sensor nodes can receive it. Sensor nodes that receive the broadcast of the sink synchronize themselves to the sink.

Sink node is assumed to have a dual range transmitting power. The sink can use its low range transmitter to communicate with its neighboring nodes and high range transmitter to broadcast for all nodes. Clock drifts are assumed as 30 μs in one second. Sink node has GPS and its clock drift of is about 200 ns [83]. Clock drift of the sink is neglected when it is compared with receive error and clock drift of regular sensor nodes. Time frame is assumed as one second. Underlying MAC layer of the system is TDMA based and time slot of the sink is not used for another sensor node. In this way, when the sink decides to send its broadcast, it never waits to access to the media.

Time synchronization systems have to handle packet latency errors, clock drifts instability problems.

Packet latency has four major components, send time, access time, propagation delay and receive time [68]. In many sensor networks, nodes are deployed densely [3], and distance between nodes is in the order of tens of meters. Due to short distance between sensor nodes, propagation delay is very small and can be neglected.

Send time and access time errors are the most indefinite errors, and SyncHRT eliminates send time and access time with the TDMA based MAC layer properties. The sink sends its broadcast in its reserved slot, it does not wait and access time becomes zero. Because of the reserved time slot, sending time of the broadcast is definite. In this case, broadcast packet can be constructed much earlier than its sending time. There is no need to wait for constructing the packet. The constructed packet can be placed into buffer. Interrupt of the timer of the GPS can be used to send the broadcast. In this way, with the help of the properties of TDMA based MAC layer, send time and access time errors can be eliminated. Receivers receive the broadcast signal without send time or access time errors.

SyncHRT minimizes the remaining errors with MAC-layer time stamping which is proposed in Flooding Time Synchronization Protocol, FTSP [69]. In MAC layer time stamping; when the sink sends its synchronization packet, it produces multiple time records and calculates the final time stamp by averaging the time records that include minimum synchronization error. Only the final time stamp is sent to the receivers in a synchronization packet. The nodes, that can receive the packet, also produce multiple records and its final time stamp with the same technique. The latest time stamps are used for linear regression. If the node is not in the range of the sink, it synchronizes with other synchronized nodes as in the flooding technique used in FTSP [69]. When a node is synchronized, it broadcasts a beacon. The node that can receive this beacon can synchronize with the synchronized sender node. In this way, the entire network is globally synchronized to the sink.

Another problem that should be handled is clock skew. If clock skews between nodes are perfectly stable, clock of node *i* can be modeled as

$$t_i(t) = a_i t + b_i, \tag{4.1}$$

where a_i is the clock drift and b_i is the offset. In ideal case, all a_i's and b_i's are the same. Clock skew is the difference between a_i's. Linear regression of differences between the sink's time and the sensor node's time can give the skew between the sink and the sensor node. In this way, every sensor node can calculate clock skew with respect to the sink; adjust its clock according to its calculated skew. This method eliminates or decreases clock skew error.

In order to calculate linear regression, the past broadcast signal receive times should be used. However, because of the clock instability, clock skew is not constant every time. It may change with environmental factors or aging problems. The solution for clock instability is to use the recent broadcast receives times. In SyncHRT, as it is used in FTSP [69], the latest six time stamps are used for linear regression.

4.2. Delay Sensitive, Energy Efficient and Fault Tolerant Distributed Slot Assignment Algorithm for Wireless Sensor Network Under Convergecast Data Traffic (ft_DTSM)

One of the main problems of TDMA based systems is slot assignment. In this section, a new distributed time slot assignment mechanism (ft_DTSM) is presented for TDMA based sensor networks [84]. The most important design considerations of this new mechanism are energy efficiency, delay, reliability.

Distributed time slot assignment is not a new topic for wireless networks. DTSAP (Dynamic Distributed Time Slot Assignment Protocol) [85-87], FPRP (Five-Phase Reservation Protocol for Mobile Ad Hoc Networks) [88], E-TDMA (Evolutionary-TDMA Scheduling Protocol) [89], HRMA (Hop-Reservation Multiple Access) [90] are examples of distributed scheduling protocols for ad hoc networks. Most of the ad hoc network algorithms are developed for peer to peer data traffic. However, data traffic in WSN is mostly from sensor nodes to the sink or simply convergecast Existing slot assignment algorithms for ad hoc networks can not satisfy energy and delay requirements of WSNs under convergecast traffic. Wireless sensor networks need a delay sensitive and energy efficient slot assignment algorithm under convergecast data traffic. This is why slot assignment algorithms developed for ad hoc networks can not be directly applied to WSN.

There are some researches about distributed time slot assignment for sensor networks. Patro et.al. has proposed Neighbor Based TDMA Slot Assignment Algorithm for WSN [91]. A mobile agent visits every node and assigns a proper slot. This method reduces required number of slots and increases channel utilization. However, it is not energy efficient. Copying and running the agent consumes high amount of energy. Kanzaki et. al. has also proposed an adaptive slot assignment protocol for WSN [92]. However, the main design objective is the channel bandwidth, not delay or energy efficiency. Another distributed slot assignment algorithm for sensor networks is presented in [93]. It reduces delay for broadcast, convergecast, and local gossip traffic patterns for different grid topologies. However, in many sensor network applications, sensor nodes are deployed randomly. In addition to this difficulty, it does not consider energy consumption; its design consideration is only to minimize delay. SMACS [60] uses a different distributed time scheduling algorithm. After a series of handshaking signals, neighbor nodes can agree on a frequency and time pair to construct a link. SMACS produces a scalable and reliable flat network. However, SMACS needs FDMA as well as TDMA, but sensor nodes are so tiny and limited that current sensor nodes cannot meet the requirements of SMACS. DRAND [94] is a randomized dining philosophers algorithm for TDMA scheduling of wireless sensor networks. This algorithm is the first distributed implementation of RAND [95], a commonly used, centralized channel assignment algorithm. Randomized dining philosophers approach is scalable and robust. However, in DRAND, before having a schedule, nodes communicate with each other using a contention based MAC protocol, and it increases energy consumption. In [96], another distributed slot assignment algorithm is proposed. It also uses CSMA/CA to schedule the slots and it consumes high energy during slot assignment period, like DRAND [94]. μMAC has another slot assignment mechanism that includes a contention period [97]. Oriented Link Scheduling is also distributed time slot assignment algorithm, but its main design considerations are complexity of the algorithm and number of used time slot minimization [98]. It does not deal with delay or fault tolerance. In [99], a new slot assignment algorithm is proposed to minimize energy consumption by the help of special control slots. However, it can not reduce delay.

TRAMA [64] is a TDMA-based sensor network system and it includes a distributed slot assignment mechanism. It has random access period to be able to assign proper slots, and its random access period is also contention based. In TRAMA, all the nodes have to listen to medium in random access periods. It increases energy consumption of TRAMA. FLAMA [65] is another slot assignment algorithm that uses random access period, like TRAMA [64]. FLAMA adapts medium access schedules to the traffic flows exhibited by the application. By the help of application awareness, FLAMA achieves significantly smaller delays (up to 75 times) when compared to TRAMA with significant improvement in energy savings and reliability. However, contention based structure of FLAMA still consumes considerable amount of energy. DTSM [100, 101] is another distributed time slot assignment scheme for wireless sensor networks. It uses tiny time slots like FPRP [88]. DTSM distributes the slots based on hop numbers of the nodes. In this way, it achieves significant delay reduction. However, it is very vulnerable against single point of failure. Because of high failure rates of sensor nodes, it is a significant handicap.

There are a number of advantages of ft_DTSM design over the existing designs. Firstly, unlike existing slot assignment protocols that includes 802.11 like contention sessions, nodes in ft_DTSM contend in time slots, like FPRP [88]. In 802.11 [50] like random access based sessions, all nodes must listen to medium and keep their radio circuits open during contention based session. This strategy may consume high amount of energy. Contention in tiny time slots results in lower energy consumption. Another important feature of ft_DTSM is its convergecast traffic aware design. Most of the time, wireless sensor networks use convergecast traffic pattern. In convergecast traffic, data relays from nodes into the sink. The sink collects all the data produced by the nodes. ft_DTSM assumes that nodes always forward data to their neighbors that are with lower hop number. In order to decrease delay, ft_DTSM assigns the slots on the basis of the hop number of the nodes. Unlike the other slot assignment algorithms, DTSM allows to assign the same slots into the nodes within two hop region, if the assignment allows convergecast traffic. Another advantage of ft_DTSM is its reliable structure. It can distribute the slots so that sensor network can recover itself against single point of failures.

4.2.1. Literature Survey about WSN Time Slot Assignment

Instead of distributed algorithms, many TDMA based MAC protocols for sensor networks prefer to assign time slots centrally [102]. Especially, sensor networks based on small clusters prefer centralized approach as in [103, 104]. Sensor nodes connect to the nearest cluster head. Cluster head collects data about the nodes in its cluster and creates a schedule centrally. Cluster head broadcasts this schedule to its nodes. However, this approach has disadvantages. Data about sensor nodes must be forwarded to cluster head with a contention based system like 802.11 which increases energy consumption. Inter-cluster interference is another problem of these sensor networks. In most of the cases, interference is handled with CDMA approach which requires considerable computation power. Even if these disadvantages can be handled, all sensor networks are not cluster based and there is no easy way to implement central time slot assignment for wireless sensor networks that are not cluster based.

Most of the existing slot assignment algorithms for sensor networks are based on 802.11 like contention periods. SMACS [60], DRAND [94], TRAMA [64], FLAMA [65] can be classified in that kind of networks. They have a random access period. Slot request and slot grant data exchange is performed in this period. FLAMA [65] is presented as an example.

Time slot organization of FLAMA is presented in Figure 4.1. It has two major components, scheduled access and random access. Scheduled access is basically used for data transmission. In this period, there is no contention. However, as the name suggests, during the random access period, nodes perform contention-based channel acquisition and thus signaling packets are prone to collisions. All the nodes have to listen to medium in random access period. Although FLAMA can save considerable amount of energy in scheduled period, very serious energy waste still takes place in random access period.

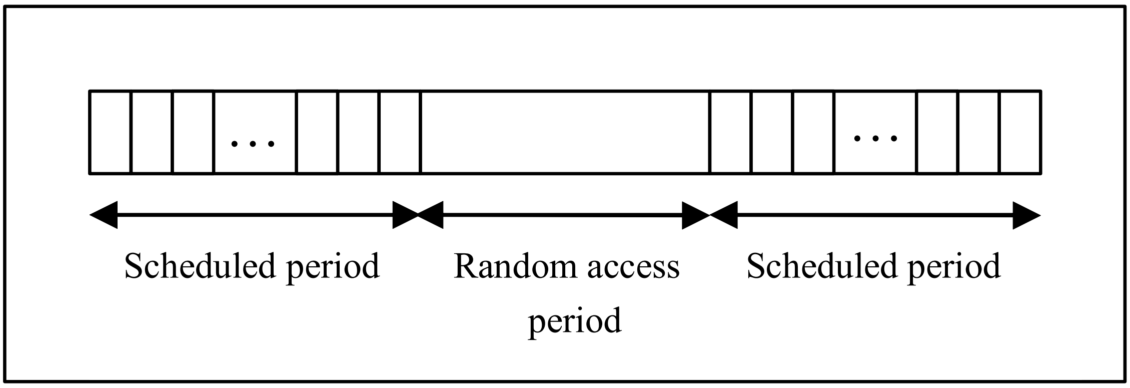


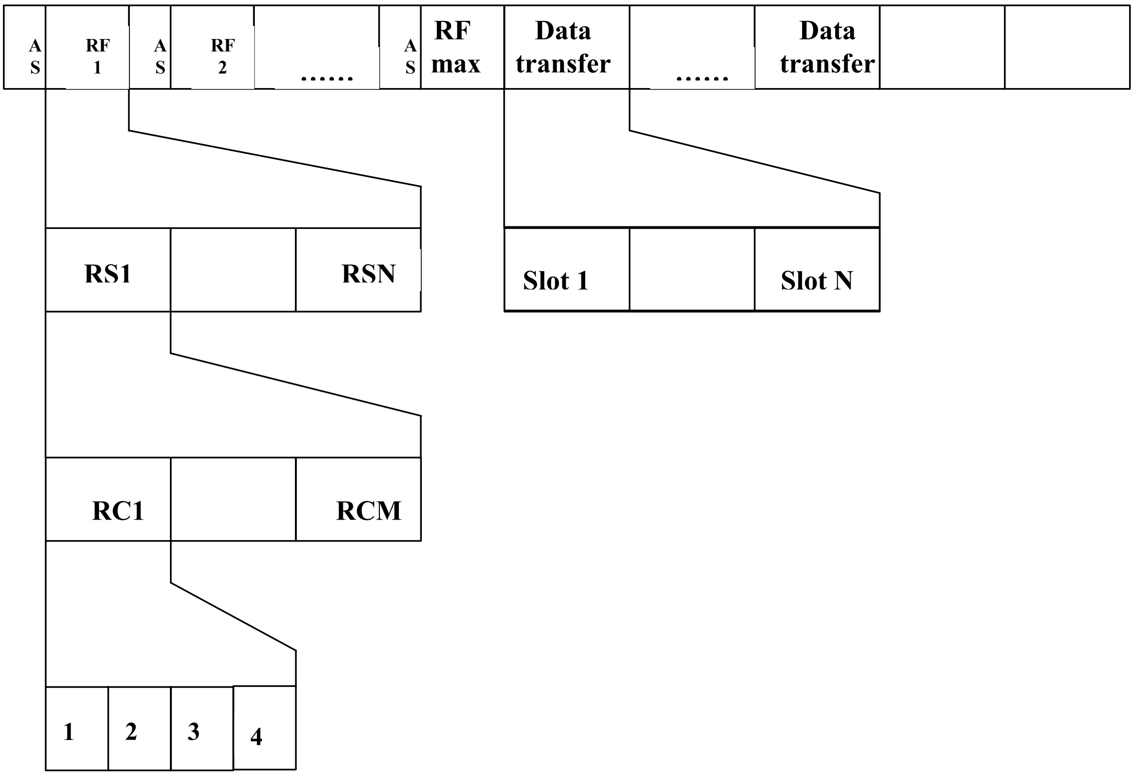
Figure 4.1. Time slot organization of FLAMA

Another approach for distributed slot assignment is to use tiny time slots. In this approach, all nodes are assumed as synchronized. Nodes send their requests and grants in tiny slots. After a series of handshaking, if a node can receive the required signal successfully, it gets the slot. Five-Phase Reservation Protocol, FPRP [88], is one of the most known examples of this class of distributed slot assignments. FPRP is a slot assignment protocol which uses five-phase reservation process to establish TDMA slot assignments [88]. It is fully distributed and can run parallel in the network. In other words, it is entirely insensitive to network size. Unlike TRAMA or FLAMA, it does not need the support of additional MAC protocol like 802.11.

In FPRP, slots are assigned in reservation frame which is a collection of reservation slots. The number of reservation slots is equal to the number of slots that will be assigned. Each reservation slot corresponds to a slot. Reservation slots are composed of a certain number of reservation cycles. The number of reservation cycles in each reservation slot is a fixed parameter of the algorithm. The nodes perform a special five phased handshaking procedure in a slot reservation cycle. All the phase handshaking are sent in slots, not in random access period. The first phase is reservation request. In this phase, the node that has no valid slot sends a request signal with a certain probability. The nodes that do not send a signal listen to the medium. In the second phase, collision report phase, nodes that receive a jammed signal in phase 1, send a collision report. In reservation confirmation phase, a node that has sent a request in phase 1 and did not receive any signal in phase 2 allocates the current slot. In this case, it sends a confirmation signal in phase 3. In reservation acknowledgment phase, a node that receives a signal in phase 3 sends a signal. At the end, in phase 5, a node that receives a signal in phase 4 sends a signal. A reservation cycle is composed of these phases. If a node can achieve to allocate a slot in a reservation cycle which belongs to n^{th} reservation slot, it allocates n^{th} slot. The nodes that can not allocate the current slot may contend in the next reservation cycle. In this way, all the nodes can have one valid slot.

4.2.2. ft_DTSM Slot Organization

Instead of random access period like TRAMA [64] or DRAND [94], ft_DTSM uses tiny time slots to exchange scheduling signals [84]. The slot organization of ft_DTSM is presented in Figure 4.2. The number of reservation frame, the number of reservation slot and the number of reservation cycles for each reservation slot is fixed parameters of ft_DTSM.



AS: Advertisement Slots, RF: Reservation Frame, RS: Reservation Slot, RC: Reservation Cycle
Signal exchanges; 1: request slot, 2: collision report slot, 3: confirmation slot, 4: acknowledgement slot

Figure 4.2. Slot organization of ft_DTSM.

ft_DTSM assigns the slots in reservation frames. Every reservation frame begins with advertisement slots (AS). There is one corresponding tiny slot for every time slot in AS. All the nodes with a valid slot send a special signal in the corresponding slot in AS. All the nodes that have no valid slot listen to all tiny slots in AS. If a node can get valid signal more than a certain number, fault tolerance parameter, it means the reservation frame for its hop number is about to begin and it can compete for slot assignment. Every reservation frame is used for corresponding hop numbered nodes. According to this hop numbered structure, in the first reservation frame, the nodes with hop number one can get slots. After that, the nodes with hop number two get the slots and so on.

Slot assignments for a specific hop number are performed in reservation slots. Each reservation slot corresponds to a specific available data transmission slot. In this case, the number of reservation slots and the number of available data transmission slots for a particular hop number are the same. There are a certain number of reservation cycles in each reservation slot. The number of reservation cycles for each reservation slot is a constant and it is a parameter of ft_DTSM algorithm. There are four tiny slots in each reservation cycle and signal exchanges are realized in these slots.

4.2.3. Signal Exchange

In traditional slot assignment, no node within two hop radius can get the same slot. This necessity is sourced from the hidden terminal problem. However, if the only traffic in the network is convergecast, this rule can be relaxed. If data flow through the higher hop numbered nodes to lower hop numbered nodes, convergecast traffic can be realized. ft_DTSM assumes that a node with hop number h sends its data only to a node with hop number $h-1$. In such a network, the only collision that must be handled is between the nodes with hop number h and the nodes with hop number $h-1$. Even if they are in two hop neighborhood, the nodes that are with the same hop number can get the same slot, because they will never communicate.

In fact, the minimum condition for convergecast traffic is that for every node in hop number h, there must be at least one node in hop number $h-1$ that can receive its signal. In this way, every node in hop number h can send its signal to a node in hop number $h-1$. However, if there is only one parent alternative for a set of nodes, and if parent node fails, some nodes can not communicate. This is why having at least one parent alternative is called as minimum condition.

Sensor nodes are prone to failures and sensor network must be fault tolerant against failures. Minimum condition for convergecast traffic is not reliable under single point of failure. If there is only one node that can receive the signal of a certain node and if the receiving node goes off, the sender can not send its message. There should be more than one node that can receive the message of the sender. The solution for a reliable convergecast traffic network lies on between traditional network condition (no node within two hop radius can get the same slot) and minimum condition (having at least one parent alternative). The nodes in hop number *h* should get the slots so that, a certain number of nodes in hop number *h-1* can receive all the signals from *h* hop numbered nodes clearly. The number of *h* hop numbered nodes that can receive the signal of *h-1* hop numbered nodes is defined as fault tolerance parameter, *ftp*. A node can find at least *ftp* alternative parent nodes to send its message. This model increases the number of receiving nodes in hop number *h-1* for the senders in hop number *h* and it results in a more fault tolerant network. Figure 4.3 shows the situation with a certain example.

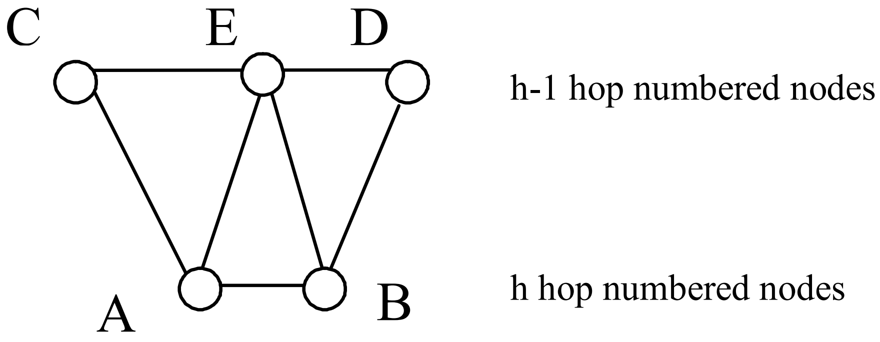


Figure 4.3. Sample slot assignment.

In Figure 4.3, node A and B are with hop number *h*, node C, D and E are with hop number *h-1*. If there is a connection between two nodes, they can hear each other. In traditional slot assignment algorithms, A and B can not have the same slot. However, the minimum condition of convergecast traffic is satisfied. According to the minimum condition, there must be at least one node in *h-1* hop number that can receive the signals of the nodes in *h* hop number. If A and B get the same slot, A can send its data to C and B can send its data to D. However, reliability may be a problem in this case. For example, if C fails, A can not send its data or if D fails, B can not send its data. According to ft_DTSM signal exchange structure, slots must be distributed so that for each *h* hop numbered node, there must be more than fault tolerance parameter nodes with *h-1* hop number that can receive the signals of the nodes in *h*. In Figure 4.3, if A and B get the same slot, E can not receive the signals clearly. ft_DTSM does not give the same slot to A and B. In this way, if C or D fails, E can still receive the signals of A or B.

Although ft_DTSM is more constrained than the minimum condition of convergecast traffic, it is more relaxed than FPRP. In Figure 4.3, even if E does not exist, A and B can not get the same slot in FPRP, because of the connectivity between A and B. However, ft_DTSM can give the same slot to A and B, because ft_DTSM checks only the collision between *h-1* hop numbered nodes and *h* numbered nodes.

According to this new situation, signal exchange is designed based on four tiny slots. These are request slot, collision report slot, confirmation slot and acknowledgement slot. The structure of the signal exchange is as follows:

The first slot of a reservation cycle is request slot. In this slot, every node that receives more than *ftp* signal in the advertisement slots and can receive at least one clear advertisement signal requests a slot with a certain probability. Let us assume that the hop number of the node is *h*. For a valid request, it sends a signal in request slot. Only the nodes with hop number *h-1* may suffer from the collision of the requests. The nodes that are with hop number *h-1* listen to request slot. If these nodes receive a jammed signal in this slot, it means there is a collision. The nodes that detect a collision in the request slot send a signal in the second slot which is collision report slot. If the node can get a clear signal in the request slot, it does not send any signal. The nodes in the hop number *h* listen to the second slot. If a node has sent a request and if it does not receive any collision report in the second slot, it can get a valid slot. In the third slot, which is confirmation slot, the node can get a valid slot sends a signal. The other nodes with hop number *h* and *h-1* listen to the confirmation slot. If a node with hop number *h-1* receives a signal in the third slot, it means there is a node that gets the current slot. If a node with hop number *h* receives a signal in the third slot, it can not give the current slot to another node. In this case, all the *h* hop numbered neighbors of *h-1* hop numbered nodes that receive a confirmation signal in the third slot send acknowledgment in the fourth slot. The receivers of the acknowledgment signal do not content for the current slot anymore. The diagram of all slot assignment signal exchanges is illustrated in Figure 4.4.

In this scenario, A and B can hear each other. C can communicate with only A. Hop number of A and B is *h*. Hop number of C is *h-1*.

At the end of this signal exchange some nodes can get valid slots. The nodes that can not have a valid slot try to get one in the next reservation cycles. Reservation cycles are repeated for a certain number of times in a reservation slot.

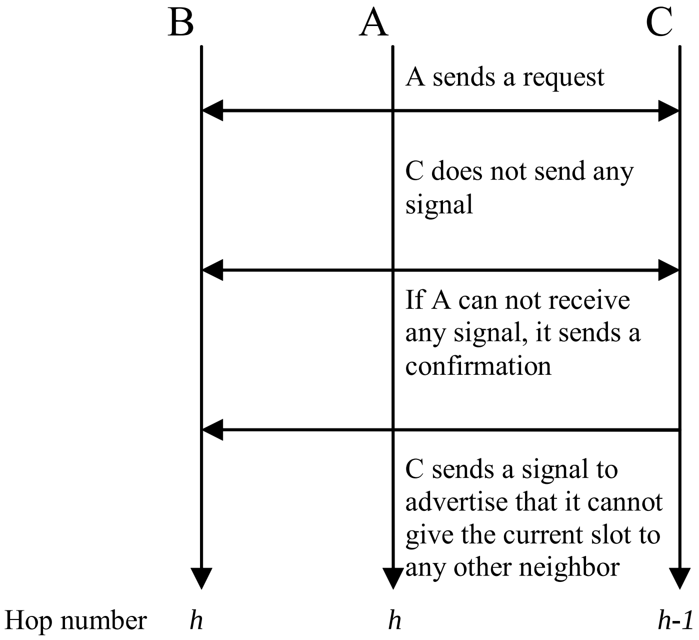


Figure 4.4. A successful signal exchange scenario.

The following theorem proves the correctness of ft_DTSM signaling structure.

Theorem : ft_DTSM signaling structure ensures that no two nodes with hop number $h+1$ can take the same slot number, if both of them can be received by a node with hop number h.

Proof : Let us assume that A and B are the nodes with hop number $h+1$, and C is the node with hop number h and C can receive signals from A and B. According to ft_DTSM signal structure, if A and B want to get a slot, they transmit signal in the first tiny slot, in request slot. C listens to the request slot. C can receive the signal from A and B, so it receives a jammed signal in the request slot. In the second slot, collision report slot, C sends a signal. In this way, it reports the existence of the collision. The nodes that request slot, like A or B, listen to the second slot. Neither A nor B receive the signal of C in the second slot. In this case, A and B do not send any signal in the confirmation slot and do not get the slot. It shows that two nodes with hop number $h+1$ can not get the same slot number, if there is a h hop numbered node that can receive A and B.

Signal exchange mechanism of ft_DTSM can provide more candidate parents for a child node by the help of its fault tolerance parameter. However, there must be a mechanism to maintain the network connectivity, when a sensor node fails. If a sensor node goes off and if it has a certain number of children node, children nodes can not send their data. In this case, each sensor node should be able to sense the functionality of its parent. Heartbeat signal can be a solution of this problem. Each node sends a heartbeat signal periodically. The sensor node listens to the other time slots to detect the functionality of its neighbors. If its parent is not alive, it initiates a two way handshaking to establish a new connection with another parent candidate.

4.2.4. Updating Slot Request Probability

At the beginning of signal exchange, every node can send a request signal with a certain probability. This probability is not constant. It is calculated as $1/N_c$, where N_c is the number of contender nodes in one hop neighborhood. N_c is updated at the end of each reservation cycle. N_c must be forecasted as realistic as possible. If N_c is forecasted larger than it is, contention probability will be lower than it should be and slot assignment algorithm may take longer than it is needed. If N_c is forecasted smaller than it is, contention probability becomes larger than it should be and collisions increases. N_c should be updated according to the result of reservation cycle. If a neighbor node achieves to get a valid slot, number of contenders decreases. If there is a collision, N_c must be increased to decrease contention probability. If nothing happens, in other words, if reservation cycle is idle, N_c should be decreased to increase contention probability.

ft_DTSM slot request probability update strategy is similar to FPRP [88]. FPRP is also adopted from Rivest’s pseudo-Baynesian Broadcasting Algorithm [105], which is designed for distributed single hop ALOHA broadcast network. Slot request probability update algorithm of ft_DTSM is presented in Figure 4.5.

According to ft_DTSM strategy, if a node can not receive or send any signal in a reservation cycle, it is idle. In idle state, N_c is decreased by one. If a node sends a request in the first slot and if it gets a collision report in the second slot, it is a collision. For a collision situation, N_c is incremented by $(e-2)^{-1}$.

```
At the beginning of each reservation slot $N_c = N_b$, $N_b = 0$.
(for the first reservation slot $N_c$ =the number of slots that will be distributed in one reservation frame)

For each reservation cycle, contention probability=1/ $N_c$.
If the state is
 Idle:                    $N_c = N_c - 1$, if $N_c >= 1$.
 Collision:               $N_c = N_c + (e-2)^{-1}$
 Success one hop away:    (node does not contend in this reservation cycle)
$N_c = N_c - 1$, if $N_c >= 1$.
                          $N_b = N_b + R_1 * N_c$.
                          $N_c = (1 - R_1) * N_c$.
Success two hops away:    (node does not contend in this reservation cycle)
$N_c = N_c - 1$, if $N_c >= 1$.
                          $N_b = N_b + R_2 * N_c$.
                          $N_c = (1 - R_2) * N_c$.
```

Figure 4.5. Slot request update procedure pseudocode.

If a node sends a request and does not receive a collision report in the second slot, it is a success for itself. It gets a valid slot and it does not contend anymore. If a node that does not send a request and receive a confirmation, it means there is a success one hop away. In addition to N_c, a new value must be calculated to update N_c for success state. This new value, N_b, represents the number of the nodes that has no valid slot and can not contend due to a success within one hop. The assumption is that if there is a success one hop away, a portion of its neighbors which is modeled as R_1 can not contend. After a success one hop away, N_b must be incremented R_1 times N_c, and N_c must be recalculated as $1-R_1$ times N_c. If a node receives a signal in the fourth slot, it means there is a success for a node which is not directly a neighbor but the success is in the neighbor of a node in hop number $h-1$. In other words, success is two hops away. In this case, N_c and N_b must be updated as in the one hop away success case. However, in this case R_2 value must be used.

4.2.5. Handling Delay Problems

One of the most important design issues for wireless sensor networks is delay. If application is delay sensitive, like military monitoring or surveillance as in [71], data latency can be very important. In a military monitoring system, the existence of enemy should be reported as soon as possible. Reducing delay is possible by the help of assigning time slots carefully. The rule is that smaller hop numbered nodes should get higher slot numbers. In order to realize rescheduling, time frame is divided into *u* sub time frames. If the whole time frame has s slots, a sub time frame has *s/u* slots. The slot number assigned to a node with hop number *h*, must be in *(u-((h-1) mod u))*th sub time frame. In this way, the slot numbers of consecutive hop numbered nodes belong to consecutive sub time frames. An example helps to understand clearer. Let us assume that the nodes in Figure 4.6 are one hop away from its consecutives.

In this particular network, time frame has 30 time slots and there are 3 sub slots. In this case, the first sub slot is from 21 to 29th slots, the second is from 11 to 20 th slots, and the third one is from 2 to 10 th. The first slot is reserved for the sink. Figure 4.6 (a) is an example of a slot assignment. Figure 4.6 (b) is a slot assignment based on ft_DTSM. Delay in Figure 4.6 (a) is 70 and delay in Figure 4.6 (b) is only 21.

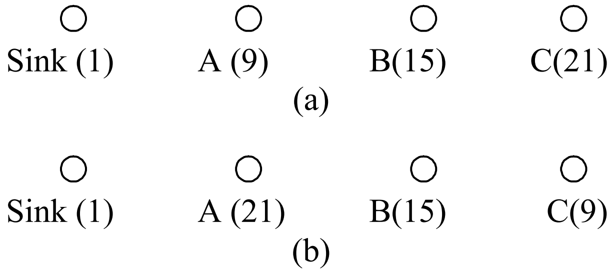


Figure 4.6. Sample network and time slots (a) A regular slot assignment. (b) ft_DTSM

The pseudo code of ft_DTSM is presented in Figure 4.7.

```
// pseudocode of ft_DTSM
s is the number of slots, u is the number of sub frames, ftp is fault tolerance parameter
MyHop=0

While AdvSignals<=ftp //receives more than ftp valid adv. signals
  Receive(AdvSignals)
  MyHop=MyHop+1
Wend

// This part is for contention (for the nodes in h)
CurrentSlot=0
While CurrentSlot<MaxSlot // loop for reservation slots
  CurrentSlot= CurrentSlot+1
  For i= 1 to M[CurrentSlot]  // reservation slot contains M reservation cycle
     for ContentionProbability, send a request signal  // request slot
     otherwise, listen to request slot
     Receive(CollisionReport) // collision report
     If it does not receive any collision report and if it has sent a request then
      Send (ConfirmationSignal)
      MySlot = CurrentSlot + (u-((h-1) mod u)-1)*(s/u)  // Current slot in the current sub frame
     Else
       Receive (ConfirmationSignal)
       Receive (AcknowledgmentSignal)
     End if
     Update (ContentionProbability)
  Next i
Wend


// This part is for approval and acknowledgement (for the nodes in h-1)
Send (AdvSignal)
CurrentSlot=0
While CurrentSlot<MaxSlot // loop for reservation slots
  CurrentSlot= CurrentSlot+1
  For i= 1 to M[CurrentSlot]  // reservation slot contains M reservation cycle
     Receive (RequestSignal) // request slot
     If it receives a collision then // collision report
       Send (CollisionReport)
     End if
     Receive (ConfirmationSignal)
     If ConfirmationSignal <> 0 then
       Send(AcknowledgmentSignal)
     End if
  Next I
Wend
```

Figure 4.7. ft_DTSM pseudocode

4.3. Data Indicator Slot Mechanism (DISM)

DISM is another mechanism which is used to decrease energy consumption [106]. In a tree based TDMA WSN, parents must listen to their children to check whether children have data or not. If children have no data, it causes overhearing and energy waste. According to DISM, one special slot is reserved for data indication. When a sensor node senses an event or when it has data to send, it sends a signal in the data indicator slot (DIS). Every node listens to its data indicator slot. If a node does not receive a signal, it does not check its children. If a node receives a signal at the DIS, it means there is at least one neighbor node that has data. Sensor node that gets a signal in the data indicator slot listens to the slots of its children. Although it can not eliminate the entire overhearing problem, it reduces most of the redundant checks.

When a node needs to check its children for data existence, it does not listen to the entire slot. It only listens to the beginning of the slot. If there is data at the beginning, it continues to listen. If it has no data, it stops to listen as in [65]. The preamble of the slot that the node listens to is called as preamble interval.

5. PERFORMANCE RESULTS

MILMON has three newly proposed mechanisms. These are SyncHRT which uses a high range transmitter to synchronize WSN, ft_DTSM which distributes time slots to reduce energy consumption and delay, DISM which also helps to minimize energy consumption. In this chapter, firstly, WSN parameters are defined. All the analysis in this section is based on these parameters. After WSN parameters, performance results of newly proposed mechanisms are presented. At the end of the section, the performance of MILMON is compared with existing WSN systems.

5.1. Simulation and Analysis Basic Parameters

Simulation parameters are presented in Table 5.1.

Table 5.1 Simulation parameters

Parameter	Value
Shape of the sensor network area	Circle
Location of the sink	Center
Diameter of sensor network	4000 m
Bit rate, B	115,2 Kbps
Receive energy (one bit), R_r	0.12 µJoule
Transmit energy (one bit), R_s	0.22 µJoule
Transmission range	150 m
Sensing range	100 m
Data Package	20 bytes
Topology	Tree

All power consumption parameters are compatible with Berkeley’s Motes [52]. The locations of the nodes are uniformly distributed over the simulation area. Topology of the network is tree. The topology construction can be realized by the request of the child.

WSN traffic load may vary quickly. When traffic load is heavy, the load may not be balanced. In order to balance the data traffic, it is possible to choose the parent dynamically so that the nodes send the data to the parents with less traffic load. The parent selection can be formulated with the loads of the candidate parents and the load of the sender node. For the simplicity, in this simulation environment, each node connects to the closest candidate parent produced by ft_DTSM.

5.2. SyncHRT Performance Results

Performance of SyncHRT is investigated in three domains: precision, energy consumption, and run time. A simulator is implemented for performance comparison.

5.2.1. Precision

The precision from the sink to the directly synchronized nodes is equal to the precision of FTSP for one hop plus propagation delay. Sensor nodes are deployed so densely that prorogation delay can be neglected. Precision of the nodes that can not receive the sink's beacon directly is calculated with the multihop precision of FTSP. The precision of the proposed system is compared with original FTSP, RBS and TPSN according to the parameters presented in Table 5.1. The comparison results are presented in Table 5.2.

Table 5.2. Precision performances of different time synchronization mechanisms

Time Synchronization Method	Avg. error (μs)	Max. error (μs)
RBS	112.70	360.19
TPSN	22.66	180.10
FTSP	8.98	40.28
SyncHRT	3.48	8.88

FTSP produces much higher precision than RBS and TPSN and the sink that covers the entire network reduces the maximum synchronization error approximately 4 times with respect to FTSP. The precision performance difference between a system with a high range transmitter and FTSP is approximately 2.5 times in average synchronization error.

The precision of the system highly depends on the range of the transmitter. The additive average and maximum error for each hop is 0.5 µs and 2.30 µs respectively in FTSP. However, if there is a high range transmitter, only propagation delay error is added. This is why the system with high range transmission produces higher precision.

Figure 5.1 shows the average and maximum error with different transmission range of the sink. When the transmission range is about 100 m, synchronization error of system equals to the synchronization error of FTSP. When the range becomes higher, lower synchronization errors are produced. The relationship between the range and the synchronization error is linear.

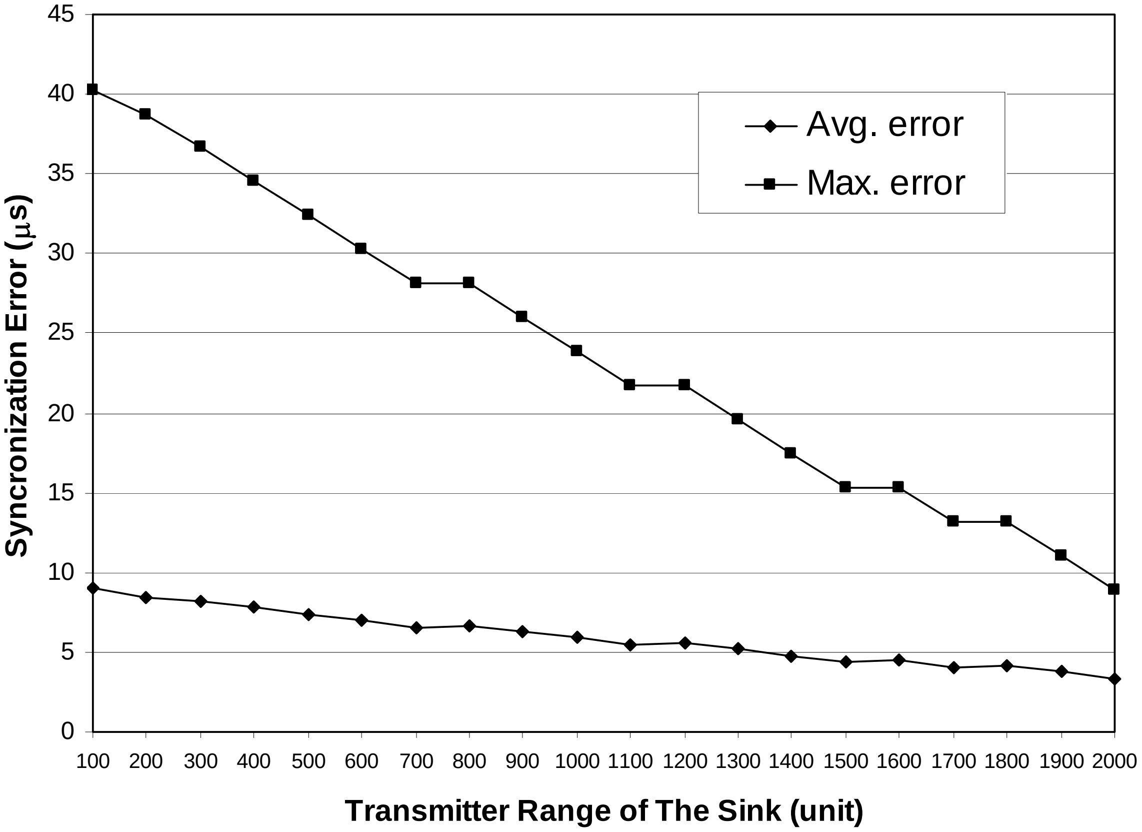


Figure 5.1. Synchronization errors for different range of transmissions.

When the sink's transmitter covers the entire network, average synchronization error becomes 3.48 μs which is approximately 2.5 times better, and the maximum synchronization error becomes 8.88 μs which is approximately 4 times better than FTSP.

5.2.2. Energy Consumption

WSN time synchronization algorithms should be energy aware like other WSN applications. In all cases, FTSP consumes less energy than RBS and TPSN [69]. Therefore, energy consumption of the system is compared only with FTSP. In FTSP, every sensor node receives and transmits only one synchronization packet. If synchronization period is 60 seconds and one synchronization packet is 8 bytes, FTSP consumes 0.36 μwatt. If the sink has higher transmission range, energy consumption decreases. Average energy consumption of the system with high range transmitter is presented in Figure 5.2 for different transmission ranges.

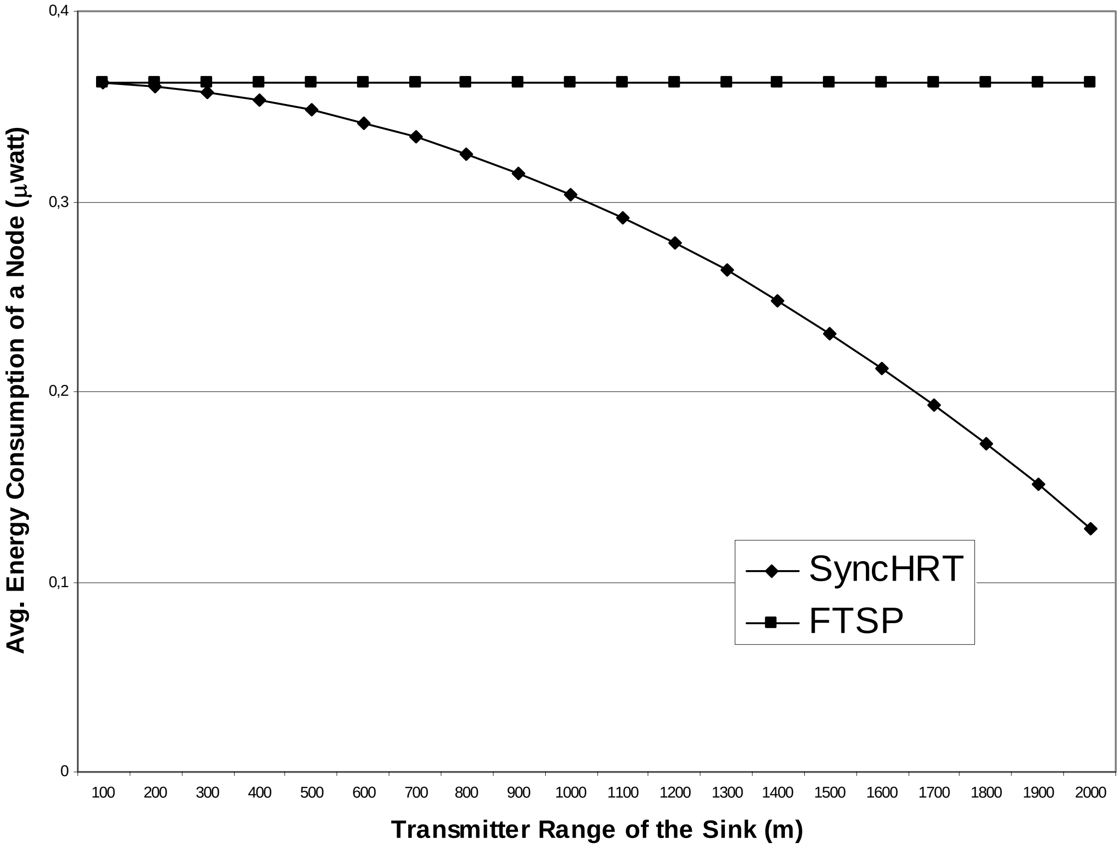


Figure 5.2. Energy consumption for different transmission ranges.

If the sink has a high range transmitter, the nodes within the range of the sink do not need to send packet. Only the nodes that have at least one unsynchronized neighbor node send synchronization packet. It decreases energy consumption. When the range is higher, it can save more energy. When the sink can cover the entire network with high range transmitter, energy consumption decreases to 0.13 μwatt. It means approximately 3 times less energy consumption than FTSP.

5.2.3. Run Time

Time synchronization run time of the system with high range transmitter is compared with FTSP, TPSN and RBS. An analytic model has been developed for run time of FTSP in [69]. The same model is used to calculate run time of FTSP and the system with high range transmitter.

Run times of RBS, TPSN, FTSP and the system with high range transmitter is presented in Figure 5.3.

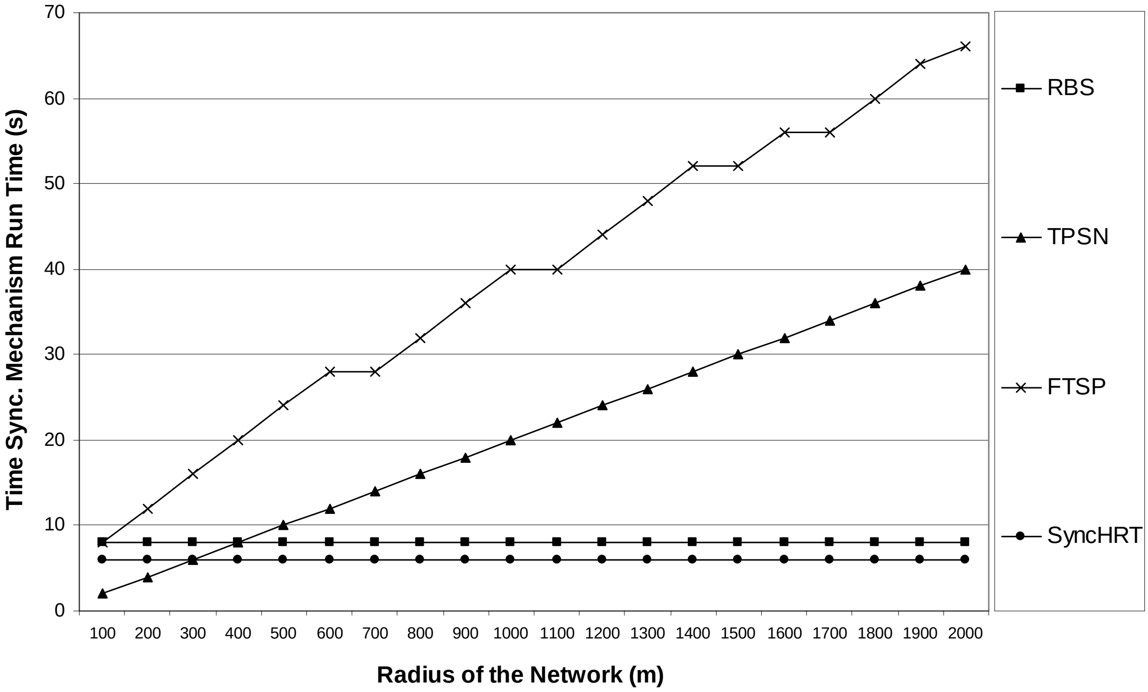


Figure 5.3. Run times of time synchronization methods for different network sizes.

Run times of TPSN and FTSP are directly related with the maximum hop number of the network. Therefore, run time of TPSN and FTSP increases linearly, as the network size increases. RBS is a completely distributed algorithm and the run time of RBS is constant for all network sizes. If the transmitter of the sink can cover the entire network, synchronization run time is also constant. Although TPSN and FTSP can achieve shorter run times for very small networks, the system with high range transmitter can run much faster for larger networks. Run time of RBS is close to the system with high range transmitter. While TPSN has the shortest run time for the networks with small diameter, the sink with high range transmitter achieve to run in much shorter time for all other cases. Run time ratio difference between FTSP and the system with high range transmitter can be up to 11 times. This difference is more than 6 times with TPSN.

Run time performance of the system with high range transmitter is directly related with the range of the transmitter. In Figure 5.4, run times are presented for different transmitter ranges.

Figure 5.4. Run times for different transmission ranges.

If the sink has higher transmission range, hop number of the network also decreases and it directly decreases run time of time synchronization mechanism. When transmission range of the sink is low, run time is around 64 seconds. While transmission range is higher, run time decreases linearly. When the sink can cover the entire network, run time is only 6 seconds. Using a high range transmitter can decrease time synchronization run time approximately 11 times.

5.3. ft_DTSM Performance Results

Performance of ft_DTSM is discussed according to delay, energy consumption, and running time. A simulator is implemented to compare ft_DTSM with DTSM, FPRP, FLAMA and DRAND. The system parameter presented in Table 1 is used for performance analysis. Another parameter for simulation is the number of reservation frames for ft_DTSM and the number of reservation cycles in each reservation frame. Number of reservation frames and reservation cycles should be set so that slot assignment algorithm can assign valid slot with high probability and it should minimize energy consumption and run time. In order to find optimum parameters, the existence of a central coordinator is assumed for FPRP and ft_DTSM. The pseudocode of central coordinator is presented in Figure 5.5.

```
While (there is at least one node that request and can not get a valid slot)
        If there is at least one node that can get the current
                do_ ft_DTSM or do_FPRP
        else
                increase reservation cycle and current slot number by 1
        end if
Wend
```

Figure 5.5. Central coordinator pseudocode.

If a node can get the current slot, reservation cycles are continued to repeat and if there is a node that can not get valid slot in the current slot, reservation frames are continued to repeat. In this way, minimum number of slots is assigned to the nodes. Average number of reservation cycles and reservation frames are calculated after 20 runs with central coordinator and the calculated numbers are used as parameters. The most important factor that affects these parameters is node density. Number of reservation cycles and reservation frames is set for each node density. Simulation results have shown that the parameters that are calculated with this methodology can assign valid slots with more than 99.5 per cent probability.

5.3.1. Delay

Delay is one of the most important problems for sensor networks. Especially, delay sensitive applications like military monitoring, may suffer latency. ft_DTSM has a special mechanism for handling delay problem. Delay performance of ft_DTSM is investigated in three ways. ft_DTSM has a sub frame structure to decrease delay. Firstly, the effect of sub frame structure is investigated. Secondly, the effect of fault tolerance parameter on the delay performance is presented. Thirdly, ft_DTSM is compared with existing WSNs.

In the simulation, one data transfer time frame is assumed as one second. If a packet is composed of 20 bytes, and a node can send one packet in one data slot, one data slot takes about 1.5 ms, including synchronization and guard bits.

The first experiment is set up just to show the effect of sub frame structure. So, fault tolerance parameter is accepted as 1 for all cases. In our simulation, sub time frames are 5, 10, 20 and 30. Delay is related with the distance between event and the sink. 100 events are generated for different distances and simulation is repeated for 20 times. Figure 5.6 shows average delay of ft_DTSM with different number of sub frames and FPRP.

ft_DTSM is successful to decrease delay with its sub frame structure. Especially, when the distance is longer and sub frame number is high enough, delay can be reduced. Although sub frame number affects delay performance, it is not always directly proportional with sub frame number. The delay performance of ft_DTSM follows a step pattern related with average hop number between the event and the sink. For example, average hop number for 1000 m. is 10, and delay of DTSM-20 and DTSM-10 is approximately the same. After 1000 m., while delay of DTSM-10 increases, delay of DTSM-20 still stays almost constant. The same structure can be found for DTSM-5.

Step pattern is closely related with average hop number, distance, and the number of sub frames. Average hop number of the distance 1000 m. is 10 and delay of DTSM-10, DTSM-20 and DTSM-30 is very close for this distance. If the distance is higher than 1000 m., average hop number exceeds 10 and delay of DTSM-10 starts to increase.

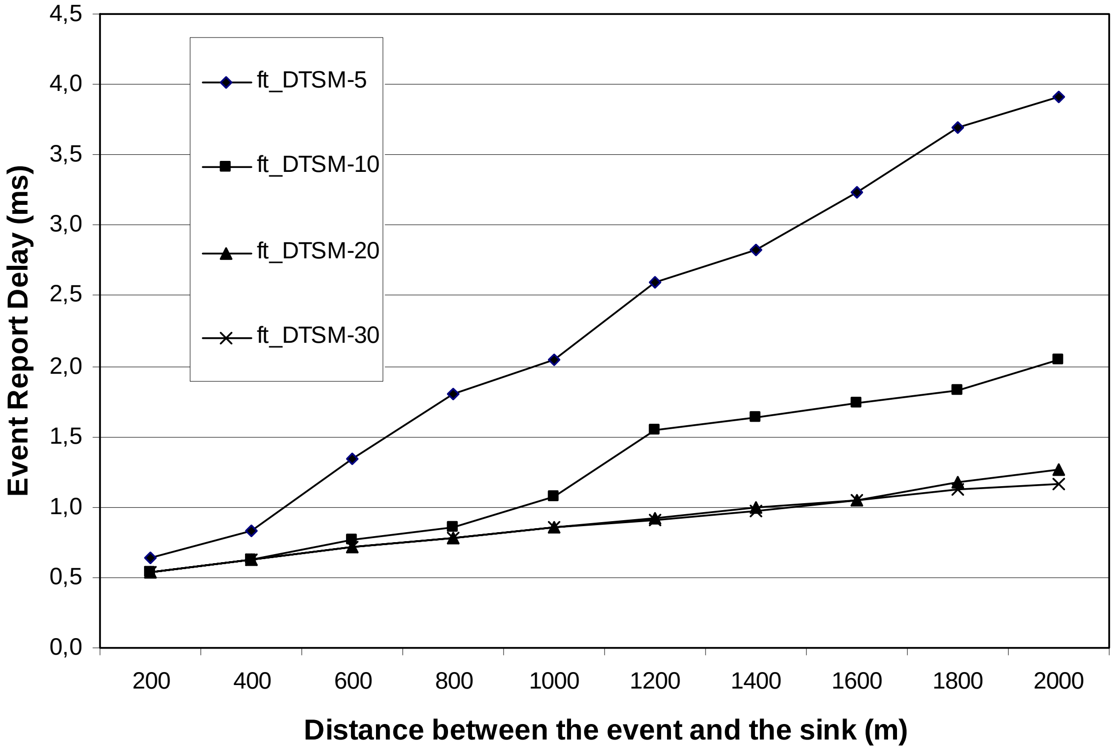


Figure 5.6. Delay performance of ft_DTSM with different sub time frame numbers.

While the delay of DTSM-10 increases, the delay of DTSM-20 and DTSM-30 stays approximately constant. It shows that sub frame number must be chosen according to maximum hop number of the sensor network. If sub time frame number is lower than maximum hop number, delay increases. If sub time frame number exceeds the maximum hop number, delay is not reduced any more, it stays the same.

Another factor that affects delay performance of ft_DTSM is fault tolerance parameter. When fault tolerance parameter increases, maximum hop number of the network also increases, and when hop number is higher, delay also increases. Delay performances of ft_DTSM with different fault tolerance parameters and DTSM which has no fault tolerance mechanism are shown in Figure 5.7. In Figure 5.7, sub slot number is assigned as 20 to minimize delay for all cases.

DTSM which has no fault tolerance mechanism achieve to realize the lowest delay. ft_DTSM produce higher delay for all fault tolerance parameters. When fault tolerance parameter in ft_DTSM increases, delay also increases. Delay increase of ft_DTSM is sourced from the increase in maximum hop numbers with higher fault tolerance parameter. When DTSM is used, delay is around 1600 ms. Delay increases to 1800 ms for *ftp* 1. However, if fault tolerance parameter is 4, delay is approximately 2500 ms. Delay difference between DTSM and ft_DTSM with 4 fault tolerance parameter is about 1.5 times. Fortunately, the resulting delay of ft_DTSM is still acceptable for many applications.

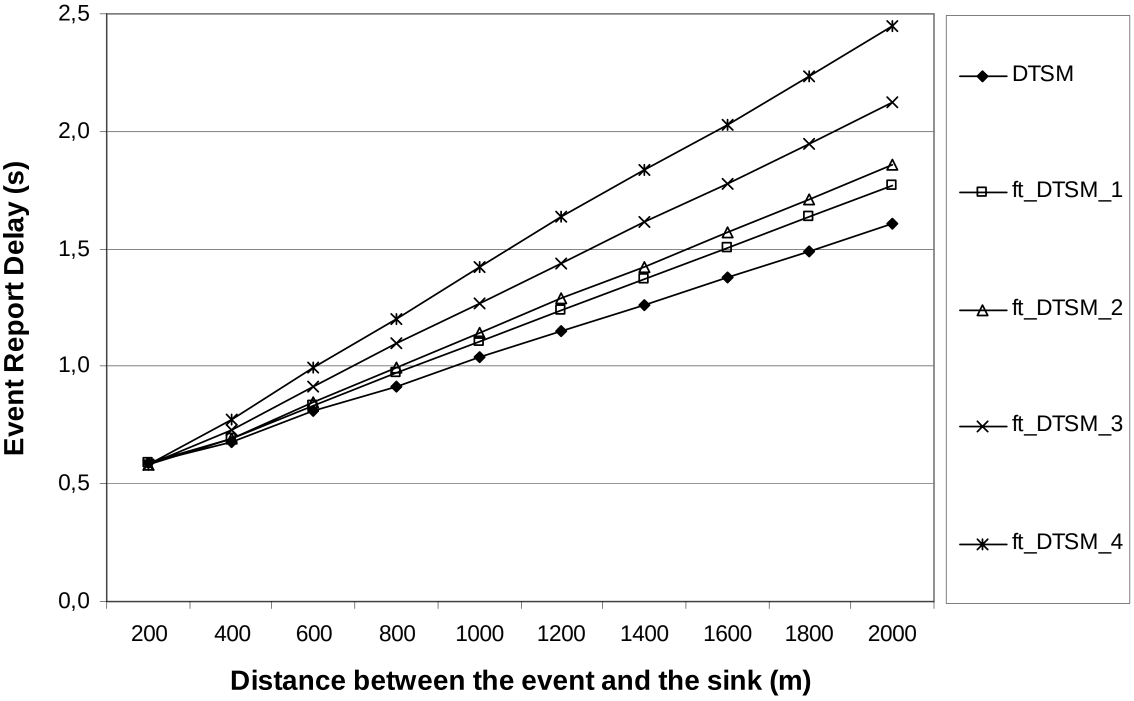


Figure 5.7. The delay performances of different WSN systems.

Delay performance comparisons of ft_DTSM and existing WSNs are investigated in Section 5.4.2, MILMON delay performance results.

5.3.2. Energy Consumption

Sensor nodes have limited energy and when power goes off, sensor node can not function. Energy is one of the most critical resources for sensor networks. Slot assignment mechanism of a sensor network must be energy saver like any other algorithm used in sensor networks.

FLAMA is based on two different kinds of periods. In time slot periods, data traffic is realized. In contention based period, FLAMA distributes time slots. Slot assignment algorithm of FLAMA is run in random access based period and all nodes have to listen or transmit in this period. It takes considerable energy. DRAND also uses 802.11 like signal exchange mechanism and it needs very large amount of signaling. Although, FPRP does not use any additional MAC layer for slot assignment, it is not designed for WSNs. ft_DTSM is an energy saving distributed slot assignment algorithm for WSNs.

One of the most important parameters for energy consumption for ft_DTSM is node density. The simulation results for different node densities are presented in Figure 5.8 to compare energy consumption of ft_DTSM with different fault tolerance parameter, DTSM, FPRP, DRAND and FLAMA.

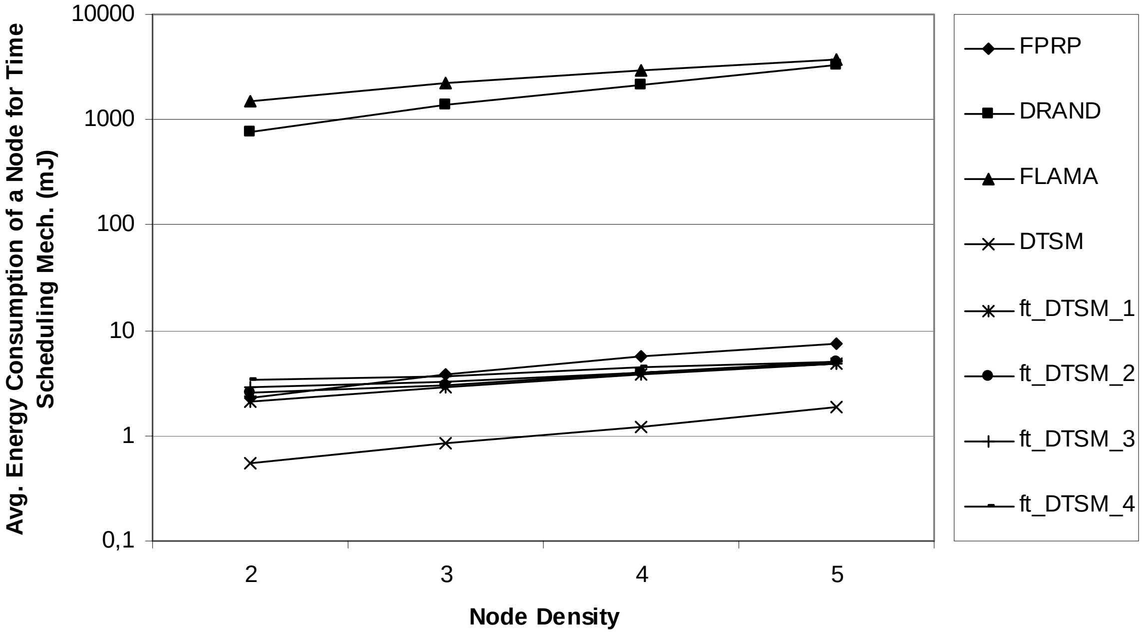


Figure 5.8. Energy consumption comparison of time slot assignment algorithms.

Energy consumptions of all slot assignment mechanisms increase with the increasing node density. While energy consumption increase structure of ft_DTSM, DTSM and DRAND is polynomial, energy consumption of FPRP and FLAMA increases linearly. It is clear that time slot assignment protocols that include contention period, like FLAMA or DRAND consume much more energy than slot assignment protocol based on tiny time slots, like FPRP, DTSM or ft_DTSM.

Fault tolerance mechanism also affects energy consumption. In ft_DTSM, a node can not start to contend for a slot, even if there are some neighbor nodes with valid slots. The number of neighbor nodes with valid slot must be higher than *ftp*. An ft_DTSM node has to wait until it can find more than *ftp* neighbor nodes with valid slots. In this case, maximum hop number of the resulting network also increases. Each reservation frame corresponds to a specific hop number. When maximum hop number increases, number of used reservation frames also increases. More reservation frame means more energy to consume.

ft_DTSM consumes more energy than DTSM, but they are very close. For all *ftp*, ft_DTSM consume less energy than FPRP and the other 802.11 based mechanisms.

5.3.3. Running time

ft_DTSM, DTSM, FPRP, FLAMA and DRAND are compared about their running times. FPRP is fully parallel and distributed algorithm. All the nodes can run FPRP algorithm at the same time. The reservation process for a given node only involves nodes within a two-hop radius, and is thus a local process. No coordination is necessary with more distant nodes. By keeping the reservation process localized (and running simultaneously over the entire network), the FPRP is insensitive to the network size. Its running time is constant. However, nodes run DTSM and ft_DTSM according to their hop numbers. It follows a wave pattern from the lowest hop number to maximum hop number. In the first reservation frame, the nodes which are one hop away from the sink allocate a certain set of slots. In the second reservation frame, the nodes which are two hop away from the sink allocate the slots and so on. In this case, running time of DTSM and ft_DTSM is fully dependent on the maximum hop number of the network.

Analytic models are developed to compare run time performances of different time slot assignment algorithms. Running time of FPRP is as follows:

$$R_{FPRP} = 5 * T_{se} *. T_{rc} \tag{5.1}$$

where, T_{se} is time for one signal exchange slot, and T_{rc} is total number of reservation cycles in reservation frame.

Running time of ft_DTSM is as follows:

$$R_{ft\_DTSM} = 4 * T_{se} * T_{rc} * N_{maxh} + T_{as} * N_{maxh} \tag{5.2}$$

where N_{maxh} is maximum hop number, and T_{as} is time for advertisement slots.

Running time of DTSM is as follows:

$$R_{DTSM} = 3 * T_{se} * T_{rc} * N_{maxh} + T_{one\_as} * N_{maxh} \tag{5.3}$$

where T_{one_as} is time for one advertisedment slot.

In Figure 5.9, average running time of DTSM, ft_DTSM with different fault tolerance parameter and FPRP are compared. It is assumed that every signal exchange slot has 5 bits and there are two guard bits between each signal exchange slots.

Running time performances are affected by node density. Higher node density implies longer run time. It is sourced from higher contention rate. While DTSM and ft_DTSM are hop number based slot assignment algorithm, FPRP has a fully distributed slot assignment mechanism. In FPRP, all the nodes try to get a valid signal at once.

However, in DTSM or ft_DTSM, higher hop numbered nodes waits for the lower hop numbered nodes to get a valid slot. In this case, FPRP always run faster than DTSM and ft_DTSM in all cases.

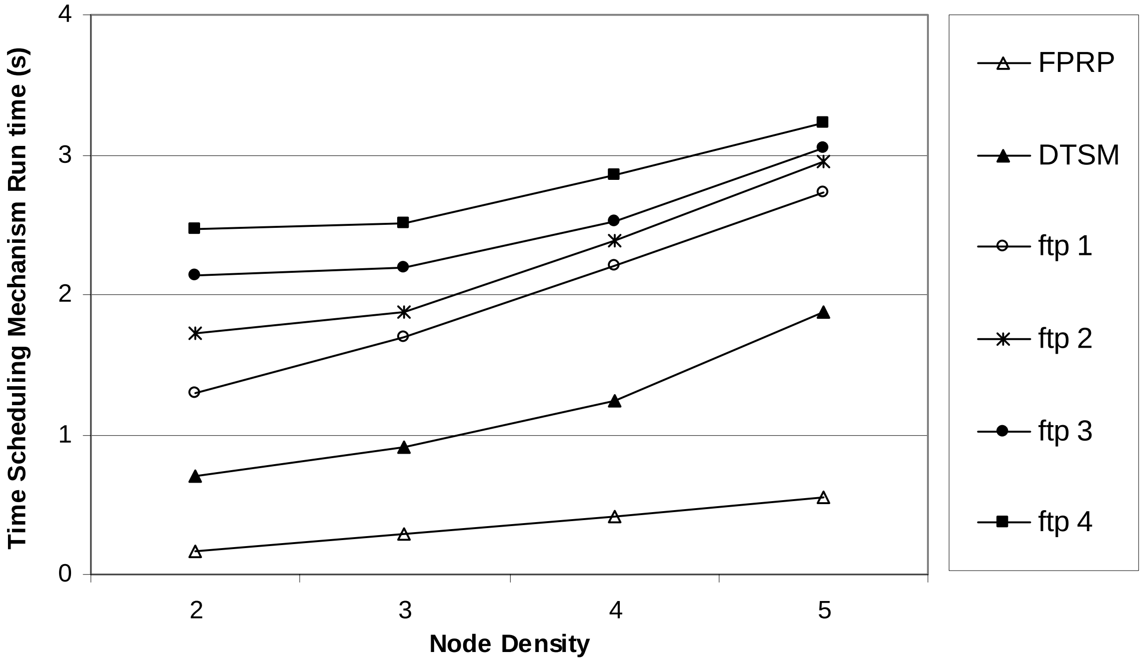


Figure 5.9. Running time of ft_DTSM, DTSM and FPRP for different network size.

While DTSM nodes wait for at least one neighbor nodes with valid slot, ft_DTSM nodes must wait for a certain number of neighbor nodes with a valid slot. According to R_{ft_DTSM} formula, maximum hop number of the resulting network contributes on total running time. This is why DTSM is always faster than ft_DTSM for all fault tolerance parameters. DTSM lies between FPRP and ft_DTSM. Run time differences between ft_DTSM with different fault tolerance parameters are clearer in lower node densities. In lower density networks, neighbor node numbers are also lower and finding an appropriate parent probability decreases. In this case, for a lower density network, maximum hop number differences between different fault tolerance parameters are higher. On the other hand, in higher densities, maximum hop numbers of different ft_DTSM with fault tolerance parameter are closer. In this case, run times also become closer. While FPRP, DTSM and ft_DTSM can run in the order of seconds, simulation implemented in [94] has shown that DRAND takes approximately 25 seconds when node density is 1. DRAND run time fits a quadratic curve with varying node densities. When node density is 5, DRAND takes approximately 240 seconds which is much longer than run time of FPRP and DTSM. FLAMA also takes much longer than FPRP or DTSM. It takes 55 s when node density is 1. Its run time increases linearly with the increasing node density. It shows that its run time for node density 5 is about 275 s.

5.4. MILMON Performance Results

MILMON is composed of SyncHRT and ft_DTSM. It also uses data indicator slot mechanism (DISM) [107]. Its main design objectives are energy consumption, delay and fault tolerance. In this section, performance of MILMON is compared with the existing WSN systems.

MILMON is designed for monitoring large areas against intruders. Default system parameters, presented in Table 5.1, are used for performance analysis. Table 5.3 shows additional system parameters for used for performance comparisons.

Table 5.3. Additional simulation parameters

Parameters	Values
Data package, S_d	20 bytes
Data time frame, T_w	1 second
DISM signal for sending, S_{dism}	5 bits
Preamble signal, S_{pr}	1 byte
Synchronization period	60 seconds
Synchronization packet	8 bytes
Sensor node density	0,02 nodes/m^2

Detection probability and sensor node density is closely related and Onur has proposed an analytical model to determine the required number of sensors to provide a required detection probability [108]. Sensor node density is calculated as 0,02 nodes/m^2 to provide detection probability higher than 99.9.

5.4.1. Energy Consumption Performance of MILMON

Energy consumption of newly developed monitoring system is compared with SyncWUF, DMAC, and FLAMA. SyncWUF is an optimized version of WiseMAC, which is one of the most energy saving MAC layer for wireless sensor networks, for low WSN traffic requirements. SyncWUF consumes up to 100 times less energy than S-MAC and T-MAC which are the most known WSN MAC layers [52]. DMAC is another WSN developed to reduce energy consumption and delay [54]. In DMAC, each node uses contention periods which are arranged to minimize delay. FLAMA, a TDMA based WSN architecture, relays its data in scheduled access slots. However, it distributes time slots in contention based periods.

Energy consumption of different WSNs is investigated with analytic model and simulation model.

Analytic model of new system's power consumption for one hop is as follows:

$$P_{ns} = 64R_r/60 + R_r/T_w + R_sS_{dism}/L \ + (R_rS_d + R_sS_d)/L + R_rN_pN_cS_{pr}/L \qquad (5.4)$$

where, N_p is the number of alternative parent for a node, N_c is number of children for a node, and L is the average number of relaying packets in one second, or in other words, it is the system load.

When a node sends data to its parent, the main energy consumption sources are:

1. Time synchronization,
2. Sending DISM signal,
3. Controlling DISM signal by the parent,
4. Sending and receiving data,
5. Overhearing data indicator slot.

If the sink can cover the entire network, every node needs to receive only one synchronization packet in a synchronization period. The first term of Equation (5.4) represents the energy consumption of SyncHRT, if the synchronization period is 60s and a synchronization packet has 8 bytes. While the entire DISM signal must be sent, to receive only one bit is enough to control the existence of DISM signal. The second and the third terms are used for controlling and sending DISM signal respectively. The fourth term is added for relaying data from one node to another. The last term is for overhearing nodes. If there are N_p nodes that listen to the same DISM signal, they have to check their children. The nodes that overhear a DISM signal listen to their children's slot for only preamble interval.

FLAMA is a TDMA based WSN. It does not have a mechanism like DISM and a FLAMA node controls its all children for all time frames. If it is assumed that preamble interval of FLAMA is equal to the preamble interval of the new system, energy consumption of FLAMA is as follows:

$$P_{FLAMA} = R_r N_c S_{pr} + (R_r S_d + R_s S_d)/L \qquad (5.5)$$

The first term of Equation (5.5) is for preamble interval and the second term is the energy consumed for relaying data from one node to another. It must be noticed that FLAMA has random access based periods and energy cost of these periods is not included in Equation (5.5).

DMAC receives and sends data in its data slots. It has to listen to the entire receive slot. It also needs synchronized nodes and it runs a time synchronization algorithm. For a fair comparison, DMAC is also assumed to use SyncHRT which consume less energy than the existing WSN time synchronization methods. DMAC energy consumption is as follows:

$$P_{DMAC} = 64R_r/60 + R_r S_d + R_s S_d/L \qquad (5.6)$$

The first term is for SyncHRT, the second is for receiving data slot and the last one is for data relay.

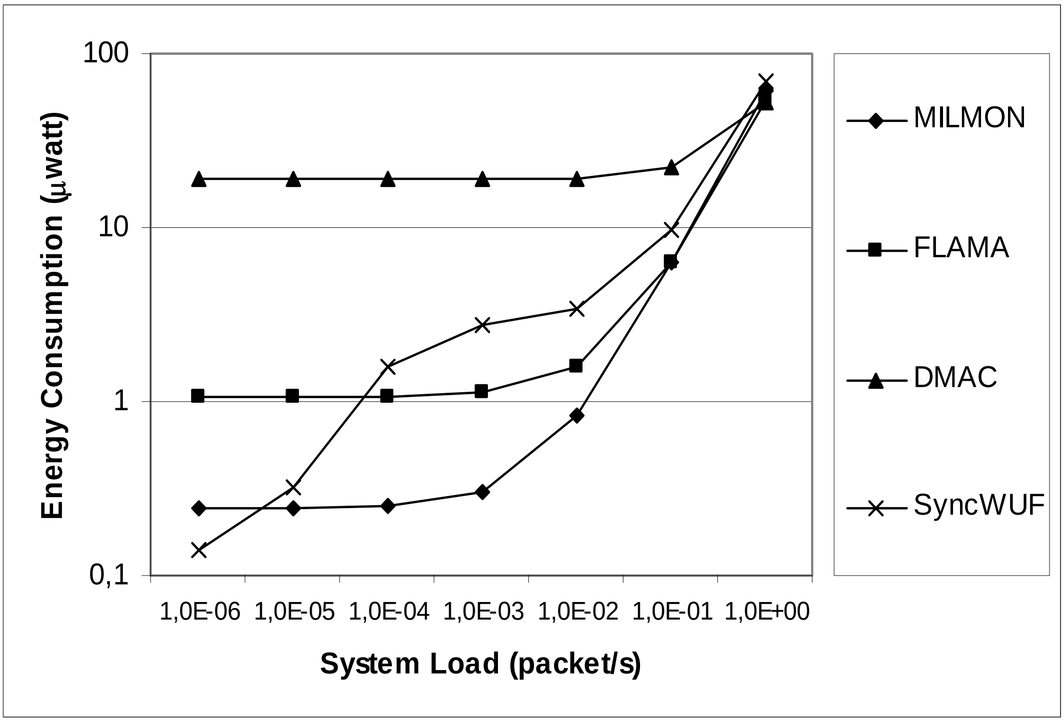
The model developed in [58] is used as the analytic model of SyncWUF.

Energy consumption performance is investigated for different number of children and different system loads. Figure 5.10 shows the energy consumption comparisons of SyncWUF, FLAMA, DMAC and MILMON.

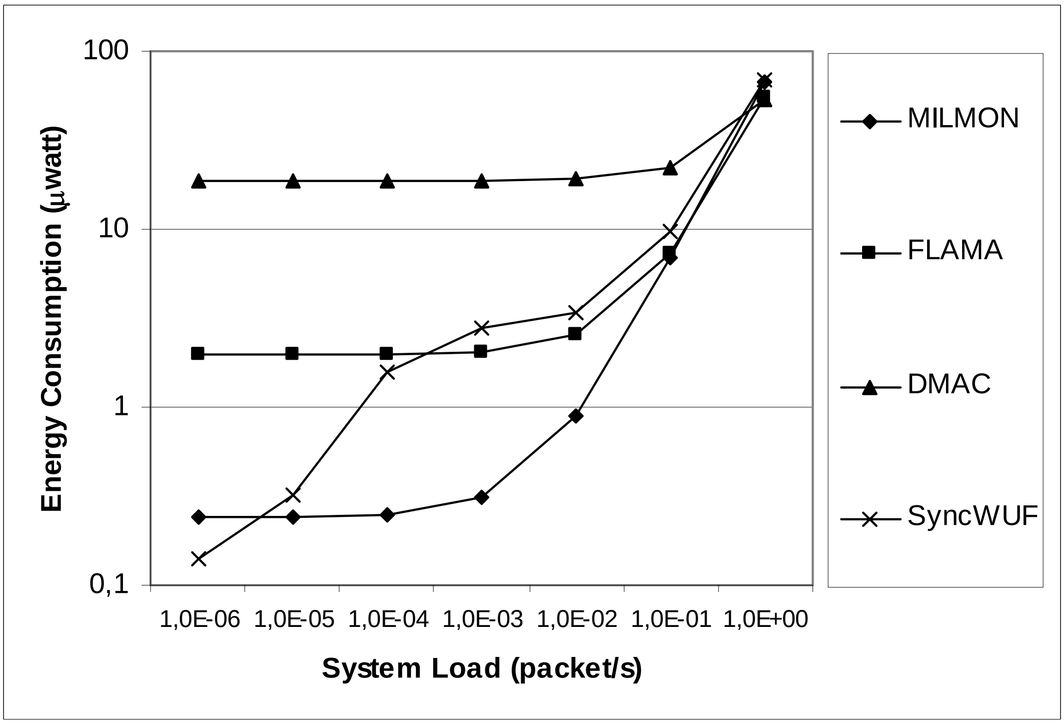
System load is sourced from the events sensed from the environment and some additional signal exchange like heartbeat signals for maintaining the network connectivity. If it is assumed that heartbeat signals take place in a couple of hours, the minimum expected system load is 10^{-4} packets/s. On the other hand, according to MILMON application scenario, the existence of an intruder or terrorist is unlikely. So the system load sourced from the intruders is not so heavy. In maximum case, it may be 10^{-2} packets/s. It should be kept in mind that system load definition is the data traffic of a single node; it is not the whole data traffic. In most of the cases, MILMON operates between 10^{-4} and 10^{-2} packets/s.

Energy consumption performances of DMAC and SyncWUF do not change with the number of children. Only FLAMA and our system are affected by the number of children. Energy consumption of FLAMA dramatically increases with the number of children, especially for low system load. However, the effect of the number of children is limited for FLAMA in higher system loads. On the contrary, while energy consumption of our system has a very small change for low system load, the change in higher system load is considerable. If the system load is 1, and if there is only one child, energy consumption is 60.39 μw. However, if the number of children is 4, it increases to 78.25 μw.

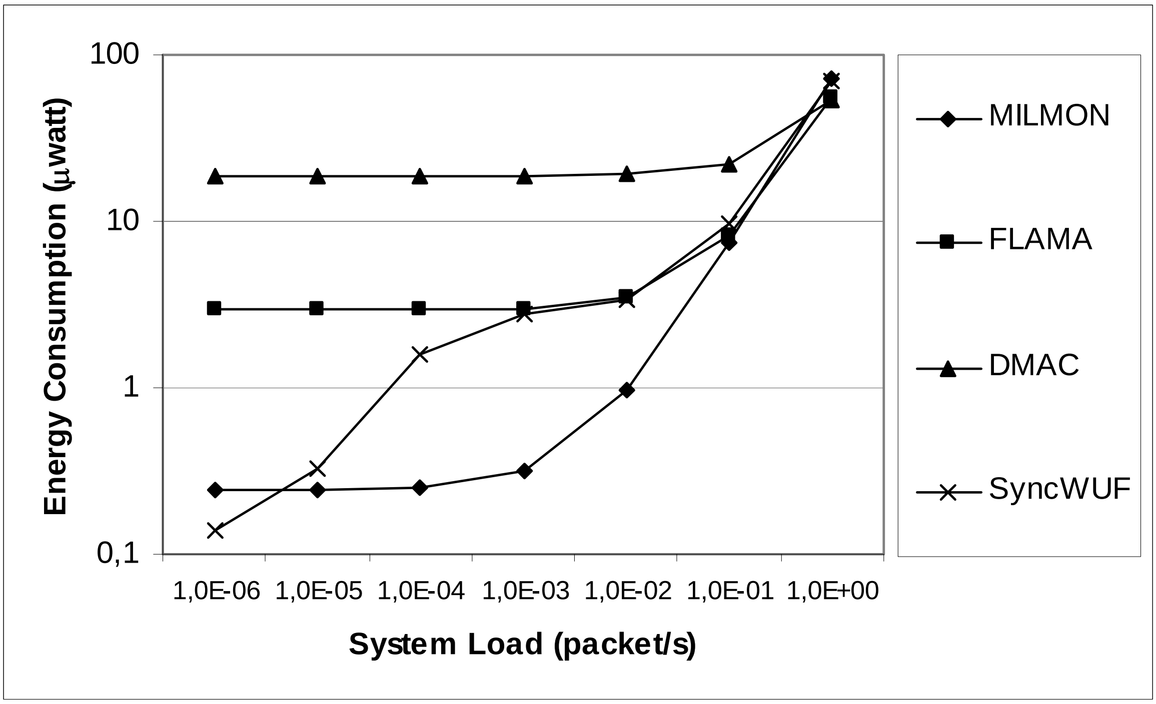
For any number of children, the newly proposed system consumes less energy than DMAC and FLAMA, for low system loads. Only SyncWUF can save more energy, if the system load is one event per one million seconds which is out of the MILMON operation range. Energy consumption difference between SyncWUF and the new system is realized as 1.8 times for very low system loads.



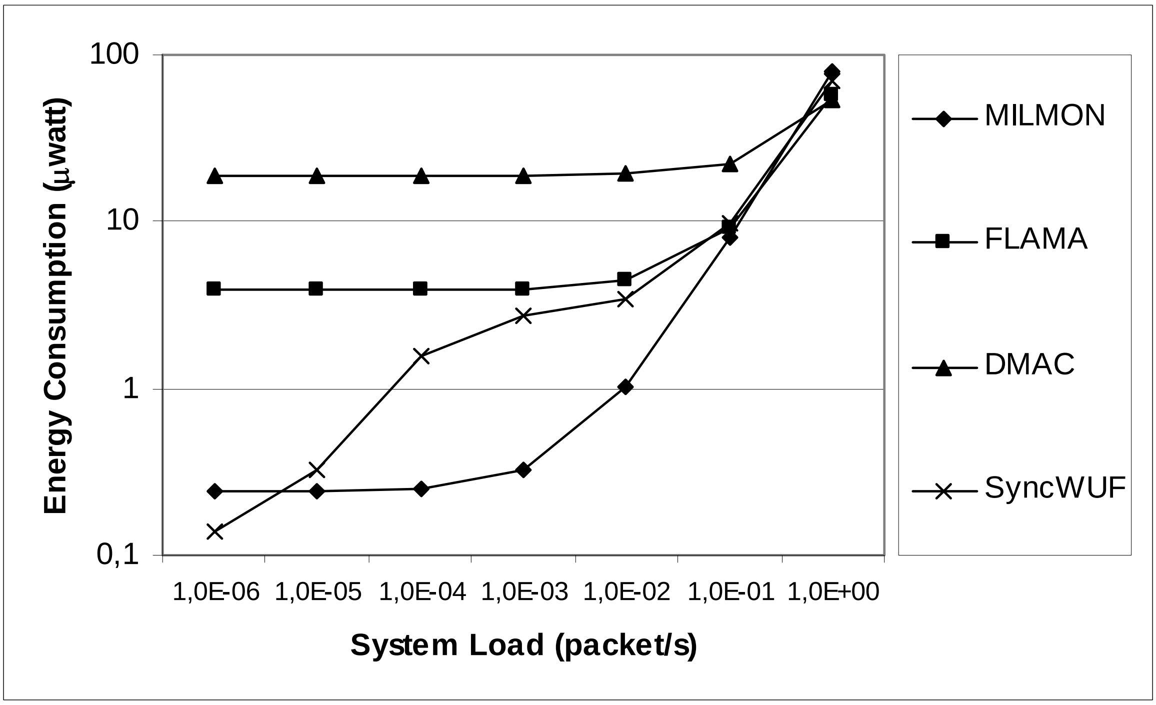
(a) Energy consumption of a node with one child.



(b) Energy consumption of a node with two children.



(c) Energy consumption of a node with three children.



(d) Energy consumption of a node with four children.

Figure 5.10. Energy consumption for different number of children.

If the system load is higher, energy consumption of SyncWUF increases and exceeds not only the new system, but also FLAMA. Energy consumption of SyncWUF increases and exceeds the energy consumption of FLAMA and the new system in higher system loads. The new system consumes less energy than FLAMA and DMAC. The difference is much more remarkable for lower system loads. Newly developed system can save 4.4 times more energy than FLAMA. The difference becomes larger; when the number of children is higher. However, the new system consumes the most energy, if the system load is high. When the system load is 1, its energy consumption of MILMON exceeds DMAC, FLAMA and SyncWUF. Fortunately, MILMON does not operate frequently with the system load 1.

Analytic model is developed to calculate energy consumption for one hop and the data traffic used in analytic model is the incoming data packets to a sensor node. However, energy consumption should be compared for the energy consumption of the whole network under realistic data traffic generated by sample intruders. A simulator is developed to model energy consumption different systems under realistic data traffic.

The system load of the simulator is the number of intruder per second. Intruder comes into system from a random border point of the network area and moves linearly. The speed of the intruder may vary between 3 km/hr which is the speed of a walking man and 100 km/hr which is the speed of a highly mobile vehicle. Inter arrival times between intruders are modeled according to exponential distribution. Network area is modeled as a circle and the sink is at the center of the network area. Network topology is assumed as a tree structure as in the analytic model.

The average energy consumption of FLAMA, SyncWUF and the new system is presented in Figure 5.11. The results of the simulation model are compatible with the results of analytic model.

In MILMON application scenario, MILMON is deployed in certain critical regions on a border. The MILMON region is supposed to be a passage used by the terrorist. Although it is known that MILMON region is used by the terrorist relatively frequently, the existence of intruder is still unlikely. We do not expect terrorist activity in each second.

On the other hand, the terrorist may use the passage more than ones in a month. The expected system load is between intruders in a couple of days and minutes. That means system load is between 10^{-5} and 10^{-3} intruder/s.

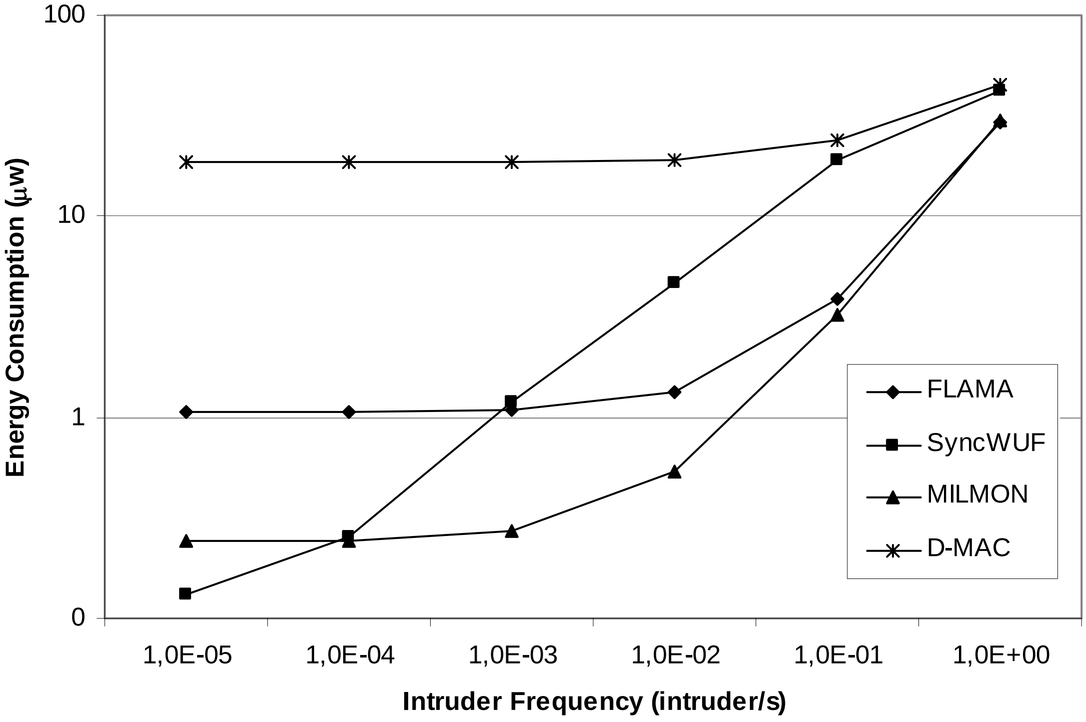


Figure 5.11. Energy consumption in simulation modal.

For all cases, D-MAC is the most energy consuming WSN system. The energy consumption difference between D-MAC and the other WSN systems is much higher in low system loads. When the system load gets higher, energy consumption difference becomes closer.

If the arrival of an intruder is very unlikely, SyncWUF is the most energy saver system. When the system load is higher, energy consumption of SyncWUF also increases, so that it consumes more energy than FLAMA and the new system. While SyncWUF consumes approximately 1.8 times less energy than the new system for very low system load, its energy consumption of SyncWUF is 2.3 times higher than the new system for high system loads.

FLAMA is another system that modeled in simulation. FLAMA consumes more energy than SyncWUF and the new system for low system loads. Energy consumption difference between FLAMA and the new system is up to 4.3 times for low system loads. If system load is higher, the difference becomes closer, so that when the system load is 1, there is no important difference between energy consumptions of FLAMA and the new system.

5.4.2. Fault Tolerance

Sensor nodes are prone to failures and sensor network should be able to operate against single point of failure. If a sensor node dies, the other should be able to continue with a little or no additional recovery procedure. We have developed a fault tolerance metric to compare fault tolerance of different systems. Fault tolerance is the percentage of continuing sensor nodes, after a certain percentage of sensor nodes have died. For example if all the nodes can continue, after 10 per cent sensor nodes have died, its fault tolerance for 10 per cent is 100. If only 80 per cent of the remaining nodes can operate, after 10 per cent of sensor nodes have died its fault tolerance for 10 per cent is 80.

If there are limited number of alternative parents in the system for a certain node, and if all alternative parents die, the children can not send the data and children can not function. If there are more alternative parents, the system becomes more fault tolerant. In an ideal system, if the network is still connected, fault tolerance is always 100. Figure 5.12 shows comparison of fault tolerance performance of our system.

WSN systems without a fault tolerance mechanism, like FLAMA, can not survive, even after a few number of faulty sensor nodes. When the faulty node ratio is 25 per cent, only half of the FLAMA nodes can continue to run. If the fault node ratio is about 35 per cent, there is almost no connected node in the network. DTSM creates a tree topology and there is no alternative parent mechanism. FLAMA node has only one parent, and if the parent node goes off, the child can not continue to operate. Fault tolerance performance of DTSM is the same as the fault tolerance performance of FLAMA.

Fault tolerance comparison shows that fault tolerance mechanism achieves to increase reliability of the network. When fault tolerance parameter is higher, system becomes more fault tolerant. When it is 3, sensor node loses are very low, even if half of the nodes have died. However, if fault tolerance parameter is higher than 3, it has no considerable additional contribution.

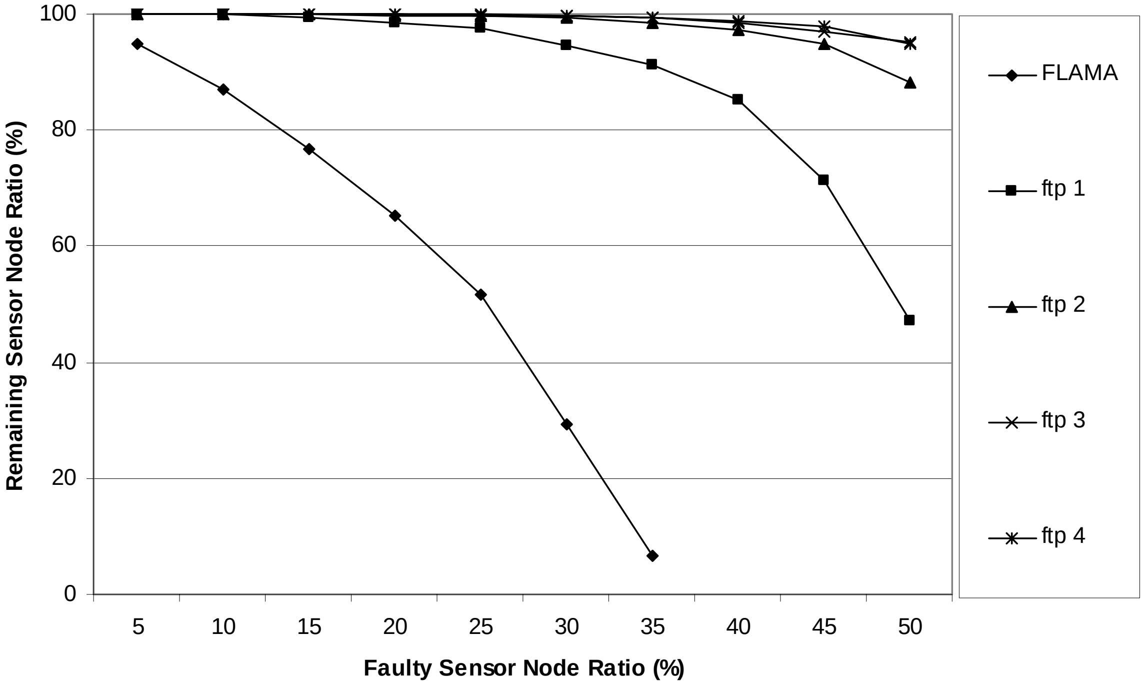


Figure 5.12. Fault tolerance of different systems.

5.4.3. Delay Performance of MILMON

Energy consumption or fault tolerance is not the only design consideration of our system. Delay is also very important for military applications.

Figure 5.13 shows delay performances of FLAMA, D-MAC, SyncWUF and our system with different fault tolerance parameters. Fault tolerance parameter, *ftp*, affects the delay performance. When the parameter is 1, delay is around 1800 ms. If fault tolerance parameter is 4, delay increases approximately 1.7 times. It is shown that when *ftp* is 3, network is more fault tolerant and if it is higher than 3, there is no additional contribution.

SyncWUF delay is much higher than our system. Delay performance difference between MILMON and SyncWUF increases up to 5.8 times, when the distance between event and the sink is high.

Figure 5.13 shows that although delay performance of the newly developed system is much better than SyncWUF and FLAMA, delay performance of D-MAC is the best among the existing designs. D-MAC performs its data traffic in contention based periods and it arrange the periods according to hop numbers of the nodes. In D-MAC, data can be relayed to the next hop immediately, if there is no collision which is unlikely to occur in low system load.

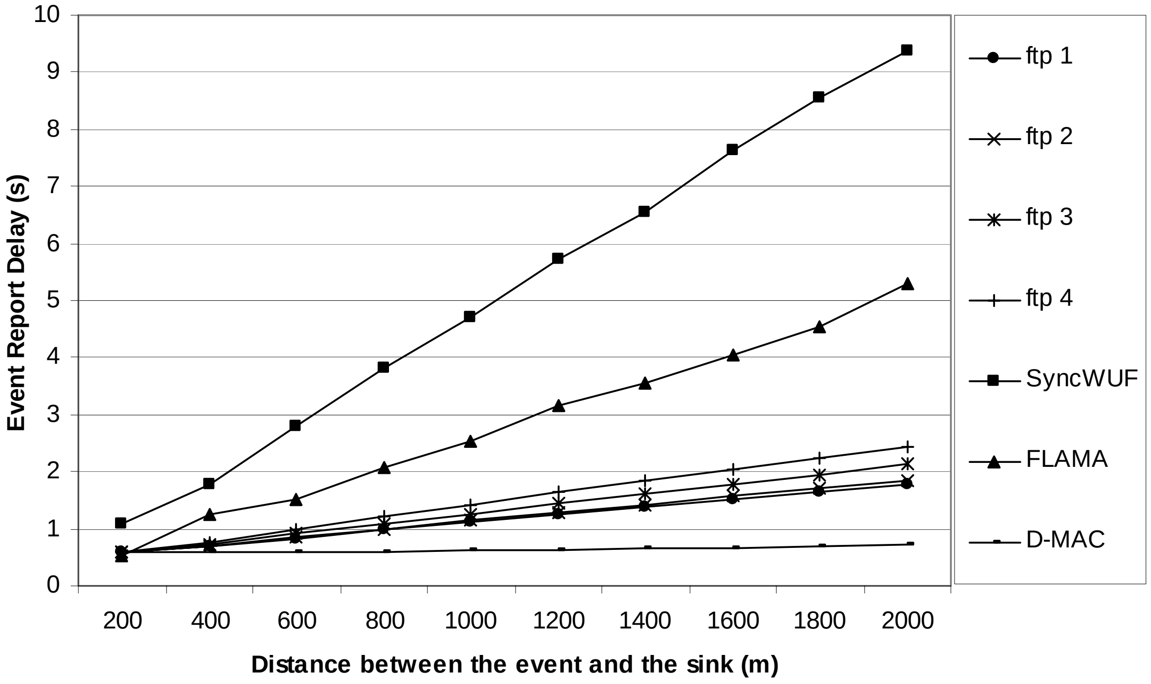


Figure 5.13. Delay performance of different systems.

5.4.4. Combined Metric

The new system is compared with SyncWUF, D-MAC and FLAMA in terms of energy consumption and delay. Our system performs better than SyncWUF in delay and energy consumption. SyncWUF consumes less energy than our system, only if the system load is very low. When the new system is compared with D-MAC, the new system consumes much lower energy than D-MAC. However, delay performance of D-MAC is better than the new system for all cases. Although FLAMA can produce lower delay than SyncWUF, the new system is still better than FLAMA in terms of delay. Energy consumption of the new system is also better than FLAMA, especially when the system load is low. On the other hand the new system has limitations on fault tolerance.

Design objectives of our system are to maximize fault tolerance and to minimize delay and energy consumption. When the performance comparisons are investigated, there is no WSN system that can perform as the best for delay, energy consumption and fault tolerance at the same time. In order to compare our system, SyncWUF, FLAMA and DMAC; a combined metric is proposed. The new metric is *fault tolerance / (delay * energy consumption)*. Fault tolerance performance is accepted as the remaining sensor node percentage, if 30 per cent of the sensor nodes are faulty. Fault tolerance parameter is assumed as 3. SyncWUF and D-MAC is assumed as perfectly fault tolerant. In other words, fault tolerance performances of SyncWUF and D-MAC are assumed as 100. Delay performance is the delay between the most outer regions to the sink. Energy consumption performance is investigated with different system loads. In Figure 5.14, combined metric performances are presented

When the system load is high, the combined performances of different WSN systems are very close. When the system load gets lower, our system performs much better than the existing WSN systems. Although SyncWUF performs better than FLAMA and D-MAC, when the system load is lower than 0.001; our system always outperforms the existing systems.

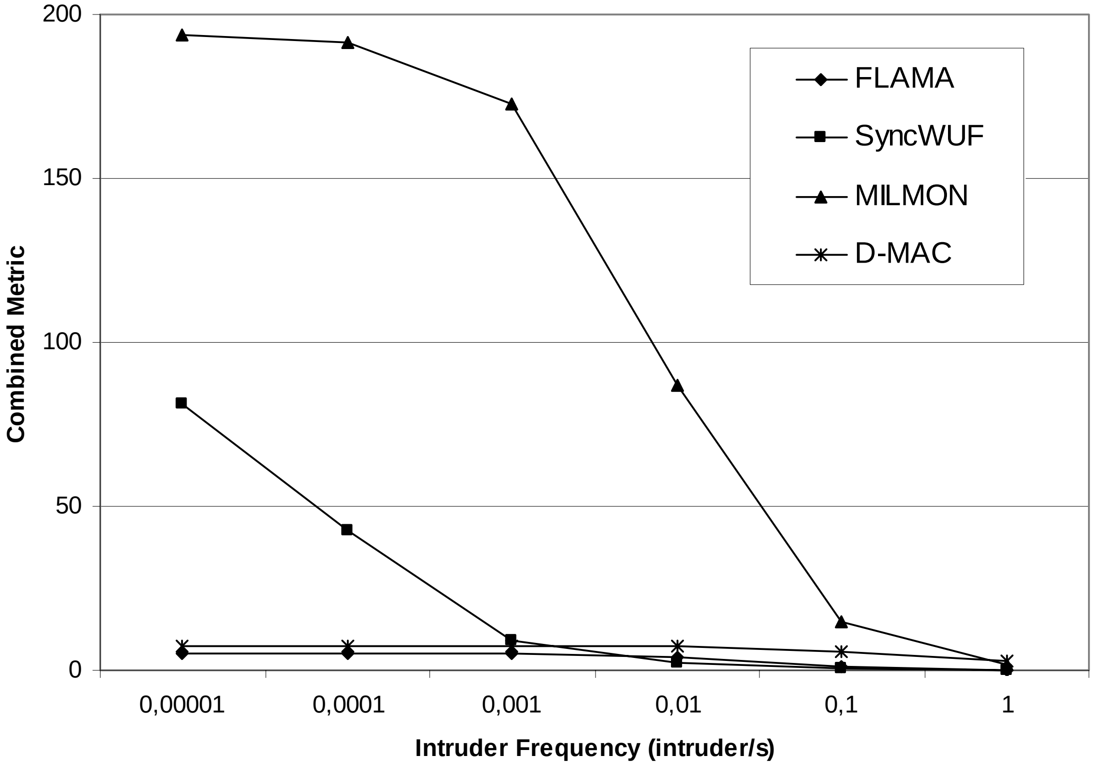


Figure 5.14. Performance of different system for combined metrics.

6. CONCLUSION AND FUTURE WORK

Final section presents the conclusions of the study. Also, possible paths for the future studies are stated.

6.1. Conclusion

Wireless sensor network is a new family of networks that collects information by the help of the collaborative work of several sensor nodes. Each sensor node has sensor/sensors, AC/DC converter, computational power, memory, power unit, and radio circuit. Sensor nodes collect information from the environment through its sensors. It converts the data into digital form, process the data with its computational power and memory. It can also send or receive data by the help of its radio circuit. The work of one sensor node is useless. However, if they organize so that the data produced by the nodes can be collected to a base station, smart environments can be realized.

Sensor nodes are very limited devices. In most of the case, power unit of sensor nodes is an irreplaceable battery. If battery goes off, sensor node also goes off. This is why energy is the most important resource for wireless sensor networks. However, energy is not the only design consideration. Computational power and memory resources of sensor nodes are also very scarce.

In a typical application, sensor nodes are deployed randomly. Nodes organizes themselves so that, sensed data flow from the nodes to a base station, called as the sink. Wireless sensor network has several application areas. It can be used as volcano monitoring, irrigation control system, or patient monitoring system. Application areas of WSN are not restricted to civilian areas. Military applications are also possible with WSN. A sniper location system has been realized by using WSN [11]. Intrusion detection, surveillance and monitoring are common military applications of WSN.

In this thesis, a new wireless sensor network for military monitoring (MILMON) is developed. MILMON monitors its environment and reports the sensed events as soon as possible to the sink. Wireless sensor networks are application specific and all design considerations are different for each application. Especially, military application requirements are very special.

MILMON is supposed to be used to secure a certain area. If there is possible terrorist activities in a certain area, patrolling companies try to secure the area. The company moves in the suspicious area and tries to contact with the terrorists for day and night. The company sleeps in the area. The operation area is not completely secure and it must be monitored constantly. In this case, MILMON can be a solution. The company deploys the sensor nodes around the area randomly. MILMON nodes organize themselves to report the events to the sink. The sink which has infinite or replaceable power source is held by the commander. Whenever terrorists attack on the company, MILMON reports the terrorist actions before they come.

According to the sample scenario, MILMON has to be energy efficient to enlarge network lifetime. Another important consideration of MILMON is delay. All intruders have to be reported as soon as possible. Sensor nodes are cheap and prone to failures. MILMON network structure has to be reliable and fault tolerant against single point of failures.

Time synchronization, data indicator slot and time scheduling mechanisms are essential parts of the new system. SyncHRT which is based on the existence of a high range transmitter is proposed to minimize energy consumption and to maximize precision. Newly proposed data indicator slot mechanism (DISM) helps to reduce energy consumption. A new time slot distribution mechanism, ft_DTSM is also proposed for the new military monitoring system. ft_DTSM uses tiny time slots instead of contention based periods. It has delay handling and fault tolerance mechanism.

SyncHRT is a global WSN time synchronization method which assumes the existence of high range transmitter in the sink. Precision, energy consumption and run time performance of new time synchronization mechanism is investigated with simulation.

Simulation results show that high range transmitter can produce up to 4 times better precision. While it improves precision, it also reduces energy consumption. Energy consumption of time synchronization algorithm can be reduced approximately 3 times. A high range transmitter also affects run time performance of time synchronization. When the range of the sink is higher, run time decreases linearly. Run time of time synchronization can be reduced 11 times.

In addition to SyncHRT, a new delay sensitive, energy efficient and fault tolerant distributed slot assignment algorithm for wireless sensor networks, ft_DTSM, is also proposed. It assumes data traffic of sensor network is convergecast and data always flow from higher hop numbered nodes to lower hop numbered nodes. Although hidden node problem does not allow assigning the same node within two hop neighbors, ft_ DTSM can assign the same slot within two hop neighbors by the help of convergecast traffic assumption. ft_DTSM has also a special mechanism to tolerate failures of the sensor nodes. In order to compare ft_DTSM and common distributed slot assignment algorithms, a simulation model is developed. Extensive set of simulation results show that although energy consumption of ft_DTSM increases with fault tolerance parameter, ft_DTSM is superior to that of FPRP [88], DRAND [94] and FLAMA [65]. Only DTSM can consume less energy than ft_DTSM. ft_DTSM can also handle delay problem successfully. Delay performance of ft_DTSM also better than FPRP, DRAND. Although fault tolerance mechanism increases delay, ft_DTSM is still better than the existing time slot assignment algorithms for wireless sensor networks. DTSM again shows better performance than ft_DTSM in delay. By the help of fault tolerance mechanism, ft_DTSM can distribute the time slots so that sensor network can continue its operation even many sensor nodes fail. ft_DTSM can produce more fault tolerant network than DTSM. Although ft_DTSM can realize low energy consumption, delay and high fault tolerance, its running time proportionally increases with sensor network area and fault tolerance parameter. Fortunately, ft_DTSM can run under in acceptable run time even for a large wireless sensor networks.

Data Indicator Slot Mechanism is another system to reduce energy consumption especially for low traffic requirements, as in MILMON. A sensor node must listen to its children to check existence of data. However, if there is no data, this check is just an overhearing and energy waste. In order to reduce energy consumption for data existence checks, a special slot, Data Indicator Slot (DIS) is defined. When a node has data to relay, is sends a signal at DIS. Instead of checking all the children, sensor nodes checks only DIS. If a sensor node receives a signal at DIS, it checks its all children. If traffic load is low, DISM can save considerable amount of energy.

MILMON is a military monitoring system that is composed of SyncHRT, ft_DTSM and DISM. Energy consumption performance of MILMON is investigated by the help of analytic and simulation models. Analytic and simulation models have shown that energy consumption of the new system is superior to SyncWUF, DMAC, and FLAMA for lower system loads. However if the system load is lower than a certain point, SyncWUF can save more energy than the new system. Delay and fault tolerance performances are also compared for the new system, DMAC, SyncWUF and FLAMA. While our system can produce up to 6 times lower delay than SyncWUF and FLAMA, DMAC can produce lower delay than our system. Newly developed system is very sensible against faulty nodes. Fortunately, it has a fault tolerance mechanism to produce more reliable network. Fault tolerant slot assignment mechanism can improve fault tolerance of our system.

While SyncWUF can save more energy than our system for low system loads, the new system can produce much lower delay than SyncWUF. DMAC can reduce delay more effectively than the new system, but the new system can save much more energy than DMAC. Although there are WSN systems that perform better than our system for only energy consumption or for only delay, new proposed sensor network realizes an optimization on energy, delay and fault tolerance.

6.2. Future Work

By the help of the developing micro-electro-mechanical systems (MEMS) technology, sensor nodes that can collect data, send and receive the data to a center can be produced at lower cost. Each sensor node has sensors, processor, memory and radio circuit. In most of the time, sensor nodes are powered by an irreplaceable battery unit. Wireless sensor networks which is developed as the collaborative work of sensor nodes, are used in many civilian and military applications. In this thesis, a new TDMA based wireless sensor network for military monitoring is proposed. Most proposed wireless sensor networks is designed to sense scalar physical phenomena like temperature, pressure, humidity, movement, or location of objects. In general, most of the applications have low bandwidth demands, and are usually delay tolerant.

More recently, the availability of inexpensive hardware such as CMOS cameras and microphones that are able to capture multimedia content from the environment has fostered the development of Wireless Multimedia Sensor Networks (WMSNs), i.e., networks of wirelessly interconnected devices that allow retrieving video and audio streams, still images, and scalar sensor data [109]. WMSN can realize all the functions of WSNs and it can be used for new applications. Advanced health care systems, traffic avoidance and control systems are example of WMSN application areas.

WMSN nodes contain multimedia sensors, like webcams and microphones. In this case, it consumes mush more energy than a standard wireless sensor node. In some applications, even energy harvesting devices are used to extend the network lifetime [110,111]. It is a fact that energy consumption is still one of the most important problems in WMSN as in WSN. There are some techniques to reduce energy consumption of WMSN nodes. Multi-tiered architecture is one of the solutions of energy consumption problem of WMSN.

SensEye [112] is a multi-tiered architecture for WMSN. It has three tiers In the third tier, classic WSN nodes are used to identify the existence of the intruder. When an intruder is detected, WSN nodes wake up the nodes in the second tier. The second tier nodes are the nodes with low-resolution multimedia sensors. If the collaborative work of the nodes in the second tier confirms of the existence of an intruder, they wake up the nodes in the first tier. The nodes in the first tier contain webcams.

SenseEye is a good example to show that WSN designs can work with WMSN designs. MILMON can also take place in a WMSN design to extend the collected data from scalar data to multimedia data. The performance of MILMON in a multi-tiered WMSN can be investigated and the new algorithms can be developed for MILMON in WMSN.

Another enhancement in wireless sensor networks is the realization of wireless sensor and actor networks (WSAN). Distributed wireless sensor and actor networks are capable of observing the physical world, processing the data, making decisions based on the observations and performing appropriate actions [113]. These networks can be an integral part of any WSN systems.

In WSANs, sensors gather information about the physical world, while actors take decisions and then perform appropriate actions upon the environment which allows a user to effectively sense and act from a distance. In this sense, MILMON can be used the sensor side of a WSAN. While MILMON can detect and locate the intruder, the actor side of WSAN decides and realizes the actions. The actors may be used to destroy the intruder. There are some examples of this kind of intruder detection mechanisms [114].

In order to provide effective sensing and acting, coordination and communication mechanisms are required among sensors and actors. In many WSNs, data flow from sensor nodes to the sink. However, in WSAN, data may flow from sensor nodes to the actor directly. There may be more than one actor, and actors may be mobile. MILMON can be revised to support mobile and multiple actor communication.

Using more than one sink can extend the scalability and the application area of MILMON. One sink can support the work of a company or a limited space. However, border surveillance which is responsible for a much larger area can not be realized with only one sink. If MILMON is extended to use multiple sinks, it is possible to use in larger area. It can be used for the surveillance of an air base, critical areas or even the surveillance of the whole border between two countries.

Another future work for this thesis is the implementation. Implementation of MILMON with real sensor nodes can give more realistic performance results. In this way, MILMON can be tested with realistic data traffic. After the implementation of MILMON in the laboratory, it can be used for military monitoring in patrolling companies in South-East Anatolian region, or it can be used against terrorist actions especially across Turkey and Iraq border.

REFERENCES

1. Gandham S.R., Musunuri R., Rentala P. and Saxena U., "A Survey on Self-Organizing Wireless Sensor Networks", in Zurawski R. (eds.), *Industrial Information Technology Hand Book*, pp. 78-1, CRC Press, 2004.

2. Qi H., Iyengar S.S. and Chakrabarty K., "Distributed Sensor Fusion-A Review Of Recent Research", *Journal of the Franklin Institute*, vol. 338(6), pp. 655-668, 2001.

3. Akyildiz I. F. and Su W. Sankarasubramaniam Y. and Cayirci E. "Wireless Sensor Networks: A Survey", *Computer Networks*, vol. 38 (4), pp. 393–422, April 2002.

4. Lillesand T.M. and Kiefer R.W., *Remote Sensing and Image Interpretation*, Pub. Wiley, 2000.

5. Kahn J. M., Katz R. H. and Pister K. S. J., "Mobile Networking for Smart Dust", "Emerging Challenges: Mobile Networking for Smart Dust", *Journal of Communications and Networks,* vol. 2 (3), pp. 188-196, September 2000.

6. Choi K.J., Song J., "A Miniaturized Mote for Wireless Sensor Networks", *Proc. International Conference on Advanced Communication Technology, ICACT 2008*, vol.1, pp.514-516, 17-20 Feb. 2008

7. Pister K.S.J., "Military Applications of Sensor Networks", *Institute for Defense Analyses,* vol.1, paper 1, 1999.

8. Shih E., Cho S.H., Ickes N., Min R., Sinha A., Wang A., and Chandrakasan A., "Physical Layer Driven Algorithm and Protocol Design for Energy-Efficient Wireless Sensor Networks", *Proc. ACM MOBICOM 2001*, pp. 272-286, Rome, Italy, July 2001.

9. Estrin D., Girod L., Pottie G., and Srivastava M., “Instrumenting the World with Wireless Sensor Networks”, *Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2001)*, Salt Lake City, Utah, May 2001.

10. Schneider B.R. and Grinter L.E. (editors), *Battlefield of the Future: 21th Century Warfare Issues*, Air War College Studies in National Security, Montgomery, AL: Air University Press, 1995.

11. Bonen Z., *Advanced Technology and Future Warfare*, Begin-Sadat Centre for Strategic Studies, Mid-East Security and Policy Studies, No. 28, Part II.

12. Kumar S. and Shepherd D., “SensIT: Sensor Information Technology for the War Fighter”, *Proc. Int. Conf. on Information Fusion*, pp. TuC1-3–TuC1-9, 2001.

13. Diamond, S.M.; Ceruti, M.G., “Application of Wireless Sensor Network to Military Information Integration”, *Proc. IEEE International Conference on Industrial Informatics,* vol.1, pp.317-322, June 2007

14. Wang Y., Wang X., Xie B., Wang D., Agrawal D.P., “Intrusion Detection in Homogeneous and Heterogeneous Wireless Sensor Networks”, *IEEE Transactions on Mobile Computing*, vol.7, no.6, pp.698-711, June 2008

15. L3 Communication Systems, *REMBASS-II Ground Sensor System*, http://www.l-3com.com/products-services, 2008.

16. SenTech Inc., *Sniper Location System*, http://www.sentech-acoustic.com/page5.htm.

17. O'Neil D.J., Ivnitski D.M., Gattuso A., Schlicht R. and Calidonna M., “Advanced Biological and Chemical Sensor Technology for Deployment in the Environmental Systems Management, Analysis and Reporting neTwork (E-SMART)”, *Proc. Workshop on Unified Science and Technology for Biological Threat Reduction*, 2001.

18. Mainwaring A., Polastre J., Szewczyk R., CullerD., and Anderson J., “Wireless Sensor Networks for Habitat Monitoring”, *Proc. ACM International Workshop on Wireless Sensor Networks and Applications (WSNA'02)*, pp.88-97, Atlanta, GA, September 2002.

19. Chandrakasan A., Amirtharajah R., Cho S., Goodman J., Konduri G., Kulik J., Rabiner W., and Wang A., “Design Considerations for Distributed Micro-Sensor Systems”, *Proc. IEEE Custom Integrated Circuit Conference*, San Diego, CA, pp. 279-286 May 1999.

20. Bonnet P., Gehrke J., and Seshadri P., “Querying the Physical World”, *IEEE Personal Communications*, vol. 7, pp. 10 - 15, October 2000.

21. Bauer P., Sichitiu M., Istepanian R., and Premaratne K., “The Mobile Patient: Wireless Distributed Sensor Networks for Patient Monitoring and Care”, *Proc. IEEE Conference on Information Technology Applications in Biomedicine (IEEE ITAB 2000)*, pp. 17-21, November 2000

22. Martinez, I., Fernandez, J., Galarraga, M., Serrano, L., de Toledo, P., Jimenez-Fernandez, S., Led, S., Martfnez-Espronceda, M., Garcia, J., “Implementation of an End-To-End Standard-Based Patient Monitoring Solution”, *IEEE Communications*, vol.2, no.2, pp.181-191, February 2008.

23. Fang W., Kedar S., “System Architecting and System-on-Chip Design of Intelligent Sensor Networks for Active Volcanoes”, *Proc. IEEE Systems Conference*, vol., pp.1-8, 7-10 April 2008

24. Yunseop K.; Evans R.G., Iversen W.M., “Remote Sensing and Control of an Irrigation System Using a Distributed Wireless Sensor Network”, *IEEE Transactions on Instrumentation and Measurement*, vol.57, no.7, pp.1379-1387, July 2008

25. Kahn J. M., Katz R. H. and Pister K. S. J., “Mobile Networking for Smart Dust”, *Proc. ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom 99)*, Seattle, WA, August 1999.

26. Khakpour, K.; Shenassa, M.H., “Industrial Control Using Wireless Sensor Networks”, *Proc. Information and Communication Technologies: from Theory to Applications, ICTTA 2008*, vol., no., pp.1-5, April 2008

27. Pottie G.J. and Kaiser W.J., “Wireless Integrated Network Sensors”, *Communications of the ACM*, vol. 43, no. 5, May 2000, pp, 51-58.

28. Heinzelman W., Chandrakasan A., and Balakrishnan H., “An Application-Specific Protocol Architecture for Wireless Microsensor Networks”, *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 660-670, October 2002.

29. Hempstead, M.; Gu-Yeon Wei; Brooks, D., “System Design Considerations for Sensor Network Applications”, *Proc. IEEE International Symposium on Circuits and Systems, ISCAS 2008*, pp.2566-2569, 18-21 May 2008

30. Lin C.R. and Liu J.S., “QoS Routing in Ad Hoc Wireless Networks”, *IEEE Journal of Selected Areas on Communication*, vol. 17 (8), p. 1426, August 1999.

31. Sobrinho J. and Krishnakumar A.S., “Quality of Service in Ad Hoc Carrier Sense Multiple Access Wireless Networks”, *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 6, pp. 1353–1368, Aug. 1999

32. Chakrabarti S. and Mishra A., “QoS Issues in Ad Hoc Wireless Networks”, *IEEE Communications Magazine*, vol. 39, pp. 142 - 148, February 2001.

33. Liu, H., Jia, X., Wan, P., Yi, C., Makki, S. K., and Pissinou, N. “Maximizing Lifetime of Sensor Surveillance Systems”, *IEEE/ACM Transactions on Network,* vol. 15(2), pp. 334-345, April 2007.

34. Chang J. and Tassiulas L., "Energy Conserving Routing in Wireless Ad-Hoc Networks" *Proc. IEEE INFOCOM*, vol.1, pp.22-31, Tel Aviv, Israel, March 2000.

35. Wattenhofer R., Li L., Bahl P., and Wang Y., "Distributed Topology Control for Power Efficient Operation in Multihop Wireless Ad Hoc Networks," *Proc. IEEE INFOCOM*, pp. 1388–1397, Anchorage, AK, April 2001.

36. Xu Y., Heidemann J., and Estrin D., "Geography-Informed Energy Conservation for Ad Hoc Routing", *Proc. ACM SIGMOBILE 7/01*, pp. 70–84, Rome, Italy, 2001.

37. Toh C.K., "Maximum Battery Life Routing To Support Ubiquitous Mobile Computing in Wireless Ad Hoc Networks", *IEEE Communication Magazine*, vol. 39, no. 6, pp. 138–147, Jun. 2001.

38. Li, J.; AlRegib, G., "Function-Based Network Lifetime for Estimation in Wireless Sensor Networks", *IEEE Signal Processing Letters*, vol.15, no., pp.533-536, 2008.

39. Heinzelman W., *Application-Specific Protocol Architectures for Wireless Networks*, PhD Thesis, Massachusetts Institute of Technology, June 2000.

40. Smithgall D., "Toward the 60 gm Wireless Phone", *Proc. Radio and Wireless Conference (RAWCON'98)*, pp. 157-159, August 1998.

41. Rappaport T., *Wireless Communications: Principals and Practice*, Prentice-Hall, Englewood Cliffs, NJ, 1996.

42. Heinzelman W., Chandrakasan A., and Balakrishnan H., "Energy-Efficient Communication Protocols for Wireless Microsensor Networks", *Proc. Hawaiian International Conference on Systems Science*, vol.2, p.10, January 2000.

43. Srinivasan V., Chiasserini C.F., Nuggehalli P. and Rao R.R., "Optimal Rate Allocation and Traffic Splits for Energy Efficient Routing in Ad Hoc Networks", *Proc. IEEE INFOCOM*, vol.2, pp.950-957, New York, NY, USA, June 2002

44. Min R. and Chandrakasan A., “Energy-Efficient Communication for Ad-Hoc Wireless Sensor Networks”, *Proc. Conference on Signals, Systems, and Computers*, vol. 1, pp. 139-143, November 2001.

45. Mahfoudh, S.; Minet, P., “Survey of Energy Efficient Strategies in Wireless Ad Hoc and Sensor Networks”, *Proc. International Conference on Networking, ICN 2008*, pp.1-7, April 2008

46. Burrel J., Brooke T. and Beckwith R., “Vineyard Computing: Sensor Networks in Agriculture Production”, *IEEE Pervasive Computing*, vol.3, pp.38-45, 2004.

47. Hoblos G., Staroswiecki M., Aitouche A., “Optimal Design of Fault Tolerant Sensor Networks”, *Proc. IEEE International Conference on Control Applications*, pp. 467–472, Anchorage, AK, September 2000.

48. Cai W., Jin X., Zhang Y., Chen K., Tang J., “Research on Reliability Model of Large-Scale Wireless Sensor Networks”, *Proc. International Conference on Wireless Communications, Networking and Mobile Computing, WiCOM 2006*, vol., no., pp.1-4, 22-24 Sept. 2006

49. Bluetooth Inc., *Specification of Bluetooth System volume 1 version 1.1*, http://www.bluetooth.com, Feb. 2001.

50. Crow B.P., Widjaja I., Prescott J.G., Sakai T., “IEEE 802.11 Wireless Local Area Networks”, *IEEE Communication Magazine*, vol. 35(9), pp. 116-126, September 1997.

51. Bharghavan V., Demers A., Shenker S., and Zhang L., “MACAW A Media Access Protocol for Wireless LAN's”, *Proc. ACM SIGCOMM Conference (SIGCOMM '94)*, pages:212-225, August 1993.

52. Ye W., Heidemann J. and Estrin D., “An Energy-Efficient MAC Protocol for Wireless Sensor Networks”, *Proc. IEEE INFOCOM 2002*, vol.3, pp. 1567-1576, New York, NY, USA, June 2002.

53. Stemm M. and Katz R.H., “Measuring and Reducing Energy Consumption of Network Interfaces in Hand-Held Devices”, *IEICE Transaction on Communication*, E80-B(8), pp. 1125-1131, August 1997.

54. Lu. G., Krishnamachari, B., Raghavendra C.S., “An Adaptive Energy-Efficient and Low-Latency MAC for Data Gathering in Wireless Sensor Networks”, *Proc. Parallel and Distributed Processing Symposium 2004*, pp. 224, 26-30 April 2004.

55. Demirkol, I., Ersoy, C., Alagoz, F., “MAC Protocols for Wireless Sensor Networks: A Survey”, *IEEE Communications Magazine*, vol. 44(4), pp.115 – 121, April 2006.

56. Polastre J., Hill J., and Culler D., “Versatile Low Power Media Access for Wireless Sensor Networks”, *Proc. International Conference on Embedded Networked Sensor Systems (SenSys 2004)*, pp. 95–107.

57. Enz C. C., El-Hoiydi A., Decotignie J-D., Peiris V., “WiseNET: An Ultralow-Power Wireless Sensor Network Solution”, *IEEE Computer Magazine*, vol. 37(8), pp.62 - 70, Aug. 2004.

58. Xiaolei S.; Stromberg G., “SyncWUF: An Ultra Low-Power MAC Protocol for Wireless Sensor Networks”, *IEEE Transactions on Mobile Computing*, vol. 6(1), pp. 115-125, Jan. 2007.

59. Paladina, L.; Scarpa, M.; Puliafito, A., “Advantages in Synchronization for Wireless Sensor Networks”, *Proc. Wireless Pervasive Computing ISWPC 2008*, pp.160-164, May 2008

60. Sohrabi K., Gao J., Ailawadhi V., and Pottie G. J., "Protocols for Self-Organization of A Wireless Sensor Network", *IEEE Personal Communications*, vol. 7, pp. 16-27, October 2000.

61. Shepard T., *Decentralized Channel Management in Scalable Multihop Spread Spectrum Packet Radio Networks*, Tech. Rep. MIT/LCS/TR-670, Massachusetts Institute of Technology Laboratory for Computer Science, July 1995.

62. Haiming Yang; Sikdar, B., "Optimal Cluster Head Selection in the LEACH Architecture", *Proc. IEEE International Performance, Computing, and Communications Conference IPCCC 2007,* pp.93-100, 11-13 April 2007

63. Kottapalli, A. S. Kiremidjian, J. P. Lynch, E. C., T. W. Kenny, K. H. Law, Y. Lei, "Two-Tiered Wireless Sensor Network Architecture for Structural Health Monitoring", *Proc. Conference on Smart Systems and Nondestructive Evaluation for Civil Infrastructures 2003*, San Diego CA, vol. 5057, pp. 8-19.

64. Rajendran V., Obraczka K., Garcia-Luna-Aceves J.J., "Energy-Efficient, Collision-Free Medium Access Control for Wireless Sensor Networks", *Proc. ACM SenSys 03*, pp. 181 - 192, 5-7 November 2003.

65. Rajendran V.; Garcia J. J.; Obraczka K., "Energy-Efficient, Application-Aware Medium Access for Sensor Networks", *Proc. IEEE International Conference on Mobile Ad-Hoc and Sensor Systems MASS 2005*, November 7-10; Washington; D.C.; USA.

66. Frieman J., "Portable Computer Power Sources", *Proc. the Ninth Annual Battery Conference on Application and Advances*, pp. 152-158, January 1994.

67. Wellenhof H., Lichtenegger H., and Collins J., *GPS Theory and Practice*, Springer-Verlag Wien, NewYork, 1997.

68. Kopetz H. and Schwabl W., *Global Time in Distributed Real-Time Systems*, Technical Report 15/89, Technische Universit¨at Wien, 1989.

69. Maroti M., Kusy B., Simon G., Ledeczi A, "The Flooding Time Synchronization Protocol", *Proc. ACM SenSys 04*, pp. 39-49, Baltimore.

70. Elson J., Girod L., and Estrin D., "Fine-Grained Network Time Synchronization Using Reference Broadcasts", *Proc. Fifth Symposium on Operating Systems Design and Implementation (OSDI 2002)*, pp.147-163, December 2002.

71. Bekmezci I. and Alagoz F., "A New TDMA Based Sensor Network for Military Monitoring (MIL-MON)", *Proc. IEEE MILCOM 2005*, vol.4, pp. 2238 – 2243, Oct. 2005.

72. Sundararaman B., Buy U. and Kshemkalyani A.D., "Clock Synchronization for Wireless Sensor Networks: A Survey", *Ad Hoc Networks*, vol. 3, pp. 281-323, May 2005.

73. Ganeriwal S., Kumar R. and Srivastava M.B.. "Timing-Sync Protocol for Sensor Networks", *Proc. ACM Conference on Embedded Networked Sensor Systems 2003*, pp. 138 – 149, November 2003.

74. Mills D.L., "Internet Time Synchronization: The Network Time Protocol", *IEEE Transactions on Communications*, vol. 39(10), pp.1482–1493, October 1991.

75. Cristian F., "Probabilistic Clock Synchronization", *Journal of Distributed Computing*, vol. 3, pp.146–158, 1989.

76. Srikanth T. K. and Toueg S., "Optimal Clock Synchronization", *Journal of ACM*, vol. 34(3), pp. 626–645, 1987.

77. Lundelius J. and Lyunch N., "A New Fault-Tolerant Algorithm for Clock Synchronization", *Information and Computation*, vol. 77(1), pp. 1–36, 1988.

78. Rentel, C.H.; Kunz, T., “A Mutual Network Synchronization Method for Wireless Ad Hoc and Sensor Networks”, *IEEE Transactions on Mobile Computing*, vol.7, no.5, pp.633-646, May 2008

79. Sichitiu M.L. and Veerarittiphan C., “Simple, Accurate Time Synchronization for Wireless Sensor Networks”, *Proc. IEEE Wireless Communications and Networking Conference (WCNC’03)*, vol.2, pp.1266-1273, March 2003.

80. Greunen J. and Rabaey J. “Lightweight Time Synchronization for Sensor Networks”, *Proc. 2nd ACM International Workshop on Wireless Sensor Networks and Applications*, vol.1, pp. 11–19, September 2003.

81. Hill J., Szewczyk R., Woo A., Hollar S., Culler D., and Pister K., “System Architecture Directions for Networked Sensors”, *Proc. Ninth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS-IX)*, vol.1, pp. 93–104, Cambridge, MA, USA, November 2000.

82. Vig J.R., *Introduction to Quartz Frequency Standards*, *Technical Report SLCET-TR-92-1*, Army Research Laboratory, Electronics and Power Sources Directorate, October 1992.

83. Mannermaa J., Kalliomaki K., Mansten T., and Turunen S., “Timing Performance Of Various GPS Receivers”, *Proc. 1999 Joint Meeting of the European Frequency and Time Forum and the IEEE International Frequency Control Symposium*, pp. 287–290, April 1999.

84. Bekmezci I., F. Alagoz, “Delay Sensitive, Energy Efficient and Fault Tolerant Distributed Slot Assignment Algorithm for Wireless Sensor Network Under Convergecast Data Traffic”, to appear in *International Journal of Distributed Sensor Networks*, vol. 4.

85. Pond L. C. and Li V. O. K., “A Distributed Time-Slot Assignment Protocol for Mobile Multi-Hop Broadcast Packet Radio Networks”, *Proc. IEEE MILCOM 89*, pages 70–74, 1989.

86. Pond L. C. and Li V. O. K., “A Distributed Media Access Protocol for Packet Radio Networks and Performance Analysis - Part 1: Network Capacity”, *International Journal of Communication Systems*, vol. 8, pp. 27-47, Jan.-Feb. 1995.

87. Pond L. C. and Li V. O. K., “A Distributed Media Access Protocol for Packet Radio Networks and Performance Analysis - Part 2: Network setup time and data rate”, *International Journal of Communication Systems*, vol. 8, pp. 49~53, Jan.-Feb. 1995.

88. Zhu, C. and Corson, “A Five-Phase Reservation Protocol (FPRP) for Mobile Ad Hoc Networks”, *Wireless Networks*, vol.4, pp. 371-384, Sep. 2001.

89. Loscrì V., De Rango F., Marano S., “Performance Evaluation of AODV Protocol over E-TDMA MAC Protocol for Wireless Ad Hoc Networks”, *Lecture Notes in Computer Science*, vol.3124, pp.417 – 424, Jan 2004.

90. Tang Z., and Garcia J.J., “Hop Reservation Multiple Access (HRMA) for Multichannel Packet Radio Networks”, *Proc. IEEE Seventh International Conference on Computer Communications and Networks IC3N 98*, pp.398-395, October 1998.

91. Patro R.K., Perumal K., “Neighbor Based TDMA Slot Assignment Algorithm for WSN”, *IEEE INFOCOM 2005 poster session*, Miami, March 2005.

92. Kanzaki, A., Hara, T., and Nishio, S. “An Adaptive TDMA Slot Assignment Protocol in Ad Hoc Sensor Networks”, *Proc. 2005 ACM Symposium on Applied Computing*, pp. 1160-1165, New York, NY.

93. Kulkarni S.S., Arumugam M.U., “TDMA Service for Sensor Networks”, *Proc. 24th International Conference on Distributed Computing Systems Workshops - W5: ADSN (ICDCSW'04)*, vol. 5(5), pp.604-609, May 2004.

94. Rhee I., Warrier A., Min J., Xu L., "DRAND: Distributed Randomized TDMA Scheduling for Wireless Ad Hoc Networks", *Proc. International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2006)*, pp. 190–201.

95. Ramanathan S., "A Unifed Framework and Algorithms for (T/F/C)DMA Channel Assignment in Wireless Networks", *Proc. IEEE INFOCOM 1997*, pp. 900,907.

96. Herman T. and Tixeuil S., "A Distributed TDMA Slot Assignment Algorithm for Wireless Sensor Networks", *Proc. First Workshop on Algorithmic Aspects of Wireless Sensor Networks (AlgoSensors'2004)*, pp.45-58, Turku, Finland, July 2004.

97. Barroso A., Roedig u., and Sreenan C.J., "µMAC: An Energy-Efficient Medium Access Control for Wireless Sensor Networks", *Proc. 2nd IEEE European Workshop on Wireless Sensor Networks (EWSN2005)*, pp.70-80, Istanbul, Turkey, February 2005.

98. Herman, T.; Pemmaraju, S.; Pirwani, I.; "Oriented Edge Colorings and Link Scheduling in Sensor Networks", *Proc. Communication System Software and Middleware* (*Comsware 2006),* pp. 1-6.

99. Liu, A.; Hongyi Yu; Lin Li; "An Energy-Efficiency and Collision-Free MAC Protocol for Wireless Sensor Networks", *Proc. Vehicular Technology Conference 2005*, vol. 2, pp.1317 – 1322.

100. Bekmezci I., Alagoz F., "A New Delay Sensitive and Energy Efficient Distributed Slot Assignment Algorithm for Sensor Networks Under Convergecast Data Traffic", *Journal of Communications Software and Systems*, vol. 2(4), pp. 331-339, December 2006.

101. Bekmezci I., Alagoz F, "Delay Sensitive and Energy Efficient Distributed Slot Assignment Algorithm for Wireless Sensor Network Under Convergecast Data Traffic", *LNSC 2006*, vol. 4003*, pp.* 353-365.

102. Ye W. and Heidemann J., "Medium Access Control in Wireless Sensor Networks", in Znati T., Sivalingam K.M. and Raghavendra c. (eds.), *Wireless Sensor Networks* , Chapter 4, Kluwer Academic Publishers, May 2004.

103. Bari A., Jaekel A., Bandyopadhyay S., "Maximizing the Lifetime of Two-Tiered Sensor Networks", *Proc. IEEE International Conference on Electro/Information Technology 2006,* pp.:222 – 226, May 2006.

104. Ergen, S.C., Varaiya, P., "PEDAMACS: Power Efficient and Delay Aware Medium Access Protocol for Sensor Networks", *Proc. IEEE Transactions on Mobile Computing* vol. 5(7), pp.920 – 930, July 2006.

105. Rivest R. L., *Network Control by Baynesian Broadcast*, Report MIT/LCS/TM-287, MIT, Laboratory for Computer Science, Cambridge, MA, 1985.

106. Bekmezci I, Alagoz F., Arslan M.G., "Periodic Global Broadcast Time Synchronization (PGB-TS) for TDMA Based Sensor Networks", *Proc. IEEE Recent Advances in Space Technologies (RAST 07)*, pp. 531 – 536, Istanbul, June 2007.

107. Bekmezci I.; Alagoz F., "Energy Efficient, Delay Sensitive, Fault Tolerant Wireless Sensor Network for Military Monitoring", *Proc. IEEE Sensors Applications Symposium. (SAS) 2008*, pp.172-177, 12-14 Feb. 2008

108. Onur E., *Deployment Quality Measures in Surveillance Wireless Sensor Networks*, PhD Thesis, Boğaziçi University, July 2007.

109. Akyildiz, I. F., Melodia, T., and Chowdhury, K. R., "A Survey on Wireless Multimedia Sensor Networks", *Computer Networks*, vol. 51(4), pp. 921-960, May 2005

110. Jiang X., Polastre J., Culler D., "Perpetual Environmentally Powered Sensor Networks", *Proc. IEEE Workshop on Sensor Platform, Tools and Design Methods*

for Networked Embedded Systems (SPOTS 2005), Los Angeles, CA, USA, April 2005.

111. Paradiso J., Starner T., “Energy Scavenging for Mobile and Wireless Electronics”, *IEEE Pervasive Computing,* vol. 4(1), pp. 18-27, 2005.

112. Kulkarni P., Ganesan D., Shenoy P., Lu Q., “SensEye: A Multi-Tier Camera Sensor Network”, *Proc. ACM Multimedia*, Singapore, pp. 229-238, November 2005.

113. Akyildiz I. F., Kasimoglu I. H., “Wireless Sensor and Actor Networks: Research Challenges”, *Ad Hoc Networks Journal*, vol. 2(4), pp. 351-367, October 2004.

114. Li Y., Parker L.E., “Intruder Detection Using a Wireless Sensor Network with An Intelligent Mobile Robot Response”, *Proc. IEEE SoutheastCon 2008*, pp.37-42, April 2008.