A FRAMEWORK FOR THE USE OF WIRELESS SENSOR NETWORKS IN FOREST FIRE DETECTION AND MONITORING

A THESIS

SUBMITTED TO THE DEPARTMENT OF COMPUTER ENGINEERING

AND THE INSTITUTE OF ENGINEERING AND SCIENCE

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

By

Yunus Emre Aslan

August, 2010

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Asst. Prof. Dr. İbrahim Körpeoğlu (Advisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Dr. Özgür Ulusoy(Co-Advisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Uğur Güdükbay

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Dr. İlyas Çiçekli

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Ahmet Coşar

Approved for the Institute of Engineering and Science:

Prof. Dr. Levent Onural
Director of the Institute

ABSTRACT

A FRAMEWORK FOR THE USE OF WIRELESS SENSOR NETWORKS IN FOREST FIRE DETECTION AND MONITORING

Yunus Emre Aslan
M.S. in Computer Engineering
Supervisors:
Asst. Prof. Dr. İbrahim Körpeoğlu and
Prof. Dr. Özgür Ulusoy
August, 2010

Wireless sensor networks have a broad range of applications in the category of environmental monitoring. In this thesis, we consider the problem of forest fire detection and monitoring as a possible application area of wireless sensor networks. Forest fires are one of the main causes of environmental degradation nowadays. The current surveillance systems for forest fires lack in supporting real-time monitoring of every point of the region at all time and early detection of the fire threats. Solutions using wireless sensor networks, on the other hand, can gather temperature and humidity values from all points of field continuously, day and night, and, provide fresh and accurate data to the fire fighter center quickly. However, sensor networks and nodes face serious obstacles like limited energy resources and high vulnerability to harsh environmental conditions, that have to be considered carefully.

In our study, we propose a comprehensive framework for the use of wireless sensor networks for forest fire detection and monitoring. Our framework includes proposals for the wireless sensor network architecture, clustering and communication protocols, and environment/season-aware activity-rate selection schemes to detect the fire threat as early as possible and yet consider the energy consumption of the sensor nodes and the physical conditions that may hinder the activity of the network. We also implemented a simulator to validate and evaluate our proposed framework, which is using an external fire simulator library. We did extensive simulation experiments and observed that our framework can provide fast reaction to forest fires while also consuming energy efficiently.

Keywords: Wireless sensor networks, forest fire detection, fire forecast.

ÖZET

KABLOSUZ DUYARGA AĞLARI KULLANILARAK ORMAN YANGINLARI İZLEME VE ERKEN TESPİT SİSTEMİ

Yunus Emre Aslan
Bilgisayar Mühendisliği, Yüksek Lisans
Tez Yöneticileri:
Yrd. Doç. Dr. İbrahim Körpeoğlu ve
Prof. Dr. Özgür Ulusoy
Ağustos, 2010

Kablosuz duyarga ağları kullanılarak doğal ortamların izlenmesi üzerine bir çok uygulama alanı geliştirilmiştir. Bu tez çalışmamızda, bizler de orman yangınlarının erken tespitinde ve yangının izlenmesi sürecinde kablosuz duyarga ağlarını kullanarak bir sistem tasarladık. Orman yangınları dünyada çevresel tahribata neden olan başlıca sebeplerden biridir. Şu anki yangın gözetleme ve takip sistemleri ormanları anlık olarak bütünüyle izleme ve olası bir yangın tehlikesini önceden tespit etme konusunda başarısız olmaktadır. Öte yandan, kablosuz duyarga ağlarını kullanarak geliştirilen çözümler sıcaklık ve nem değerlerini, anlık olarak, sahanın farklı noktalarından, gece ve gündüz farketmeksizin sürekli olarak alabilmekte ve de merkezi birimlere taze ve güvenilir bilgi sunabilmektedir. Fakat, duyarga ağlarında kullanılan duyarga düğümleri kısıtlı enerji kaynaklarına sahiptir ve zorlu dış koşullara karşı dayanıklı değillerdir. Geliştirilen uygulamalarda bu engellerin dikkatli bir şekilde ele alınması gereklidir.

Tez çalışmamızda kablosuz duyarga ağlarını kullanarak orman yangınlarını erken tespit etmek ve izleyebilmek amacıyla geniş kapsamlı bir sistem geliştirdik. Sunduğumuz sistem kablosuz duyarga ağlarıyla ilgili bir ağ altyapısı, düğümlerin ormana yerleştirilmesi ile ilgili özel bir mekanizma ve düğümlerin küme içi ve kümeler arası iletişim protokollerini içermektedir. Sistemimiz orman yangınlarını mümkün olan en kısa sürede tespit etmeyi hedeflerken, düğümlerin enerji harcama oranlarını da dikkatlice gözetmektedir. Ayrıca sistemin çalışmasını engelleyebilecek zorlu çevresel koşullar için de önlemler hazırlanmıştır. Sunduğumuz sistemi geliştirebilmek, test edebilmek ve farklı yapılarla kıyaslayabilmek adına bir de simülator gelistirdik. Bununla birlikte yangının başlaması ve ilerlemesi ile ilgili olarak 3. parti bir yangın simülatörünü kullandık. Simülatör üzerinde çok çeşitli testler yaparak sunduğumuz sistemin potensiyel yangınları tespit etmekte daha hızlı tepki verdiğini ve daha az enerji tükettiğini gözlemledik.

Anahtar sözcükler: Kablosuz duyarga ağları, orman yangınları erken tespit sistemi.

Acknowledgement

I would like to thank to my supervisor, Asst. Prof. Dr. İbrahim Körpeoğlu for always being available to me when I needed help. For the last five years, I have learned a lot from him not only about my research but also about life.

I also thank to my co-advisor Prof. Dr. Özgür Ulusoy for his valuable comments, helps and patience throughout this study.

I am grateful to my jury members, Assoc. Prof. Dr. Uğur Güdükbay, Dr. İlyas Çiçekli and Assoc. Prof. Dr. Ahmet Coşar for reading and reviewing this thesis.

I would like to acknowledge TÜBİTAK BIDEB for their financial support.

I am also grateful to my friends, Enver Kayaaslan, Alper Rıfat Uluçınar and Eren Algan for their help and support throughout last three years.

I thank to my family for supporting me with all my decisions and for their endless love.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this thesis, we explore the use of wireless sensor network technology in real-time forest fire detection. Forest fire is a fatal threat throughout the world. It is reported that for the last decade, each year, a total of 2000 wild fires happened in Turkey and more than 100000 in all countries [1]. Early detection of forest fires is the most effective factor in the struggle against fires. Spread features of forest fires show that, in order to put out the fire without making any permanent damage in the forest, the fire fighter center should be aware of the threat in at most 6 minutes after the start of the fire [4]. Also, together with the early detection capability, estimating the spread direction and speed of fire is another critical point which is important in extinguishing the fires.

Unreliability of human observation towers, in addition to the difficult life conditions of fire lookout personnel, have led the development of several technological studies aiming to make the fire fighters aware of the forest fire as early as possible. Some important technologies and systems that are currently used towards this goal are: systems employing charge-coupled device (CCD) cameras and Infrared (IR) detectors, satellite systems and images, and wireless sensor networks.

In a camera based system, CCD cameras and IR detectors are installed on top of towers. In case of fire or smoke activity, the cameras and detectors sense this abnormal event and report it to the control center [5, 7]. However, the accuracy of this system is highly affected by weather conditions such as clouds, light reflections and smoke from industrial or social and innocent activities. Moreover, since most of the forests are located on mountains, the sight view of devices will not be clear enough to control the whole forest. Also, considering the cost of the system and technical incapabilities of the devices whose view areas are not enough to cover a forest, it is seen that automatic video surveillance systems cannot be always applied effectively to large forest fields.

Another alternative technology for forest fire detection is the use of satellites and satellite images. Current satellite-based forest fire detection systems use the data gathered by two satellites; Advanced Very High Resolution Radiometer (AVHRR) which was launched in 1998 and Moderate Resolution Imagining Spectroradiometer (MODIS) which was launced in 1999 [6]. The satellites provide a complete image of the Earth every 1 to 2 days. This long scan period is unacceptable in forest fire detection case. Also, it should be noted that the smallest detectable fire size is accepted as 0.1 hectare and fire location accuracy is 1 km; however the accuracy and reliability of the satellites are highly affected by clouds and rain which can increase the location accuracy of the satellites to hundreds of kilometers. For a satellite system, in order to be successful in detecting forest fire, the satellite has to focus on a single forest which is not the current practise due to several reasons.

As a promising alternative, wireless sensor networks (WSNs) are an emerging technology which consists of small, limited powered and low-cost devices that have the capability of computation, sensing and wireless communication [17]. Environment monitoring is one of the most appealing areas of wireless sensor networks. Wireless sensor nodes that are deployed to various locations in a forest can collect temperature, humidity and barometric pressure values and deliver this highly important data to the sink without requiring a manual control at the control center. However, the limited energy resources of the sensor nodes and the though environmental conditions can hinder the success of forest fire detection system that is based on wireless sensor nodes.

The most critical issue in a forest fire detection system is immediate response to the fire threats in order to reduce the scale of the disaster. This requires constant surveillance of the whole forest area. Considering the deficiencies of camera and satellite based systems and the fact that WSNs are very promising as an alternative technology, in our work, we decided to study the use of WSNs for forest fire detection and monitoring. We propose a WSN architecture and related protocols that will enable rapid detection of forest fires while consuming energy cautiously in times when there is no fire. Hence, our proposed design not only aims detecting the forest fire effectively and quickly, but also considers the energy limitations of the sensor nodes. In our system, except for the periods of forest fire, the sensor nodes mostly work under regular day conditions. That is, sensor nodes, will not consume much energy while the environmental conditions are normal and there is no fire. A distributed protocol is used to run in each sensor node to consider the fire threat cautiously and in case of an abnormal temperature change, inform the control center about the possibility or occurrence of fire rapidly.

The remainder of this thesis is organized as follows. Chapter 2 discusses related studies on forest fire detection with wireless sensor networks. Chapter 3 describes the proposed method that includes four major components: deployment of sensor nodes, the network architecture, the intra-cluster communication protocol and the inter-cluster communication protocol. Chapter 4 presents our simulation environment and evaluation results. Finally, Chapter 5 concludes this thesis with a discussion on future work.

Chapter 2

Related Work

During the last decade, a considerable number of studies have been carried out regarding the involvement of WSNs in early detection problem of forest fires. Doolin and Sitar, performed several experiments through controlled fires in San Francisco, California [11]. Their system is composed of 10 sensor nodes with GPS capability. The sensor nodes are deployed with ranges up to 1 kilometer and they sense and forward temperature, humidity and barometric pressure data to a base station. The most important feature of this study is that Doolin and Sitar have implemented the system and gathered real observations from the field. However, because of the long distance between sensor nodes, the data aggregated in the sink was not valuable enough to detect a fire and forecast the spread direction of the fire. Also, with the growth of fire and burning out some of the sensor nodes, the sensor network had failed to propagate the data.

Lloret et al. used Wireless Local Area Network (WLAN) technology for the aim of fire detection [18]. Their system mixes multisensor nodes with IP cameras in a wireless mesh network in order to detect and verify fire. When a fire is detected by a wireless multisensor node, the sensor alarm is propagated through the wireless network to a central server on which a software application runs for selecting the closest wireless cameras. Then, real time images from the zone is streamed to the sink. Combining sensor data with images is the most important contribution of that study.

Hartung et al. presented a multi-tiered portable wireless system for monitoring environmental conditions, especially for forest fires [14]. Integrating web-enabled surveillance cameras with wireless sensor nodes, they provide real time weather data from the forest. In that study, three different sensor networks were deployed to different parts of a forest and the communication between the networks was provided by powerful wireless devices that can send data up to 10 kilometers range. The objective of their study is to determine the behavior of fire rather than its detection. It consists of a WSN that is used to measure weather conditions around the active fire. Webcams are also used to get visual data of the fire zone. Data gathered from the sensor nodes and the webcams are aggregated at the base station which has the capability of providing long distance communication over satellites. Periodically, the sensor nodes measure the temperature, relative humidity, wind speed and direction. On the other hand, webcams provide continuous data to the sink. Regarding the studies [11], [18] and [14], it is seen that sensor nodes are deployed to large distances from each other and sensors are supported with visual data gathered by cameras. On the other hand, our proposed system considers a denser deployment with shortened distances among sensor nodes, which will help detection of fires rapidly and propagation of valuable data to the center regularly.

Son et al. proposed a forest fire surveillance system in South Korea in which a dynamic minimum cost path forwarding protocol is applied [12]. After gathering the data, the sink node makes several calculations regarding the relative humidity, precipitation and solar radiation data, and produces a forest fire risk level. Rather than making calculations only at the sink, we propose to make local computations in the cluster-heads (i.e., in some special sensor nodes) and in this way the sink node gathers filtered data. Also, Son applies a minimum cost path forwarding method that causes some sensor nodes (especially the ones that are closer to the sink) to consume their energy much faster than the others. Our system, on the other hand, aims a low and fair energy consumption strategy, and the data propagating protocol is based on regular intra and inter cluster communication which takes the remaining energy level of the sensor nodes into account.

Yu and Wang proposed a model which applies neural network methodology for in-network data processing in environmental sensing applications of WSN [13]. Several data fusion algorithms are presented in that study. Maximum, minimum and average values of temperature and humidity data are calculated by the cluster-heads. Data is propagated to the sink if only it is worth sending (i.e., exceeding a threshold). However, since the main focus of the study is data aggregation methods, energy consumption and forecast capability issues are not discussed.

Ngai et al. proposed a general reliability-centric framework for event reporting in WSNs which is also applicable to forest fire detection systems [16]. They consider the accuracy, importance and freshness of the reported data in environmental event detection systems. They present a data aggregation algorithm that focuses on filtering important data and a delay-aware data transmission protocol which has the aim of forwarding accurate data rapidly to the sink.

Wenning et al. presented a proactive routing method for WSNs to be used in disaster detection [15]. The routing protocol's main contribution is being aware of the node's destruction threat and adapting the routes in case of a sensor death. The method adapts it routing tables based on the possible failure threat due to the sensed phenomenon.

Hefeeda and Bagharei presented a WSN for forest fire detection based on the Fire Weather Index (FWI) system which is one of the most comprehensive forest fire danger rating systems in USA [19]. This system determines the risk of propagation of a fire according to several index parameters. In the study of Hefeeda and Bagharei, weather data is collected by the sensor nodes, and the data collected at the center is analyzed according to FWI. A distributed algorithm is used to minimize the error estimation for spread direction of forest fire.

Garcia et al. proposed a simulation environment called Equipment Destined for Orientation and Security (EIDOS) [20]. This platform creates a model of the fire by analyzing the data sensed by the sensor nodes and the geographical information of the area. The feature of using topography of the environment distinguishes that study from the other solutions presented. The spread estimation of fire is sent to handheld devices of fire fighters to help them in the fight against the fire in field. Considering the studies [12], [13], [16], [15] and [19], it is seen that a single aspect of environmental monitoring is handled. However, in our proposed system, both energy and early detection goals are taken into account with overseeing the environmental obstacles.

Chapter 3

System Overview and Design Goals

Wireless sensor nodes and networks have unique features which provide many advantages and challenges in their application to forest fire detection and monitoring. Limited power resources, vulnerable node structures and harsh environmental conditions should be taken into account while constructing a solution for forest fire detection via the use of wireless sensor networks. Considering the wild forest conditions which complicate the installation of the network and the limited resources of the wireless sensor nodes, the following are some of the design goals that are important to satisfy in order to install and operate a successful network:

- consuming energy in an efficient and load-balanced manner,
- detecting the forest fire as early as possible,
- forming a network structure that will be adaptive to various environmental conditions, and
- forecasting the spread direction and speed of the forest fire.

These system goals are elaborated in the following chapters.

3.1 Energy Efficiency

Since sensor nodes have limited power resources, a wireless sensor network to be deployed for the aim of forest fire detection should have a highly efficient and load balanced energy consumption strategy. Sensor network deployment area is usually very large (measured in dozens of kilometer square) and the environment has harsh conditions which can negatively affect the sensor nodes. The sensor nodes work via batteries, and therefore changing the batteries of the nodes in these circumstances or replacing a dead sensor node with a new one will be a very costly action. In order to ease the maintenance of the system, changing the batteries of all sensor nodes at once or replacing all sensors with new ones will be more feasible. As a result, the sensor nodes should have an efficient energy consuming strategy.

Additionally, sensor nodes may malfunction on the occurrence of a forest fire, when they are exposed to high temperature. If a sensor node does not perform its regular tasks in the network for a specific amount of time, this might be considered as a possibility of fire. While many other wireless sensor network protocols like LEACH, SPIN or TEEN [21] try to create a new message forwarding path in case of a sensor node death, in our case, this is a serious indicator of fire and different precautions should be taken. The network energy consumption should be distributed evenly by considering this feature, in order to minimize the chance of malfunction of a sensor node due to energy exhaustion. In order not to cause false-alarm situations regarding the death of sensor nodes, the possibility of dying of a sensor because of energy consumption should be minimized. As a result, fair energy consumption should be obtained throughout the network. In short, in a wireless sensor network designed for forest fire monitoring, the energy consumption should be as low as possible, and the energy consumed by different sensor nodes should as balanced as possible.

3.2 Early Detection

The very early minutes of a forest fire is the most important time duration for a successful fire detection system. The forest fire grows exponentially and it is crucial that the fire should be interfered in the first 6 minutes [4]. The success of the fire detection system is related with the achievement of small fire detection time. Also, the ignition location of the fire should be identified with small error margins so that fire fighter center can intervene to the most convenient location of the forest.

We propose a sensor deployment scheme and a network architecture which will act as fast as possible in case of a fire event in a forest and make the sink aware of the fire danger and the location of the ignition place in the forest.

3.3 Forecast Capability

Forecasting the progress of forest fire is another important issue. Forest fires spread very quickly and the fight against forest fires requires accurate and fresh data. Temperature and humidity values from critical zones should be propagated to the sink node as soon as possible. And then the sink node at the center can perform the necessary calculations for forecasting the spread direction of the forest fire rapidly. After making the forecast, the sink node should be able to order the cluster-heads in the critical areas to be more active (send data more frequently to the center) and the ones in non-critical areas to be less active. Even though, proposing the final forecast algorithm is out of scope of this work here, we aim the following regarding the fire forecast capability to aid the forecasting algorithm that will run in the center.

- Provide only required data to the sink node that will be worthwhile when making a forecast,
- Propagate the critical data as soon as possible.

3.4 Adaptiveness To Harsh Environment

Robustness of the system depends on the ability of the network protocols to recover from node damages and link errors caused by harsh environmental conditions. Different from indoor applications, in environment monitoring applications, the wildlife objects and conditions (i.e., animals, humans and weather circumstances) are effective on the success of WSNs. For our specific problem, the conditions are even harder since extremely high temperature values will destroy the sensor nodes. When these circumstances are considered, we can envision that the probability of malfunctioning of a sensor node is quite high.

Another important point is that, for the sake of less and balanced energy consumption goal, the pattern of sensor node deployment may be important. We propose to have a regular and homogenous deployment of sensor nodes. However, in real deployment scenarios, this may not always be possible. While the deployment plan is constructed by the system, there could be some places, such as a lake or a swamp, where sensor nodes can not be placed. Therefore some sensor nodes will have to be deployed to different and distant locations from other sensor nodes. Considering the harsh environmental conditions, the proposed fire detection network should have the following features:

- The death of a sensor node should not affect the functioning of the whole system. Especially the death of a cluster-head node should be carefully considered and handled.
- The system should allow deploying some sensor nodes to distant locations and those sensor nodes should operate with the same functions as the other sensor nodes. Also, the energy of those sensor nodes should be kept at similar levels with the sensor nodes which are deployed regularly and homogenously.

Chapter 4

Proposed Fire Detection Framework

Our study aims to propose a comprehensive framework that considers all the four basic goals of fire detection 1) low energy consumption, 2) early detection, 3) adaptiveness to harsh environments, and 4) capability of forecasting fire spread.

Our proposed framework involves the design of four main parts: a sensor deployment scheme, a clustered network architecture, an intra-cluster communication protocol and an inter-cluster communication protocol. Regarding sensor deployment, we inspect how the sensor nodes should be deployed to a forest. In the section related to the network architecture, our clustered network architecture and hierarchy is specified. Following that, the communication scheme that is applied between the ordinary sensor nodes and the cluster-heads, and the communication scheme applied among the cluster-heads are described in detail. Next, we describe the design of each of these parts in more detail.

4.1 Sensor Deployment Scheme

How sensor nodes are deployed is an important factor that affects all aspects of the system. In our proposed system, the goals of the sensor node deployment phase are the followings:

- The distances between sensor nodes should be similar to each other, so that the nodes consume nearly the same amount of energy.
- The deployment of sensor nodes should try to minimize the chance of collisions of data packets.
- According to the importance of the region to protect (i.e., a portion of the forest that is close to a cultural heritage area) and the fire danger rate of each portion in the region, the system should be able to cover the whole region with minimum number of sensor nodes.
- In order to detect the fire as early as possible, the sensor nodes should effectively cover the forest.

In a deployment scheme, there are two major decisions to be given: the distances between the sensor nodes and the deployment pattern of the sensor nodes (i.e., a regular pattern or an irregular pattern, etc.). While making decisions for a sensor deployment scheme, the effects of energy consumption model and early detection goal should be taken into account.

In case of regular and homogenous deployment, sensor nodes will send their messages to similar distances and this will lead to equal energy consumption throughout the network. In the non-homogenous deployment case, however, some sensor nodes will have to send their messages to long distances. Since the energy consumption increases exponentially with the distance, those sensor nodes that have to transmit to long distances will run out of energy earlier. Configuring sensor nodes to send to long distances will also increase interference and collision probability in the network. This may require a heavy-weight collision avoidance and detection mechanism which will increase the energy consumption in the sensor nodes further.

Another design parameter is the distance between sensor nodes which is a critical value affecting the success of a fire detection system. Sensor nodes can measure the temperature and humidity values at specific locations they are installed. In order to detect a fire that started at a distant location from a sensor node, the heat of the fire should arrive at the sensor node's location, and obviously, a large distance between the fire ignition location and the sensor node's location increases the detection time of fire. Our experiments and several other studies that focus on the spread characteristics of the forest fires show that, the time required for a sensor node to be aware of the fire thread depends on the environmental conditions like the fuel type of the forest, the ignition level, the slope of the location and the power of wind [2, 3]. These values should be considered while determining the distances between the sensor nodes. Nevertheless, considering the early detection goal, in order to reduce the fire detection time, the distance between sensor nodes must be kept at the lowest possible level (i.e., the density of sensor network should be high).

Towards this goal, we investigated the approach of National Fire Danger Rating System (NFDRS). NFDRS is a set of computer programs and algorithms that analyzes the behavior of forest fires and it aims to estimate the fire danger of some specific zones in North America [4]. Analyzing several inputs, the system produces the fire behavior of a forest. NFDRS calculates the spread component (SC) value of a forest which represents the forward rate of spread of a head fire in meters per minute. Spread component is calculated by investigating the fuel model characteristics of the forest; live and fuel moisture value, wind speed and slope of the zone. These values increase the speed of the spread of fire; and the higher spread component value a forest has, the faster the forest fire develops. In our system, while determining the appropriate distance between the sensor nodes, we take an importance value (I) of the forest as a parameter. As the importance of the forest area and the vulnerability of the forest fire (which is defined as SC) increase, the offered distance between the sensor nodes decreases in order to reduce the fire detection time.

Some other parameters are also taken into consideration besides the importance value of the forest. Required maximum fire detection time directly affects the distance value. Also, initial energy of the sensor nodes and the required network lifetime are taken into account. The formula used to determine the distance between the sensor nodes is:

$$\triangle D = \frac{n_i \times E \times T}{N \times I^2} \tag{4.1}$$

where

- $\triangle D$ = Optimum distance between the regular sensor nodes (in meters),
- n_i = Normalization value,
- T = Required maximum fire detection time (in seconds),
- N = Required network lifetime (in seconds),
- E = Initial energy of the regular sensor node (in Joules), and
- I = The importance value and vulnerability of the forest area.

Together with the distance between the sensor nodes, the layout choice for the deployment of the sensor nodes is also important for achievement of the early detection goal. For early detection, the closeness of a sensor node to the ignition location of the fire is the most crucial factor in a deployment choice. If the sensor nodes are deployed in a regular and deterministic manner, the chance of having a fire in such a place that is far away from the sensor nodes becomes lower. In our specific application, the worst case should be considered; in other words, the possible longest distance between the fire ignition location and the nearest sensor node should be considered for alternative deployment models. In regular deployment case, two popular layout models are preferred by researches, square and hexagonal shapes [18]. In square model, 4 ordinary nodes are placed at each corner and cluster-head remains at the center. In this case, the maximum distance between the fire and the closest sensor node will be $\frac{a\sqrt[2]{2}}{4}$, where a is the distance between two corners. In hexagonal model, 6 ordinary sensor nodes are placed at the corners and cluster-head is placed at the center and in this case, the maximum distance will be $\frac{a}{2}$. Sample models of 4 and 6 nodes are shown in Figures 4.1 and 4.2. A comparison between the two models should consider the total number of sensor nodes required for covering the forest, the energy consumption level and the initial energy level of each sensor node. However, this design choice between 4 or 6 nodes in a cluster is not in the scope of our study. In our experiments, we use the square shaped model. On the other hand, in irregular and heterogeneous deployment case, we can not guarantee a maximum distance level between a sensor node and fire ignition location. Therefore, on the average, the distance between the closest sensor node and the fire ignition location is lower in regular deployment scheme.

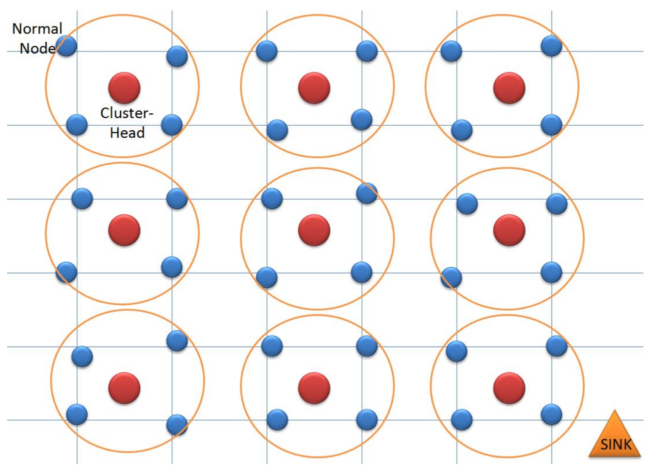

Figure 4.1: A sample network architecture of a cluster with 4 nodes

Considering the real life conditions, it is sure that in some cases it will not be possible to deploy all sensor nodes in a regular grid shape. There will be some nodes which will have to be deployed to distant locations from other sensor nodes. Those sensor nodes will have to send their messages to longer distances and therefore will consume more energy than the other nodes. In order not to ruin the balanced energy consumption strategy applied in the whole network, the sensor nodes that will transmit to longer distance should have higher initial energy levels.

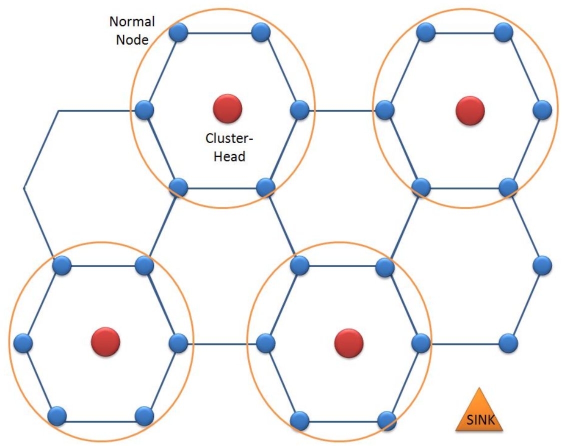

Figure 4.2: A sample network architecture of a cluster with 6 nodes

For the sensor nodes that have to be deployed to far distances unlike the regularly deployed sensor nodes, we determine the initial energy level via the following formula;

$$E_{ext} = E \times \left(\frac{D_{ext}}{\triangle D}\right)^2 \tag{4.2}$$

where

- $\triangle D$ = Average distance between regular sensor nodes (in meters),
- D_{ext} = Distance of a sensor to the closest neighbor sensor node in the extreme case (in meters),
- E = Initial energy level of regular sensor nodes (in Joules), and
- E_{ext} = Initial energy level of the sensor nodes that are not deployed regularly (in Joules).

As seen in Figure 4.3, the node that is located to a distant location will be deployed with higher initial energy. This feature of the system will provide each sensor node to have the similar energy level throughout the network and its lifetime.

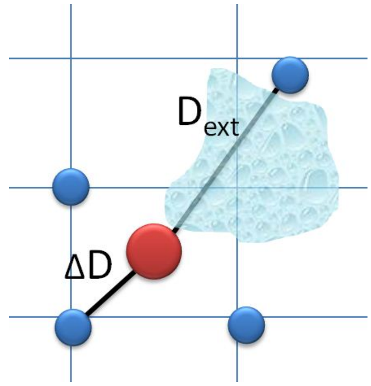

Figure 4.3: Placing the nodes to distant locations due to environmental conditions

4.2 Network Architecture and Topology Design

The architecture and logical topology of the network should be designed considering the goals of a fire detection system and limitations of wireless sensor nodes. The main focus of the network architecture depends on various environmental conditions where the network is deployed. In regular times, when there is no fire and the risk of fire is quite low, the network system should aim to decrease the message overhead throughout the network and the data should be forwarded to the sink with minimum cost, so that less energy is consumed at sensor nodes. However, while considering the energy limitations, the goal of detecting fire as early as possible should not be compromised.

In a possible fire threat time, as the fire spreads, many sensor nodes will sense the threat and each sensor node will try to send their own local critical information to the sink many times. However, the sink node will not need to get these critical messages over and over again. After being aware of the start of the fire, the new focus of the network should be trying to figure out the development of the fire. At this time, sink requires to get data which will be helpful in forecasting the spread direction of the fire such as the number of newly dead sensors since the last period in a cluster.

Therefore, the actions of sensor nodes and the decision of which goals are the most important at a given time are highly dependent on environmental and weather conditions, as well as whether there is fire or not at that time. Our architectural design considers this and offers a clustered logical topology for the network to properly and adaptively control the sensor nodes under various conditions that we may have in a forest.

For the clustered architecture we propose, a specific number or percentage of sensor nodes (where this depends on some system parameters) will form a group (a cluster) and connect to a cluster-head which will have additional responsibilities. The cluster-heads may have superior physical capabilities, such as higher memory and computational power. An example illustration of the cluster hierarchy is shown in Figure 4.1.

When the cluster-heads are determined, before deployment, the fire danger rate table, which contains the specific features of the environment will be installed to the cluster-heads. Also, in same cases, cluster-heads may have GPS (Global Positioning System) capability so that they can send the location information together with the environmental data. They should also have the capability to adjust their transmit power to transmit to longer distances when necessary.

In our architecture, cluster-heads perform critical roles in the network: aggregating temperature and humidity data from member nodes, determining the sleeping ratio of the child (member) sensor nodes, managing the child nodes in fire danger time so that only critical data will be forwarded to the sink, and forwarding cluster report messages to the sink. Clustered hierarchy is favorable for both early detection and energy conservation. However, assigning some critical responsibilities to specific nodes (cluster-heads in our case) increases vulnerability of the system. To make the system more robust, a dynamic cluster-head selection mechanism could be applied in a possible death scenario of a cluster-head, but this is out of the scope of our study here.

There are two possible alternatives for the network topology. Either the sensor nodes completely run in a distributed manner and each sensor node individually acts in the network, or a clustered hierarchy is implemented by designating some cluster-heads which control the ordinary sensor nodes. We performed several tests and decided that the use of a clustered topology is better. There are three important reasons for this decision:

Data fusion: Data fusion is a well-known method in which the cluster-heads aggregate messages from child nodes and construct a single message which leads to less message overhead in the network. Forest fire detection application is very suitable for data fusion. Consider a network topology in which a number of ordinary nodes, lets say 4 nodes, are sending `RegularInform` messages to the cluster-head in every 5 minutes and each message contains temperature and humidity values. The cluster-head will gather 48 messages in an hour and by applying an aggregation function to those messages, it can construct a single and more meaningful packet to be delivered to the sink which will dramatically decrease the load over the network.

Balanced energy consumption: This is a very critical goal to be achieved in wireless sensor network applications. Especially in environmental monitoring applications where the messages are gathered at one control center, the sensor nodes that are closer to the sink node will consume more energy since more packets are forwarded through them comparing to the nodes that are far away from the sink. As seen in Figure 4.4, the sensor nodes which are closer to the sink will consume more energy. In our application, sensor nodes send regular information messages to the cluster-head and cluster-heads send a cluster-wise information package to the sink. Cluster-heads also collect messages from other cluster-heads and applying a special message forwarding time table, each cluster-head sends message in each period. As a result, regardless of whether the nodes are close to or far away from the sink, each node consumes similar amount of energy.

Less messaging overhead: Providing only the necessary and required data to the center not only prevents unnecessary traffic throughout the network, but also simplifies the data processing at the center by eliminating unnecessary data. After gathering data from the regular nodes, cluster-heads make a local computation based on the data coming from their own children. In fire time, rather than continuously sending temperature and alarm messages, a cluster-head makes evaluations for all children by investigating the temperature and humidity data

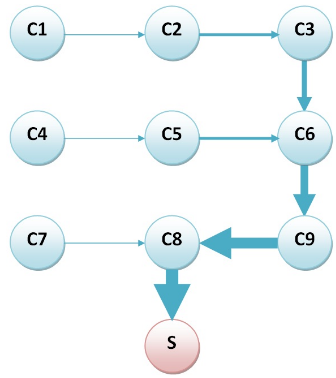

Figure 4.4: Nodes closer to the sink forwards more message

and searching for any threat or dangerous situation. Basically cluster-heads look for the existence of any node having abnormal temperature or humidity increase that shows difference from the other children. For this, a cluster-head monitors the following information:

- Number of sensor nodes that are in fire,
- number of sensor nodes that are close to fire,
- number of sensor nodes that are not sensing any threat, and
- number of dead sensor nodes.

Based on this data that is periodically obtained, the cluster-head compares the previous and next such data and derives the following information:

- Number of sensor nodes that were not in fire in previous time frame and now in fire,
- number of sensor nodes that were not sensing any threat but now sending fire threat messages, and
- number of alive and dead sensor nodes.

The cluster-head derives this valuable data and sends a single packet to the sink node which will be helpful in forecasting the forest fire spread. A sample illustration of a forest fire detection and the communication between the nodes is shown in Figure 4.5.

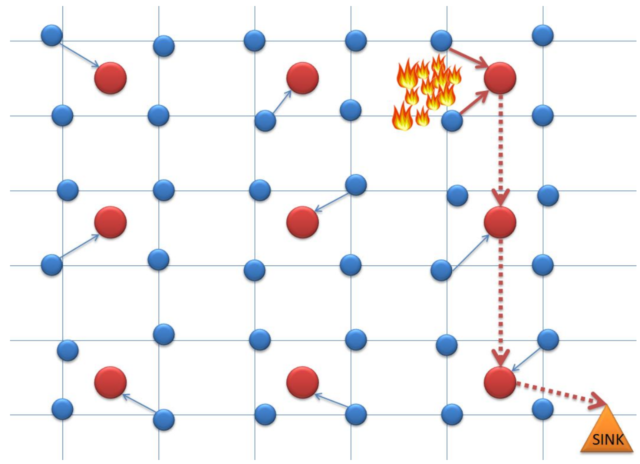

Figure 4.5: Detection of fire event and message forwarding among the nodes

4.3 Environment Aware Intra-Cluster Communication Protocol

The communication scheme between sensor nodes and cluster-heads consists of 4 phases: initialization phase (which involves defining message sending sequence), risk-free time (regular time) phase, fire threat (fire time) phase and progressed fire phase. The messages transmitted between a cluster-head and the child nodes are listed in Table 4.2 where phase 0 is initialization, phase 1 is risk-free time and phase 2 is fire threat phase. The complete list of the states and transition between the states of the regular nodes and the cluster-heads are mentioned in Tables 4.3, 4.4, 4.5 and 4.6. Also, in the Figures 4.6 and 4.7, the life cycles of regular nodes and the cluster-heads are presented.

Initialization phase: In this phase, cluster-heads send an advertisement message `ClusterConnAdv` in order to make child nodes to connect to them. As mentioned in the sensor deployment scheme, the sensor nodes are distributed in a regular manner, such that each cluster-head has same amount of sensor nodes. When the child sensor nodes hear the announcement from the cluster-heads, they

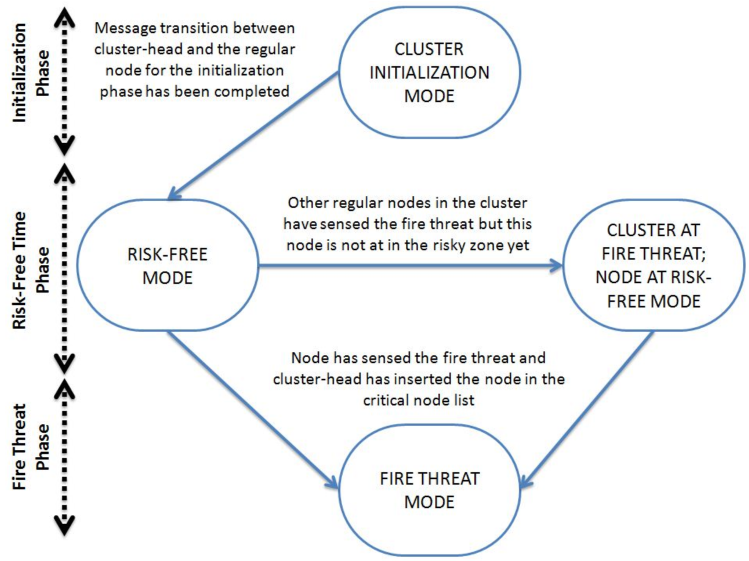

Figure 4.6: State transition diagram of a regular node

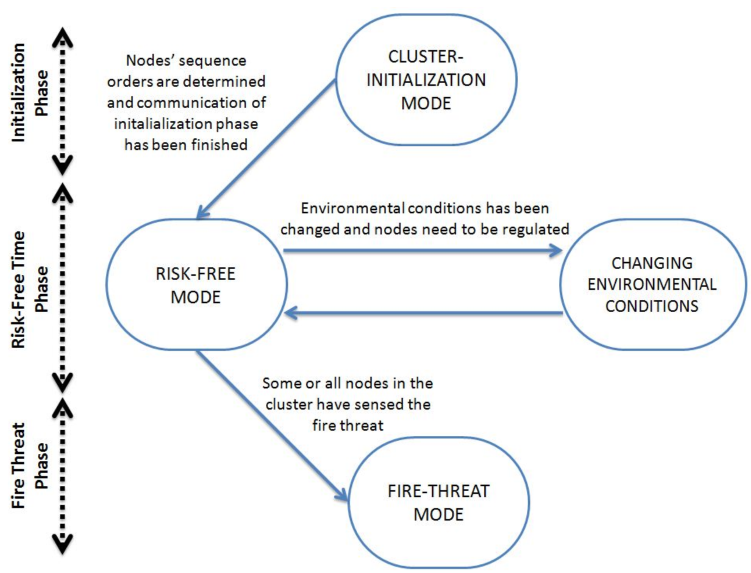

Figure 4.7: State transition diagram of a cluster-head

reply with a connection request, `ClusterConnReq`. Cluster-heads are deployed with some initial settings; i.e., the number of children node numbers are set to each cluster-head before deployment. Cluster-heads send periodic announcement messages as long as the number of nodes that send back a reply is less than the expected count. The initialization phase continues for a specific time duration and the phase ends when the number of children nodes reach the expected number of child nodes.

After the initialization phase, the cluster-head assigns a message sending sequence for each sensor node which will be used to coordinate the access to the shared wireless channel. The frequency of sensor nodes to send regular inform messages to the cluster-head depends on a variable which is related with the current fire danger rate calculated by the cluster-head which indicates the fire risk of that location. For example, if the cluster-head is located in a critical zone or the current season is summer, the fire danger rate has a higher value. The higher the fire danger rate the cluster-head has, the more often the children nodes send regular inform message. The cluster-head informs the children about the message sending frequency (i.e., the duration between two transmit events). Hence, the cluster-head sends a sequence number and a message sending duration to each sensor node. Also, the fire threshold levels are sent to the regular nodes by which nodes can determine the risk of fire. Gathering all the required parameters from the cluster-head, regular nodes pass to the risk-free time phase.

Risk-free Time Phase: During the times when fire risk is low, the main aim of the protocol is to assure that less energy is consumed, but nevertheless the sensor nodes are kept awake from time to time so that they can sense the possible fire threat. The sensor nodes listen to the environment with a period which is controlled by the cluster-head. When its turn comes, a child sensor node reports six different values to its cluster-head: the minimum, maximum and average temperature and humidity values sensed during the last time frame.

Intra-cluster communication considers energy efficiency and therefore the activity level of sensor nodes is made dependent to the environmental conditions. In other words, the frequency of sending messages and sensing environment is set to a lower value in non-risky weathers. In cold and rainy weathers, when the temperature is low and humidity is high, the risk of fire is very low so that it is not necessary for sensor nodes to be too sensitive. On the other hand, if the temperature is high and humidity is low, the risk of having a fire is quite high and therefore sensor nodes are set to be more active (i.e., the frequency of sending messages and sensing environment is increased) in those conditions.

Ordinary sensor nodes can be forced to sleep in order to save more energy. The sleeping mechanism is constructed by considering the following parameters; pre-defined importance level of the zone, fire danger rate table (see Table 4.1 as an example), fire danger rate level computed by the cluster-head, current energy level of each sensor node in the cluster, and the target network lifetime of the system. The cluster-head regularly checks the most recent values of those parameters and depending on those values, it applies a sleeping mechanism to its children sensor nodes. The basic principle is to set each child node into sleep in a sequence so that each sensor node maintains a similar remaining energy level. When the cluster-head decides to put a specific sensor node into sleep mode, the sensing frequency of that sensor node is set to be a very low value.

Table 4.1: Fire danger rate table for Kemer, Antalya, Turkey

Temperature Range	**Humidity**	**Danger Rate**
38 - 41	0,03 - 0,05	55
38 - 31	0,01 - 0,03	57
38 - 31	0,005 - 0,01	62
41 - 34	0,03 - 0,05	65
41 - 34	0,01 - 0,03	67
41 - 34	0,005 - 0,01	74

Fire threat phase: The main goal of our system is to detect the fire threat as early as possible, therefore the intra-cluster communication protocol has effective control mechanisms regarding fire threat. As mentioned in the initialization phase, each child sensor node is given a specific time slot to send the temperature and humidity values to the cluster-head. Also, the cluster-head sends temperature and humidity threshold levels to the child sensors to indicate the fire threat. When the temperature or humidity level exceeds the critical level, fire threat phase starts.

Depending on the regular message sending time, a sensor node may find the chance to send messages in every 10-15 minutes. Regarding the goal of early detection in which we aim the sink to be aware of the fire within 6 minutes, this message sending time will not be acceptable in case of a potential forest fire. A variable is set in each sensor node to set the maximum time duration in a potential fire threat to wait its own sequence to send the message. When a sensor senses a temperature and humidity level indicating a fire threat, it decides whether to interrupt the order in cluster or not. The sensor node that senses the fire threat sends `FirstFireThreatAlarm` message in such a case until the cluster-head sends back a new announcement message.

The cluster-head hearing `FirstFireThreatAlarm` message, takes several actions. Actions are performed according to the current phase of the cluster-head. If the cluster-head doesn't have any child sensor node which is in a critical zone, in other words, if this is the first time that a child sensor node is getting into a critical zone, the cluster-head sends back an `OnlyNodesThatSensedFire` message for re-arranging the message sending sequence. The aim of this message is to allow the nodes that currently sense the fire threat to send temperature data more frequently. For example, if there are 10 nodes in a cluster and 2 of them are sensing fire threat, a window frame of size 4 is created and 2 time frames are allocated to 2 fire-hearing nodes. The remaining 2 time frames are allocated to the nodes that may possibly hear fire. If a node senses fire at a further time, it sends `FireThreatListReq` message and the time frames are re-allocated.

Moreover, the data aggregation method varies according to the current risk level of the cluster-head. Rather than sending average and min - max values of temperature and humidity, more data that will be helpful for the sink to analyze the progress of the fire can be sent. The number of alive and dead sensor nodes, the change in the number of alive and dead sensor nodes, and the number of sensor nodes that sense fire can be forwarded to the sink. Especially the difference between the number of alive and dead sensor nodes in time-interval is important and it indicates how serious the fire threat is in that area.

Progressed fire phase: The chance of a dying of a sensor node is quite high in our area of application compared to other WSN applications because of the environmental conditions during fire occurrence. When the role of cluster-heads is considered, the system should be prepared for an incident where a cluster-head may not be able to perform its critical duties. There are two different conditions regarding this problem; whether the cluster-read recognizes its own death or the cluster-head suddenly dies.

In the first case, the cluster-head discerns the potential risk and selects the most suitable sensor node as the new cluster-head. When the cluster-head gathers current temperature information from its child nodes, it can guess that it will be subject to extremely high temperature after a certain time. At this point, in order not to run the cluster into a chaos, the cluster-head broadcasts a `CHDeathAlarm` message regarding this situation and asks for each child node to return their remaining energy level, current temperature and humidity values and increment value of temperature and humidity data. Gathering those values, the cluster-head finds the most appropriate regular node with the following formula;

$$r_i = \frac{T_d - t_i}{\triangle t_i} \times e_i \tag{4.3}$$

where,

- r_i = Risk level of the node i
- T_d = Death temperature level of regular nodes

- t_i = Current temperature level of the node
- $\triangle t_i$ = Increment ratio of the temperature of the node for the last period
- e_i = Current remaining energy level of the node

Then the sensor node with least fire risk is selected as the new cluster head and the following information is forwarded to the next cluster:

- regular sensor node list,
- regular sensor node message sending time frames,
- last temperature and humidity values of children,
- path of inter-cluster communication (previous and next cluster head in the inter-cluster communication scheme), and
- other parameters that cluster head should know (i.e., message sending duration, threshold values, etc.).

After having these data, the new cluster-head broadcasts `NewClusterHeadInfo` message to the other regular nodes in the cluster. Also it informs the neighbor cluster-heads about the situation and therefore cluster-heads send their cluster-wise data messages to the new cluster-head.

In the second case where because of a quick temperature increment or another weather incident like a lightning, the cluster head may suddenly die before selecting a new cluster-head and forwarding the required critical information to it. In order not to lose these valuable information, at the start of fire-thread phase the cluster-head sends `CHCriticalInfo` message to its child regular nodes. However, the recognition of the death of the cluster-head and electing a new cluster-head by the regular nodes require a special mechanism since there won't be a cluster-head to organize the sensor nodes and make a selection.

As mentioned in the risk-free time phase, the child sensor nodes of the cluster-head periodically send regular inform messages to the cluster-head and regarding to this message, the cluster-head returns an acknowledgement message named `RegularInformACK` to the corresponding sensor node. If a regular node cannot get an acknowledgement message from its cluster-head, the node re-sends its inform message. Different from the regular messages, the second inform message is also received by the other nodes in the cluster in order to make aware of the nodes about the situation that the cluster-head is not responding the messages.

All the sensor nodes receiving this message wait for the response of the cluster-head and if the cluster head doesn't send a reply message to the `Re-RegularInform` message, this time the sensor node having the next message sending order sends its `RegularInform` message to the cluster-head. The number of trials of the same routine is given as a system parameter. Even if at the last trial, the cluster-head doesn't send a `RegularInformACK` message, then it is decided that the cluster-head is dead.

The sensor node which decides that the cluster-head is dead sends an `CHIsDead` announcement message indicating the situation. After hearing this message, sensor nodes send an advertisement message `CHNominee` that contains their own remaining power level and the fire danger rate. During this period, each sensor node receives the neighbor nodes' values and using the Formula 4.3, the most appropriate sensor node selects itself as the new cluster-head and sends an announcement message `NewCH` indicating that it is the new cluster-head.

Table 4.2: Messages transmitted between a cluster-head and its child nodes

Phase	**From**	**To**	**Message**
0	CH	Node	ClusterConnAdv
0	Node	CH	ClusterConnReq
0	CH	Node	ClusterConnResponse
1-2	CH	Node	NodeCoreData
1	Node	CH	RegularInform
1-2	CH	Node	RegularInformACK

Continued on next page

Table 4.2 – continued from previous page

Phase	From	To	Message
1-2	Node	CH	FirstFireThreatAlarm
1-2	CH	Node	OnlyNodesThatSensedFire
2	CH	Node	CHCriticalInfo
2	Node	CH	FireThreatListReq
2	CH	Node	FireThreatNodeList
2	Node	CH	LastMessage
2	Node	CH	FireThreatCHRegularInform
1-2	Node	CH	NeighborAtRisk
2	CH	CH	FireAlarm
1-2	CH	CH	ClusterAverageData
3	CH	Node	CHDeathAlarm
3	CH	Node	NewClusterHeadInfo
3	CH	CH	NewClusterHeadInfo
3	Node	CH	Re-RegularInform
3	Node	Node	CHIsDead
3	CH	Node	CHIsDying
3	Node	Node	CHNominee
3	Node	CH	CHNominee
3	Node	Node	NewCH
3	CH	Node	NewCH

Table 4.3: States of a regular node

State	Explanation
N-N-1	Waiting for cluster-head initialization message
N-N-2	Cluster-head connection request sent
N-N-3	Waiting in-cluster information package
N-N-4	Risk-free time regular actions

Continued on next page

Table 4.3 – continued from previous page

State	Explanation
N-N-5	Cluster-head regular informing
F-N-6	Node at critical level
F-N-7	Wait and send alarm message
F-N-8	Interrupt the cluster regular order and send alarm message
F-N-9	`OnlyCriticalNodes` message comes from cluster-head
N-N-10	Neighbor cluster is at risk
N-N-11	Node is not at risk and fire-threat node list message has been received
F-N-12	Node is at risk and fire-threat node list message has been received
F-N-13	In fire-threat mode
F-N-14	Cluster is in fire and node is not in fire-threat phase
F-N-15	Cluster-head critical informing
F-N-16	Cluster-head non-critical informing
F-N-17	Node has just sensed fire threat
N-N-18	Environmental data update
F-N-19	Cluster-head is not responding
F-N-20	Cluster-head is dying
F-N-21	Selecting new cluster-head
F-N-22	New cluster-head is decided
F-N-23	Cluster-head is deciding the new cluster-head
F-N-24	New cluster-head is selected

Table 4.4: States of a cluster-head

State	Explanation
N-C-1	Broadcasting ClusterConnAdv
N-C-2	Accepting connection requests from nodes
N-C-3	Setting in-cluster orders of nodes

Continued on next page

Table 4.4 – continued from previous page

State	Explanation
N-C-4	Risk-free time regular actions
N-C-5	Cluster-wise temperature level calculation
N-C-6	Sending cluster-wise information package to next cluster-head
N-C-7	Receiving neighbor cluster-heads information packages
F-C-8	First fire alarm actions
F-C-9	Forwarding critical status to next neighbor
F-C-10	Neighbor cluster is in critical status
F-C-11	`OnlyCriticalNodes` message is sent, waiting for critical nodes
F-C-12	Determining new fire threat critical node list
F-C-13	Fire threat mode
F-C-14	Cluster-wise threat level calculation
N-C-15	Environmental data update
F-C-16	Cluster-head is about to die
F-C-17	Deciding the new cluster-head

Table 4.5: State transition of a regular node

States	Transition	Actions Performed
N-N-1... N-N-2	`ClusterConnAdv` message has been received	Node waits for `ClusterConnAdv` messages and the closest cluster-head that sends this message is picked as the cluster-head of the node
N-N-2... N-N-3	`ClusterConnReq` message has been sent to the cluster-head	The node picks its cluster-head and at the initialization message sending time, the node sends a `ClusterConnReq` message to the cluster-head

Continued on next page

Table 4.5 – continued from previous page

States	Transition	Actions Performed
N-N-3... N-N-4	`ClusterConnResponse` message has been received	Cluster-head confirms the cluster acceptance of the node and sends the required parameters to the regular node
N-N-4... N-N-5	Message sending time comes for the node	Node sends the minimum, maximum and average temperature and humidity values for the last period to the cluster-head at the message sending time
N-N-4... F-N-6	Temperature is at critical level and cluster threat level is normal	Node gets into the critical level and prepares for sending alarm message
N-N-4... N-N-10	`FirstFireThreatAlarm` message comes from another cluster	Node sends the information that a neighbor node has sensed fire to the cluster-head at the message sending time
N-N-4... N-N-18	Cluster-head a new `InClusterInfoPackage` message	New fire threat threshold levels, message sending frequency and several parameters' new values are received from the cluster-head
N-N-5... N-N-4	`CHRegularInform` message is sent to the cluster-head	Node turns to regular state and continues listening the environment
F-N-6... F-N-7	(Next message sending time - now) value is smaller than (Critical level wait duration)	The node waits for its message sending time to send the critical alarm message
Continued on next page		

Table 4.5 – continued from previous page

States	Transition	Actions Performed
F-N-6... F-N-8	(Next message sending time - now) value is larger than (Critical level wait duration)	The node continuously broadcasts the critical alarm message until `OnlyCriticalNodes` message comes
F-N-7... F-N-9	`OnlyCriticalNodes` message comes and node is at fire risk	Since the node has sensed the fire, when `OnlyCriticalNodes` message has been received from the cluster-head, the node replies with acception request for fire threat node list group
F-N-8... F-N-9	`OnlyCriticalNodes` message comes and node is not at fire risk	Since the node has not sensed the fire yet, when `OnlyCriticalNodes` message has been received from the cluster-head, the node doesn't send a reply message
F-N-9... N-N-11	`FireThreatNodeList` message comes and node is not in risk	Node gets into the passive-fire threat mode and continues listening the environment
F-N-9... F-N-12	`FireThreatNodeList` message comes and and node is in risk	Node gets into the active fire-threat mode and continues listening the environment
N-N-10... N-N-4	Cluster-head is informed about neighbor's `FirstFireThreatAlarm`	Node turns to regular state and continues listening the environment
N-N-11... F-N-14	Cluster at fire-threat mode and message sending time comes	Node is not at risk, therefore it sends alive messages less frequently

Continued on next page

Table 4.5 – continued from previous page

States	Transition	Actions Performed
F-N-12... F-N-13	Node is in fire-threat mode and message sending time comes	Node is in the fire threat node list and therefore it sends critical level information more often
F-N-13... F-N-15	Node is in critical node list and message sending time comes	Node prepares the fire threat regular information to be sent to the cluster-head
F-N-13... F-N-9	Node is in `FireThreatNodeList` and a new list comes	Node takes the new fire-threat phase in-cluster communication parameters
F-N-13... N-N-18	New `InClusterInfoPackage` message comes from cluster-head	Node takes the new risk-free time phase in-cluster communication parameters
F-N-14... F-N-16	Node is not in `FireThreatNodeList` and message sending time comes	Node prepares the fire threat regular information to be sent to the cluster-head
F-N-14... F-N-17	Node is not in `FireThreatNodeList` and it has just sensed the fire	Node waits for sending a request for acceptance to critical node list
F-N-14... F-N-9	New `FireThreatNodeList` message has been received	Node takes the new in-cluster communication parameters
F-N-14... N-N-18	New `InClusterInfoPackage` message comes from cluster-head	Node takes the new risk-free time phase in-cluster communication parameters
F-N-15... F-N-13	Node has sent `CHFireThreatRegularInform` message	Node returns to the active fire-threat mode and continues listening the environment

Continued on next page

Table 4.5 – continued from previous page

States	Transition	Actions Performed
F-N-16... F-N-14	Node has sent `CHRegularInform` message	Node returns to the passive fire-threat mode and continues listening the environment
F-N-17... F-N-14	Message sending time comes	Node sends request for acceptance to critical node list
N-N-18... F-N-14	New `InClusterInfoPackage` message comes from cluster-head	Node takes the new in-cluster communication parameters
N-N-5... F-N-19	`CHRegularInform` message has been sent but `CHRegularInformACK` message has not been received	`CHRegularInformACK` message has not been received from the cluster-head in risk-free time phase; number of trials in incremented by 1
F-N-15... F-N-19	`FireThreatCHRegularInform` message has been sent but `CHRegularInformACK` message has not been received	`CHRegularInformACK` message has not been received from the cluster-head in fire threat phase; number of trials in incremented by 1
F-N-16... F-N-19	`FireThreatCHRegularInform` message has been sent but `CHRegularInformACK` message has not been received	`CHRegularInformACK` message has not been received from the cluster-head in fire threat phase; number of trials in incremented by 1

Continued on next page

Table 4.5 – continued from previous page

States	Transition	Actions Performed
F-N-19... F-N-20	Connection to the cluster-head has been tried for `TrialNumberForCHDeath` times by different different sensor nodes however cluster-head doesn't send `CHRegularInformACK` message	`TrialNumberForCHDeath` has been reached; the node sends `CHIsDead` message
F-N-20... F-N-21	The sensor node which decides that cluster-head is dead sends `CHIsDead` message	Each sensor node sends its energy, temperature and humidity values at its own message sending time
F-N-21... F-N-22	Each sensor node sends `CHNominee` message	The sensor node with most energy and lowest fire risk selects itself as the new cluster-head and sends `NewCH` message with regulating new message sending times for the other regular nodes
F-N-13... F-N-23	`CHIsDying` message has been received	Cluster-head is about to die; energy, temperature and humidity values are sent to the cluster at the message sending time
F-N-14... F-N-23	`CHIsDying` message has been received	Cluster-head is about to die; energy, temperature and humidity values are sent to the cluster at the message sending time
F-N-23... F-N-24	New cluster-head is announced by the old cluster-head	New cluster-head is decided and new in-cluster parameters are processed

Table 4.6: State transition of a cluster-head

States	Explanation	Actions Performed
N-C-1... N-C-2	`ClusterConnAdv` message has been sent	Cluster-head has sent the `ClusterConnAdv` message and request messages are gathered from the regular sensor nodes
N-C-2... N-C-3	`ClusterConnReq` messages come from the child nodes	Message sending orders and other several parameters are determined for the child nodes and messages are sent to each one
N-C-3... N-C-4	Members of the clusters are determined	Risk-free time phase starts and environmental data messages are gathered from the child nodes
N-C-4... N-C-5	`CHRegularInform` messages come from the child nodes	Cluster-wise temperature level is calculated; the average, minimum and maximum temperature and humidity values gathered are calculated
N-C-4... F-C-14	`CHFireThreatRegularInform` messages come from the child nodes	During the fire-threat time, cluster-wise threat level is calculated
N-C-4... N-C-7	`ClusterAverageData` message comes from another cluster-head	Neighbor cluster-heads' cluster average data messages are gathered
N-C-4... N-C-10	`FirstFireThreatAlarm` message comes from a neighbor cluster-head	Without waiting the next cluster-head message sending time, the critical message information is sent to the next neighbor
Continued on next page		

Table 4.6 – continued from previous page

States	Explanation	Actions Performed
N-C-4... N-C-15	Environmental data has changed (probably a new season has started)	Cluster-head calculates new parameters for the sensor nodes regarding the new weather data (sensing frequency, message sending duration)
N-C-4... N-C-8	`FirstFireThreatAlarm` message comes from a child node	Critical messages gathered previously is analyzed and if the message is the first one gathered for a specific amount of time, fire alarm actions start
N-C-5... N-C-6	`ClusterAverageData` is calculated and message sending time comes	Cluster-wise temperature level message is sent to the next cluster-head
N-C-6... N-C-4	`ClusterAverageData` message is sent	Risk-free time phase actions are continued
N-C-7... N-C-6	Neighbor cluster-head's `ClusterAverageData` messages are processed	Aggregated data message is calculated
F-C-8... F-C-9	Message sending time comes	Fire alarm situation message is prepared for sending to the next cluster-head
F-C-9... F-C-11	`FirstFireThreatAlarm` message comes from a child node	Fire alarm situation message is sent to the next cluster-head
F-C-10... N-C-5	Neighbor cluster-head's `ClusterAverageData` messages are processed	Neighbor's fire alarm situation message is sent to the next cluster-head and regular state actions are performed

Continued on next page

Table 4.6 – continued from previous page

States	Explanation	Actions Performed
F-C-10... F-C-13	Neighbor and this cluster-head are in fire-threat mode	Neighbor's fire alarm situation message is sent to the next cluster-head and fire state actions are performed
F-C-11... F-C-12	`OnlyCriticalNodes` message is sent	Critical node list connection requests are coming and those nodes are getting into fire threat critical node list
F-C-12... F-C-13	`FireThreatListReq` messages come from child nodes	Critical nodes are accepted to the list and the list is broadcast to all child nodes
F-C-13... F-C-14	`CHFireThreatRegularInform` messages come from the child nodes	Number of alive and death sensors, maximum temperature values and average temperature values are aggregated into a message
F-C-13... N-C-15	Environmental data has changed (probably a new season has started)	Cluster-head calculates new parameters for the sensor nodes regarding the new weather data (sensing frequency, message sending duration)
F-C-13... N-C-6	`ClusterAverage` data is calculated and message sending time comes	Cluster-wise fire-threat phase temperature level message is sent to the next cluster-head
F-C-13... F-C-10	`FireAlarm` message comes from a neighbor cluster-head	Without waiting the next cluster-head message sending time, the critical message information is sent to the next neighbor
F-C-14... F-C-13	`ClusterAverage` data is calculated and message sending time comes	Fire-threat phase actions are continued

Continued on next page

Table 4.6 – continued from previous page

States	Explanation	Actions Performed
N-C-15... F-C-13	Environmental data changes are taken into consideration	Fire-threat phase actions are continued
N-C-15... N-C-4	Environmental data changes are taken into consideration	Risk-free time phase actions are continued
F-C-13... F-C-16	Temperature level is close to the death level	Cluster-head sends `CHIsDying` message to the nodes
F-C-16... F-C-17	`CHNominee` messages have been received	`CHNominee` messages have been received and the highest energy and lowest fire risk is selected as the new cluster-head; new cluster-head is announced to the regular nodes and neighbor cluster-heads

4.4 Environment Aware Inter-Cluster Communication Protocol

The cluster-head level communication scheme has two main goals; providing each cluster-head to consume similar amount of energy and forwarding critical messages to the sink node as soon as possible. The scheme consists of three phases; initialization phase, risk-free time phase, and fire threat phase.

Initialization phase: Our first aim is to construct a message sending mechanism between the cluster-heads which will allow each cluster-head to send the same amount of messages in one round. In case that such a scheme is not applied, the cluster-heads that are closer to the sink will consume more energy than the sensor nodes that are far away from the sink, since the closer nodes will always have to forward messages coming from all the other nodes.

Figure 4.8 shows an example inter-cluster communication scheme. In the network initialization phase, cluster-heads determine the paths that connect them to the sink and a message forwarding time table is constructed from these paths (Table 4.7). This table can either be defined before the sensor deployment phase and set to the cluster-heads, or be calculated dynamically after the deployment. Time frames are defined with a parametric value and during this time duration, cluster-heads continuously send the messages. Therefore, cluster-heads can gather messages at same time frame.

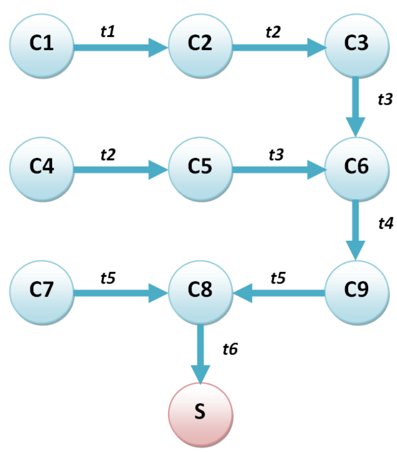

Figure 4.8: Inter-cluster communication scheme

As seen in Table 4.7, cluster-head 6 receives messages from cluster-head 5 at t3 and from cluster-head 3 at t3. Cluster-heads also apply data fusion to those messages that are received, and forward a smaller packet to the next cluster-head. When cluster-head 6 receives messages from both cluster-heads, it forwards its own data with newly incoming messages to cluster-head 9. As seen in Figure 4.8, a single round consists of 6 steps and at t6 the messages are forwarded to the sink node.

Risk-free time phase: During fire risk-free day-time conditions, the sensor nodes forward their sensed data to the cluster-heads regularly. Cluster-heads apply data fusion to the messages coming from various child sensor nodes. The data aggregation method should be designed by considering the requirements of the application. In our specific case, the maximum level of the temperature and the minimum level of the humidity are important indicators of a possible forest fire.

Also, for the sake of traceability of the forest, it will be useful for the center to be able to view the instantaneous temperature and humidity values. As a result, in risk-free time phase, cluster-heads calculate the minimum and maximum level of temperature and humidity, and send these data to the sink. Data aggregation method can also be applied at the cluster-head level communication. A cluster-head that receives an aggregated message from other cluster-heads, can apply data fusion mechanism to these data. If it is not required for the sink to view the current situation of all the network, this data aggregation will be helpful for balanced energy consuming. Because, as the cluster-heads get closer to the sink, they will have to forward more data if data fusion is not applied.

Table 4.7: Message forwarding time table

Cluster-head	**Next Cluster-head**	**Time Frame**
C1	C2	t1
C2	C3	t2
C3	C6	t3
C4	C5	t2
C5	C6	t3
C6	C9	t4
C7	C8	t5
C9	C8	t5
C9	SINK	t6

Fire threat phase: When a cluster-head receives a `FirstFireThreatAlarm` message from one of its child sensor nodes, it tries to deliver the message to the sink as soon as possible. However, it is also important to be aware about the overall conditions in the forest at that time. If only that cluster-head senses a threat, and none of the other cluster-heads have sensed a critical event, then this cluster-head is probably the first one noticing the occurrence of a fire. Therefore the event must be propagated to the sink node as rapidly as possible. If, however, there has been some other nodes (cluster-heads) that have already detected the occurrence of a fire and informed the sink about it, then this cluster-head does not need to be that fast to propagate the critical event to the sink node. In other words, this condition should be handled differently.

In order to make all cluster-heads aware of the overall fire threat of the forest, a special mechanism is applied. The cluster-heads that gather messages from other cluster-heads can easily have the last situation information (like cluster-heads 8, 6 and 9 in Figure 4.9). However some other nodes, especially the nodes that are far away from the sink, receive less messages from neighbor cluster-heads. For these cluster-heads, we offer over-hearing method.

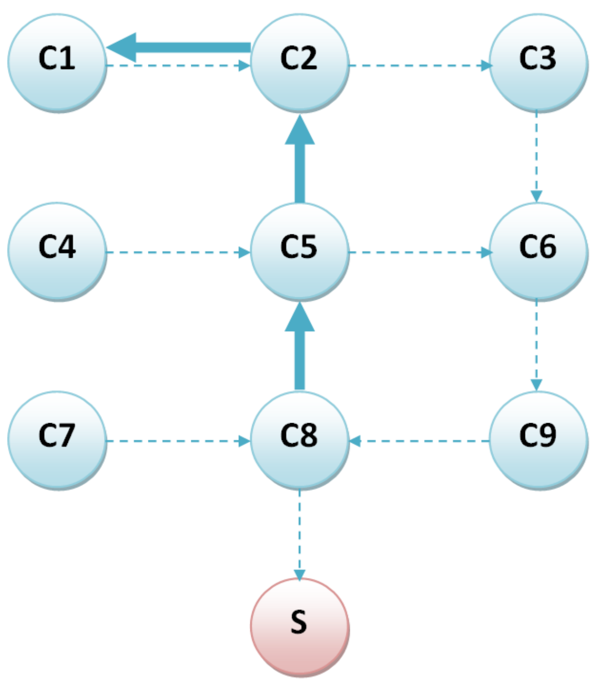

Figure 4.9: Over-hearing scheme at cluster-head level

As seen in Figure 4.9, cluster-head 8 has the most recent data. While it is sending the result to the sink, cluster 5 will also hear that message. On hearing this message, it will update its local last situation value of the network. While cluster-head 5 is sending its regular data to cluster-head 6, cluster-head 2 will hear this message and eventually cluster-head 1 will also get the latest data. With this method, all cluster-heads will be informed about the current fire threat level of the forest.

A cluster-head also computes the local fire threat level by aggregating the messages from its own child sensor nodes. Comparing the overall fire threat level and the local situation, the cluster-head decides about the priority degree of its messages. Another factor in deciding the priority level is the threat level difference between two consecutive threat level estimations. As mentioned in the initialization phase, cluster-heads forward the messages coming from their neighbors at pre-calculated times. However, at critical situations such as the followings, the messages should be propagated to the sink rapidly:

- Previously there has not been any fire threat message but a cluster-head just heard a critical level message,
- The temperature level at a zone rises or falls quickly,
- The current fire threat level at a zone gets higher than the threat levels of than the other zones in the forest.

Table 4.8 summarizes the goals and the design choices of our proposed framework.

Table 4.8: Goals and Design Choices

	Energy	**Early Detection**	**Adaptiveness**	**Forecast**
Sensor Deployment	Homogenous (same level of energy consumption)	Homogenous (guarantee the max distance btw. node and fire)	Ability to locate nodes to distant locations (higher initial energy)	Homogenous (gathering data from all locations while fire is progressing)
Network Architecture	Clustered (manage sensor nodes for low and fair energy consumption)	Clustered (forward critical messages quickly)	—	Clustered (managing the sensor nodes during the fire event)
Intra-Cluster	Sleeping mechanism and periodic informing in the cluster	In case of fire threat, breaking the periodic informing rules	Selecting new cluster-head when old CH dies	gathering more data from the nodes in fire or getting close to the fire
Inter-Cluster	Message sending time table between the cluster-heads	Forwarding the alarm message quickly to the sink	Handling the death of a cluster-head	—

Chapter 5

Experimental Results and Evaluation

In order to develop and test our network architecture and WSN protocols, we have implemented our own simulation platform. In this chapter, we present both the details of our simulator and the evaluation results of our proposed protocols.

5.1 Simulation Platform

In this section, we mention about the features of our platform which consists of two main parts: forest fire simulation and WSN simulation.

5.1.1 Forest Fire Simulation

To test our proposed framework, we need a forest fire simulator. For that purpose, we used FireLib [10]. FireLib is an open-source library that predicts forest fire behavior. FireLib is a C function library prepared for the estimation of the spread behavior of forest fires and it presents an application programming interface (API) for fire growth modeling. It is derived from the BEHAVE fire behavior algorithms and it is optimized for cell-based fire growth simulations. The library takes several inputs and predicts the spread rate, intensity, flame length and scorch height of the forest fire. The calculation of fire spread is introduced at four stages;

Fuel: This stage takes the fuel bed and fuel particle characteristics as inputs. These inputs affect the fire ignition temperature and the spread rate of the fire.

Moisture: The moisture level affects the reaction intensity and temperature increment value during the fire.

Wind and weather: The speed and direction of the wind affect the development of fire. Also, the slope, aspect of the zone, the air pressure and elevation of the field are taken as inputs.

Direction: This stage produces the elliptical growth model of the forest fire. Spread rate, fireline intensity, flame length and scorch height are produced as outputs of this stage.

After the inputs of the simulator are given, the output is produced. As the output, ignition time table of each cell is given (the region is divided into cells). For each cell in the simulation area, the simulator calculates the fire ignition time. The results are produced into a text file which is also called as output map, as seen in Figures 5.1 and 5.2. However, no temperature information is processed in FireLib, only time and location based information is presented. For the calculation of fire, we set two parameters; temperature level at the start time of the fire and increment ratio of temperature. Via these values and the start time of fire at each cell, we can get the temperature values of each cell.

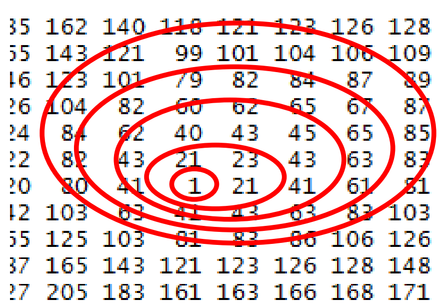

Figure 5.1: Close view of a sample output map and the development of forest fire

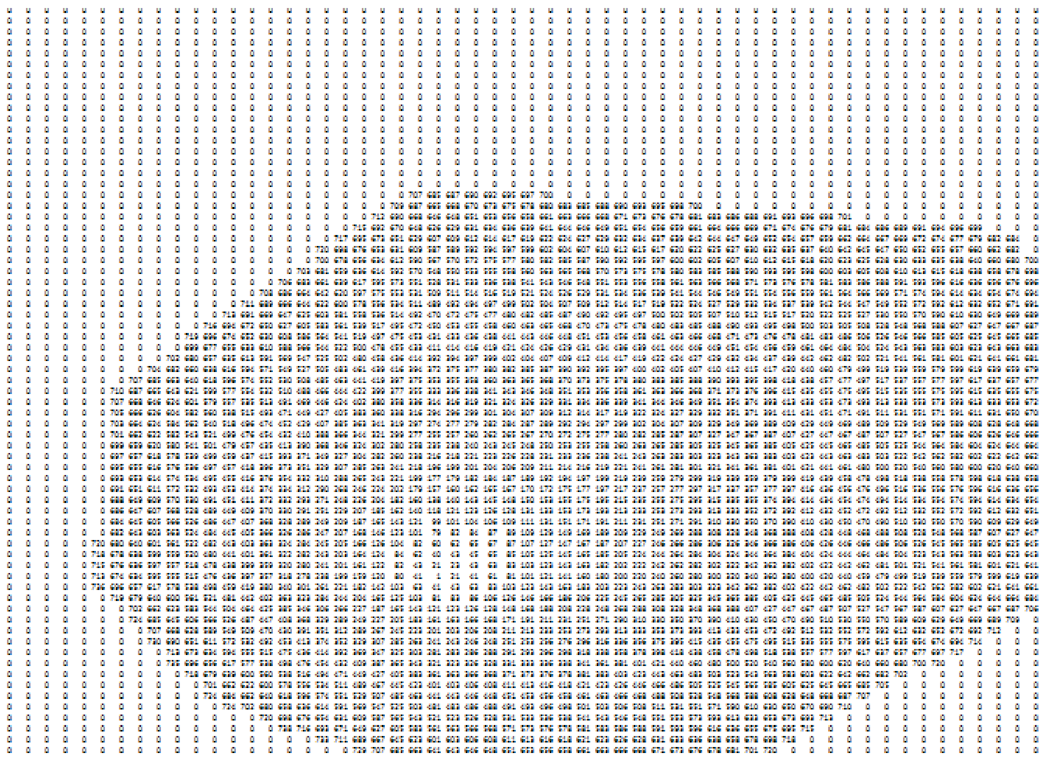

Figure 5.2: Overall view of the output map

5.1.2 WSN Simulation

The core of our simulation study is the simulation of a wireless sensor network. We implemented a simulator of a wireless sensor network that is deployed to a forest. Our simulator performs the following actions:

- taking the output map of the forest fire simulator,
- taking several simulation related inputs,
- running the wireless sensor network protocol on each sensor, and
- producing results that will be used to evaluate the proposed sensor network protocols.

We develop our simulator in Microsoft Visual Studio 2008. In Figure 5.3, a sample screen-shot from the simulator is shown. Also in Figure 5.4, the components of the simulator is shown.

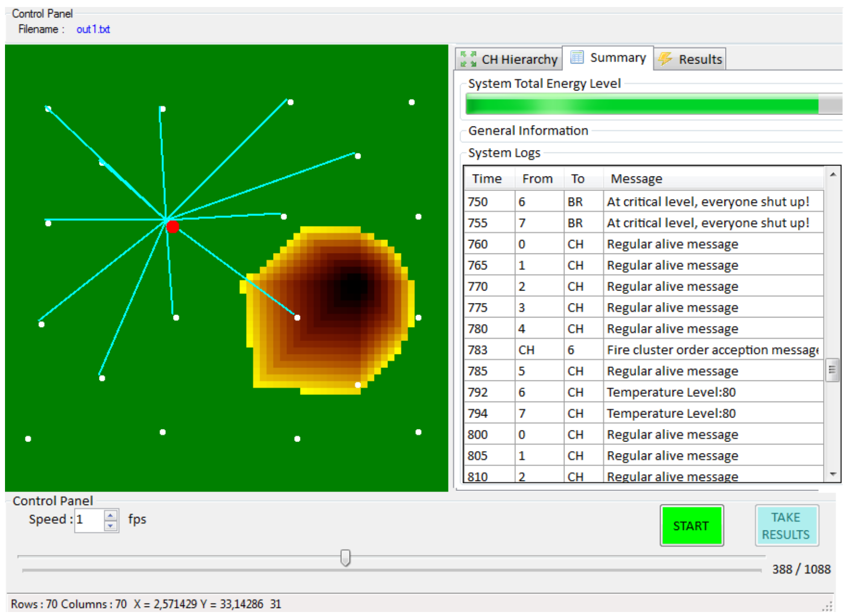

Figure 5.3: A sample screen-shot from the simulator

The simulator consists of the following components:

1. *Engine*: This is the core component and it controls actions of the sensor nodes and the message center. Its basic role is distributing necessary values between the components. For example, sharing current time value with the message center and the sensor nodes.

2. *Message center*: When the sensor nodes generate data, they forward their messages (that they want to transmit to a specific one or more destinations) to the message center. The message center makes the respective sensor nodes aware of their incoming messages. For a message to be successfully transmitted, the message center takes the following factors into account:

   - the sender and receiver of the message,
   - the physical distance between the sender and the receiver nodes,
   - the power level of the sender and receiver nodes, and

- the messages that are being transmitted in the medium (if collision control is set).

3. *Sensor node*: The sensor node component performs the most actions in the simulator. The following information is collected by a sensor node before taking an action:

    - time (from the engine component),
    - temperature (from FireLib),
    - current state and state related parameters (from the state component), and
    - incoming messages (from message center).

    Throughout the life-cycle of a sensor node, the following basic actions are performed on the simulator:

    - sending a message to another node,
    - sending a broadcast message,
    - gathering messages that are sent to itself,
    - overhearing the messages in the medium,
    - turning on and off the radio, and
    - turning on and off the temperature sensor.

    A sensor node decides its actions by following a protocol that we implement as a separate entity in the simulator. Implementation of the protocol as a separate component enables us to modify it easily and test different alternative protocols.

4. *FireLib component:* This component processes the output of the FireLib simulator. The ignition table, that is the output of the FireLib simulator, only contains time and location related information. Calculating the temperature level of each cell at different times is the job of this component. It takes the following inputs and decides about the temperature value at a given time:

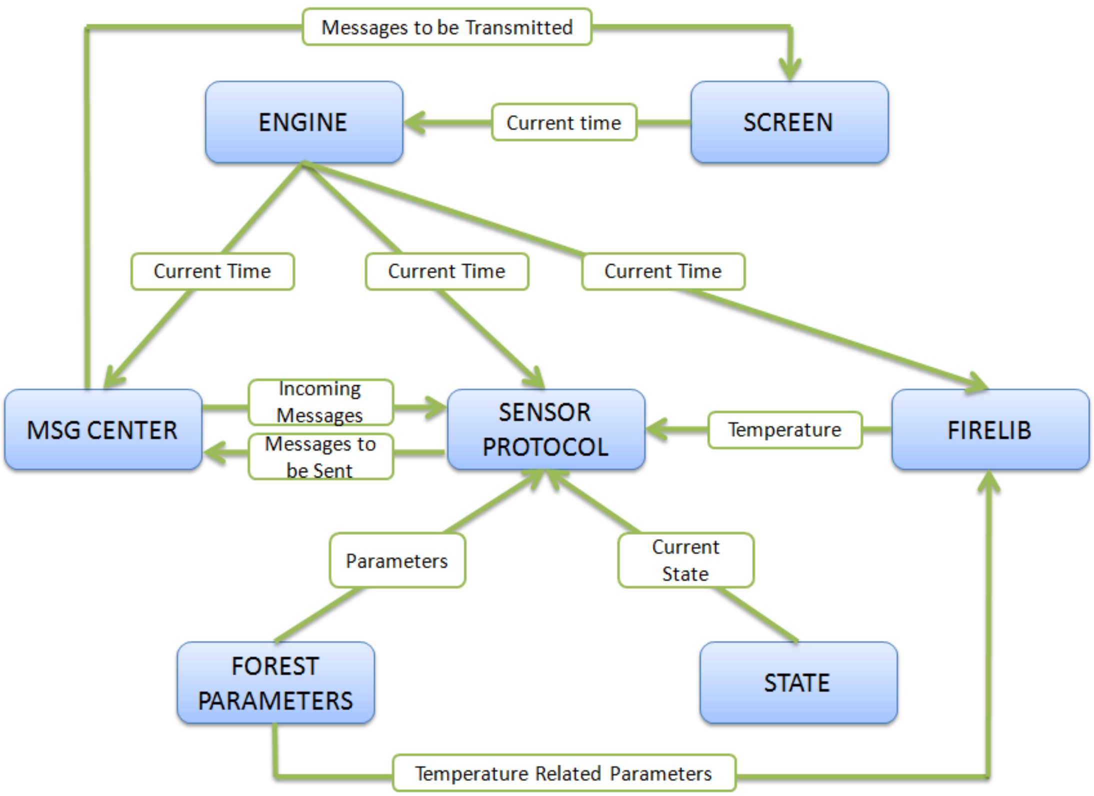

Figure 5.4: Components of the simulator

- ignition temperature,
- critical level temperature,
- risk-free time temperature,
- temperature increment ratio, and
- temperature increment duration.

In this simulator, when a sensor node asks for its current temperature value, the FireLib component does the necessary calculations and replies to the sensor node component with the current temperature.

5. *State:* This component keeps state information about the sensor nodes. States are decided and changed according to the protocol. In other words, for each different phase of a sensor node and protocol executed, a state is decided. For example, the cluster initialization phase contains several states. A state includes many variables, and depending on the current values of those variables, an appropriate action is taken.

6. *Parameters of the simulator:* There are several inputs that have to be specified for the simulator engine:

   - Deployment related inputs: These inputs are used to determine how to deploy sensor nodes to the field. Deployment can be either grid layout, or random layout. The expected distance between sensor nodes is given as input as well,
   - Sensor properties: The maximum distance that a sensor node can send a message to, the initial energy level of the sensor nodes, and the maximum temperature level up to which a sensor node can stay alive are some of the sensor specific inputs,
   - Simulation parameters: Energy consumption values and time/tick values are specified in this group.

Several sample screen-shots from the simulation are shown in Figures 5.5,. . . ,5.10.

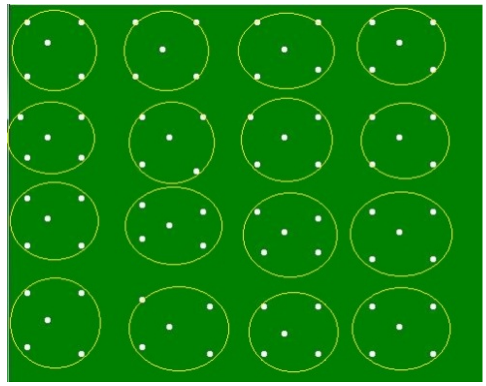

Figure 5.5: Initialization phase and clusters

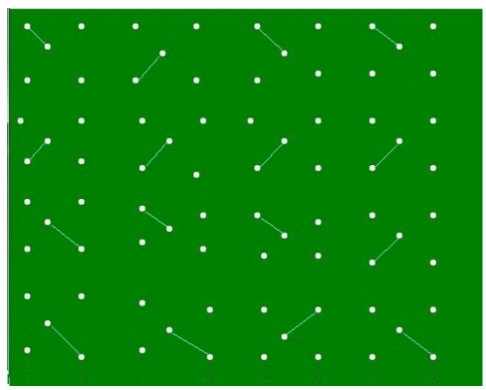

Figure 5.6: Regular inform messages

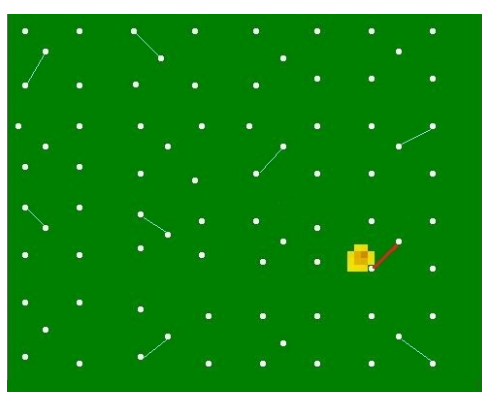

Figure 5.7: Detection of forest fire

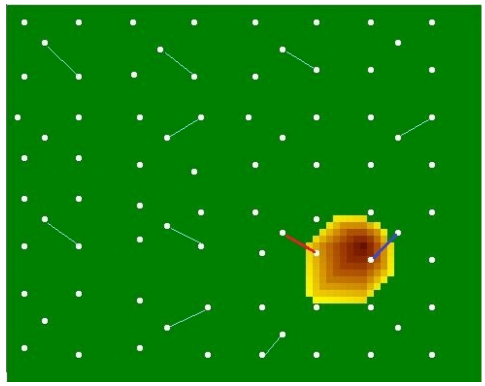

Figure 5.8: Progress of fire - I

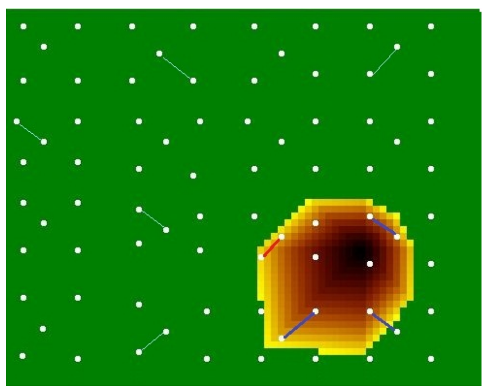

Figure 5.9: Progress of fire - II

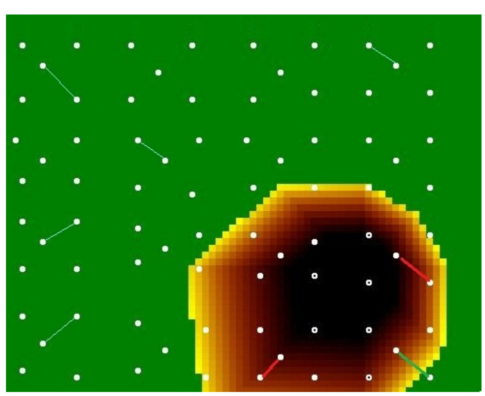

Figure 5.10: Progress of fire and dead sensors

5.2 Results and Evaluation

We performed extensive simulations to evaluate various components of our system and different design choices that we considered while deciding on a system component. Below, we analyze each system component in a separate subsection.

5.2.1 Sensor Deployment Scheme

The deployment scheme used for positioning the sensor nodes to a forest affects the system performance from various aspects.

Relation between the sensor deployment scheme and energy consumption: We have regular or random deployment choices which closely affect the energy consumption in the network as well as the reaction time to detect the occurrence of a fire. Our simulation results show that, considering the low energy consumption goal, regular (deterministic) deployment scheme is preferable. As seen in Figure 5.11, if the sensor nodes are deployed in a deterministic manner, in average they consume less energy compared to the case where they are deployed randomly. The nodes that are closer to the cluster-head consume less energy than the other nodes in the cluster. However as mentioned in Chapter 4.1, increment of the message sending distance exponentially increases the energy consumption and therefore the energy consumed in total is higher in random deployment scheme. Also, in the same figure it is seen that as the average distance between the sensor nodes increases, the difference between the energy consumption in two schemes gets larger. In this experiment, the results are gathered by simulating regular and random deployment schemes with twenty sensor nodes. For both models, in total, fourteen experiments have been conducted in which the distance between the sensor nodes varies between 5 meters and 30 meters.

Another experiment result considering the relationship between the sensor deployment scheme and energy consumption is shown in Figure 5.12. In this figure, the balanced energy consumption is analyzed in both model. In the y-axis, the difference between the energy consumption between the two nodes in the same

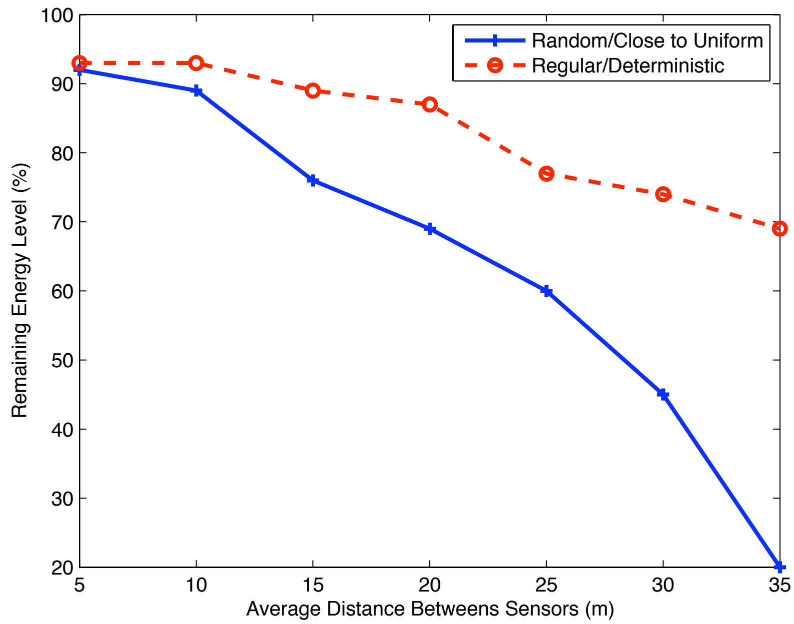

Figure 5.11: Remaining energy levels of sensor nodes of regularly deployed sensor nodes and sensor nodes that are deployed to distant locations

cluster is listed. In the deterministic case, each sensor node consumes energy in a very similar level therefore the difference doesn't increase in time and is kept at a low value. However, in the random deployment case, the node that is closer to the cluster-head consumes less energy and the difference between the nodes increases in time. This result shows that for the balanced-energy consumption goal, regular deployment is preferable.

Relation between the sensor deployment scheme and early detection: When the early detection goal is considered, again regular deployment scheme is more successful as can be seen in Figure 5.13. As the figure shows, the average distance between a fire ignition location and a closest sensor node is larger in the random deployment case. This means it will take more time until a sensor node detects the increase in temperature due to a fire ignition. Additionally, as the average distance among the sensor nodes increases, the difference between the performance of these two approaches (regular and random) gets larger. The results are gathered by simulating 12 fire ignition occurrence for each different case where the distance between the sensor nodes varies between 5 and 30 meters.

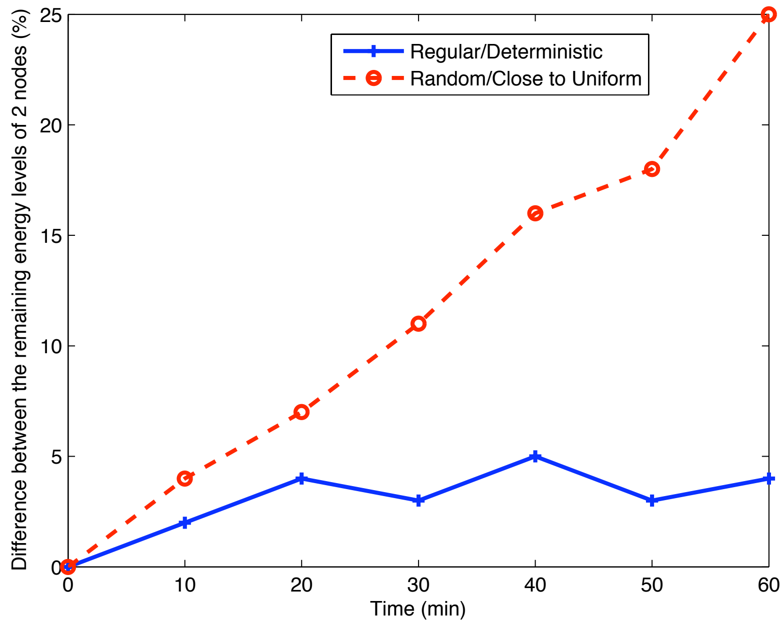

Figure 5.12: Difference between the energy levels of two nodes in the same cluster

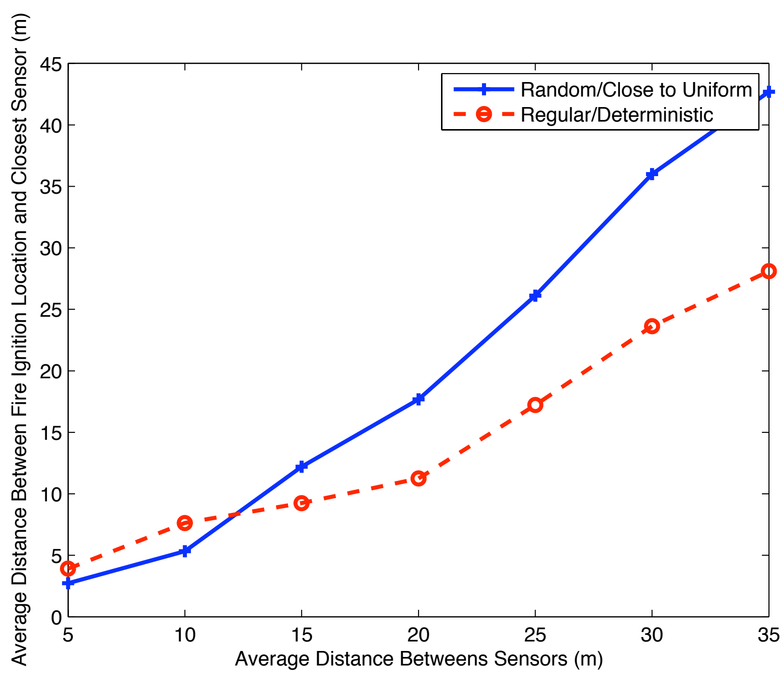

Figure 5.13: Distance between fire ignition and closest sensor in regular and random deployment schemes

The distance between the sensor nodes affects the time required for the heat waves to reach to the sensor nodes. In Figure 5.14, it is observed that it takes more than 10 minutes for the sensor nodes to sense the fire thread when the distance between fire ignition location and the sensor node gets more than 20 meters. This experiment shows that the distance between the sensor nodes should be kept under 20 meters for a successful early detection system.

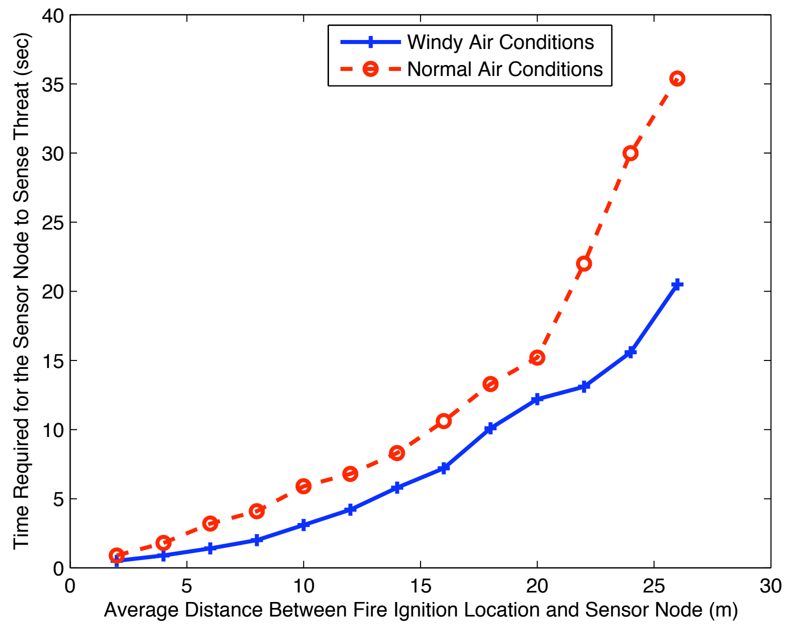

Figure 5.14: Time required for the sensor nodes to sense the fire threat at different distance levels

5.2.2 Architectural Design

A clustered network architecture is proposed in our framework for the purpose of more effective network processing and data fusion. Cluster-heads can easily be designated as the responsible points in the network to perform data processing, data fusion, and to provide coordination and cooperation. We evaluated how the clustered architecture and processing at cluster-heads affect the transmission of useful traffic in the network. We measured the ratio of important (critical) messages carried in the network to all the messages. As seen in Figure 5.15, when clustering is used and cluster-heads do local computations to fuse data and to reduce the volume of unnecessary traffic, the ratio of critical messages to all the messages sent to the sink becomes around 90 percent. When clustering is not used, this ratio becomes much lower, meaning that there exist a lot of messages (that are not critical) unnecessarily transmitted in the network from the sensor nodes to the sink node. In Figure 5.15, there are 4 curves; curve 1 (at the top) and 4 (at the bottom) belong to non-clustered network and the difference between two curves is very high. On the other hand, curves 2 and 3 (in the middle) belong to clustered networks and the difference is very low compared to non-clustered hierarchy. Hence the percent of critical messages inside all messages is quite high in clustered hierarchy. During the fire occurrence, the cluster-heads gather data from the nodes in fire more frequently and after making local computations for determining the number of dead and alive sensor nodes, the temperature increment ratio, etc., cluster-heads forward these results to the sink. Therefore, clustered-hierarchy is preferable considering the goals of early detection, and low energy consumption, as well as forecast adaptiveness.

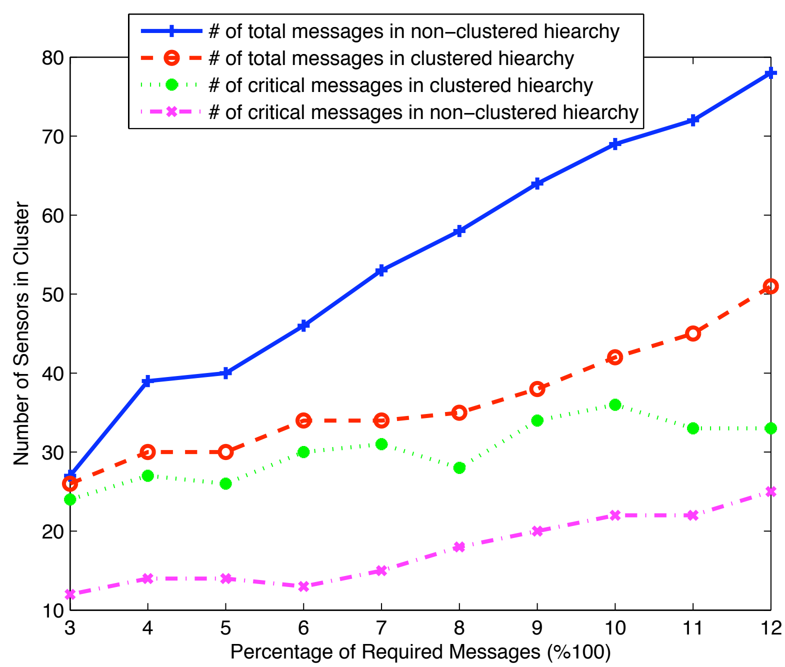

Figure 5.15: Number of messages sent to the sink when local computation at the cluster level is applied compared to when no local computation is performed

5.2.3 Environment Aware Communication Protocols

The protocol that manages the communication between the child sensor nodes and the cluster-head, and the protocol regarding the communication between the cluster-heads are regulated according to the environmental conditions. These environment aware actions are effective for both early detection goal and low energy consumption goal.

Effect of the protocol to the energy consumption goal: As seen in Figure 5.16, the energy consumed in the model that does not consider environmental conditions is kept in similar levels throughout the year. However, in our proposed scheme which considers the environmental conditions, the activity level of the sensor nodes can be kept at a lower level (i.e., by sending regular inform messages less frequently to the cluster-head) in the months when the risk of fire is quite low (like months of winter). And the activity level can be kept high in months when the risk of fire is higher (like summer time). When the activity level is increased in summer times, the sensor nodes will consume more energy but that can be compensated with the very low activity level, hence very low energy consumption, in winter times.

To better judge the proposed scheme, we take the fire risk level of different months throughout the year into consideration. For this purpose, we use a comparison index, c, by multiplying the average fire risk level and the total energy consumed at each months.

$$c = \triangle FT_i \times E_i \tag{5.1}$$

where

- $\triangle FT_i$ is the average fire risk level of that month and
- E_i is the total energy consumed by sensor nodes in that month

Figure 5.17 shows the value of such an index for each month. During the

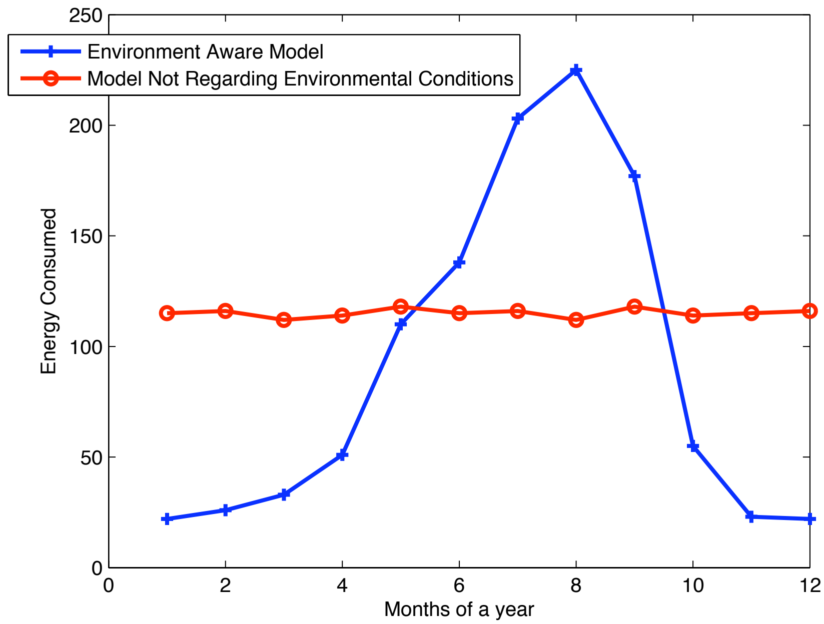

Figure 5.16: Energy consumption levels of environment aware and base models throughout the year

months that are not in fire season, the comparison index value is extremely high for the model that does not regard environmental conditions. Although the risk of fire is very low at those months, unnecessary amount of energy is consumed. As a result, it is seen that the inter-cluster communication protocol oversees both early detection and low energy consumption goals.

Additionally, in Figure 5.18, we show the cumulative energy consumption throughout the year for both our environment-aware scheme and the usual environment-unaware scheme. As the figure shows, our proposed scheme consumes less overall energy when the whole year is considered.

Effect of the protocol to the early detection goal: Managing the activity level of the sensor nodes depending on the environment conditions is also effective for the early detection goal of the system. Especially, during the fire season, the sensor nodes act pro-actively and the fire threat messages are propagated to the sink as rapidly as possible. As can be seen in Figure 5.19, fire detection time is kept at reasonable values in critical seasons with the environment-aware scheme.

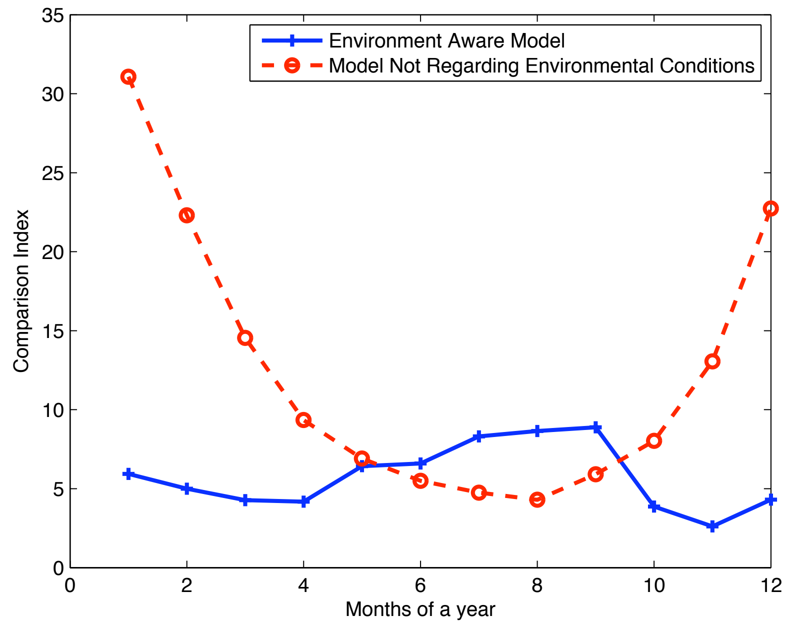

Figure 5.17: Energy consumption level × the fire risk level values of environment aware and base models throughout the year

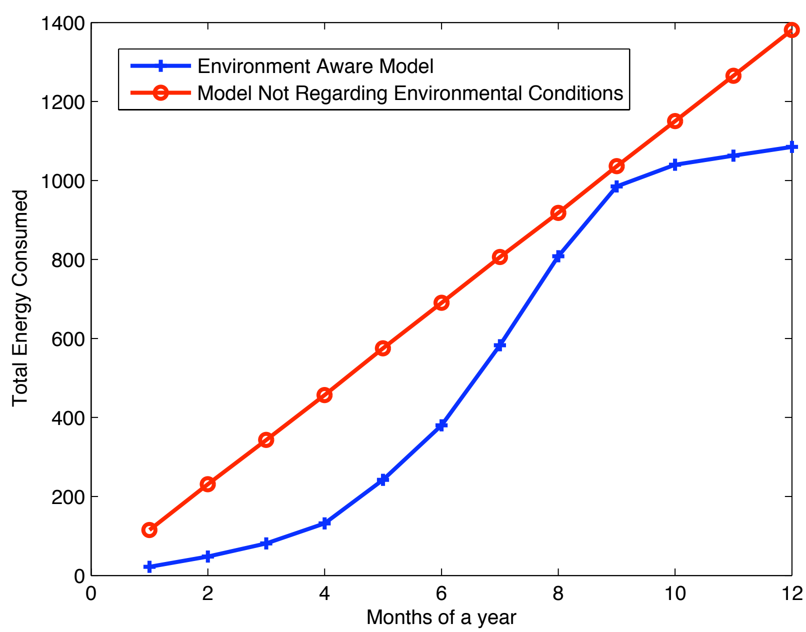

Figure 5.18: Cumulative value of energy consumption level × the fire risk level values of environment aware and base models throughout the year

The model that does not consider environmental conditions, however detects the fires nearly with the same latency throughout the year (regardless of the month of the year). This is acceptable for winters, but not for summer seasons.

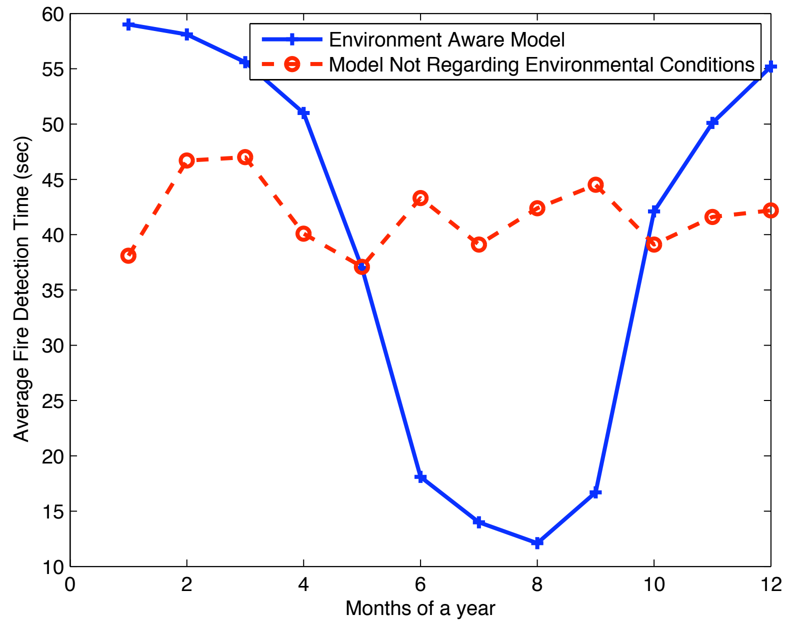

Figure 5.19: Fire detection durations of environment aware and base models throughout the year

We also propose the use of a comparison index to better evaluate the fire detection durations of the models.

$$c = \triangle FT_i \times T_i \tag{5.2}$$

where

- $\triangle FT_i$ = Average fire threat level of that month and
- T_i = Average fire detection time of the regular nodes at that month

As Figure 5.20 shows, environment-aware scheme's performance is much better in terms of this metric. During the months that are not in the fire season, the fire threat level is low and the average fire detection time is similar in both models. However, in fire-season the average fire detection time of the environment aware model is too short compared to the other model and since the extent level of fire threat is taken into account, the success of the model increases during the months in the fire season.

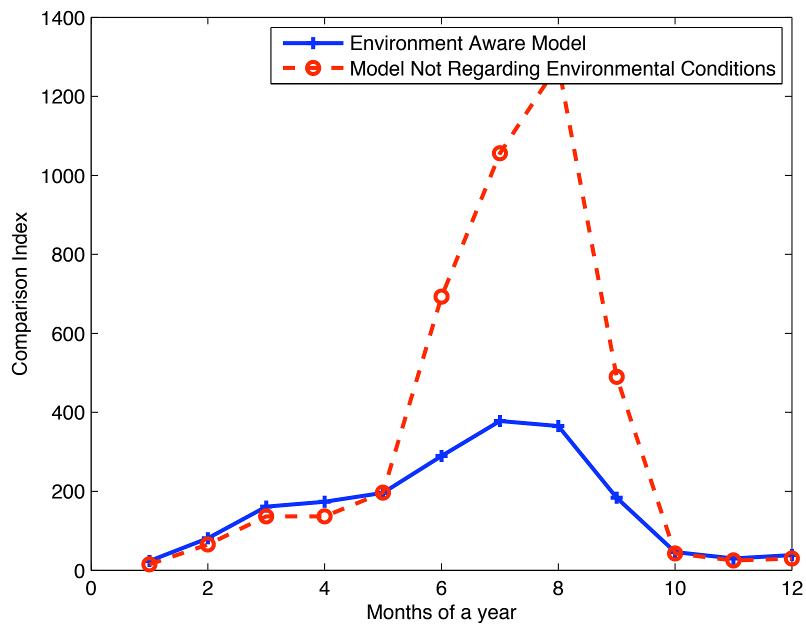

Figure 5.20: Fire detection duration × the fire risk level values of environment aware and base models throughout the year

Also, when the number of nodes increase, the success of the environment aware model rises and this can be seen in Figure 5.21. The reason behind this result is related to the inter-cluster communication protocol. In an urgent situation, cluster-heads rapidly propagate the fire alarm message to the sink.

When the number of sensor nodes in a cluster is analyzed, it is seen that the environment aware method gives better performance. In the environment aware model, when a sensor senses a fire threat, it immediately broadcasts an emergency packet so that the cluster-head becomes aware of the threat quickly. As the size of the cluster increases, the difference between the two model increases. The reason behind this result is that, in the model that does not regard environmental conditions, as the size of the cluster increases, the time for the sensor (which

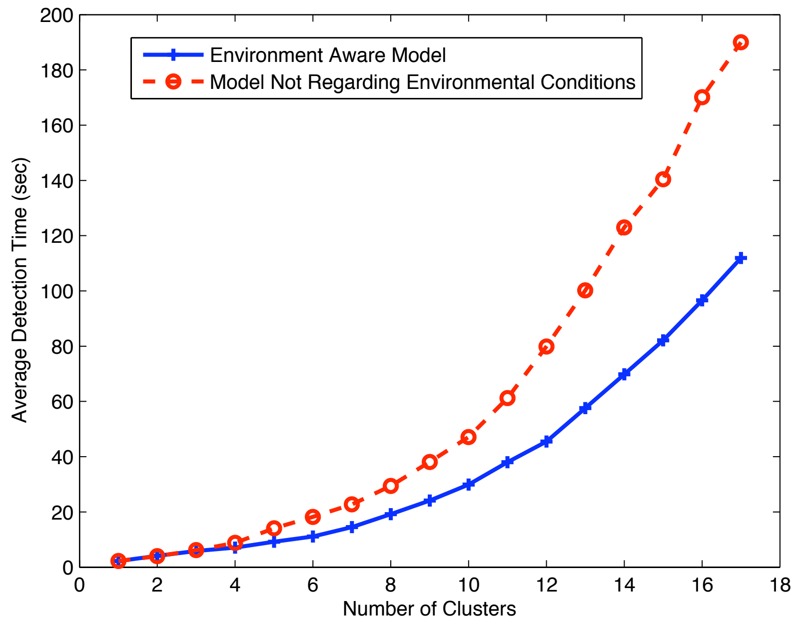

Figure 5.21: Fire detection durations of environment aware and base models as the number of clusters in the network varies

senses the threat) to send the emergency message takes longer time. Relative performance results can be seen in Figure 5.22.

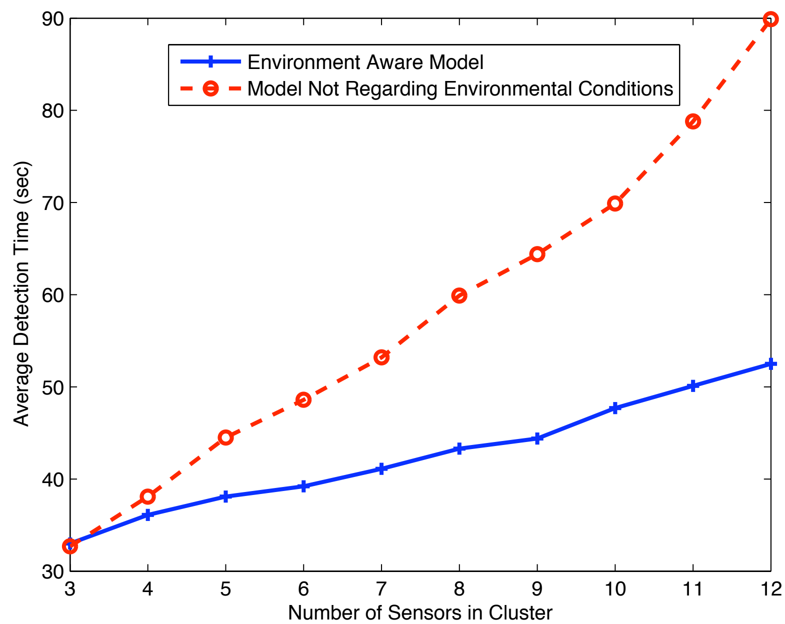

Figure 5.22: Fire detection durations of environment aware and base models as the number of sensors in cluster varies

Chapter 6

Conclusion

The aim of this thesis is to propose a comprehensive study that covers all parts of the life cycle of a wireless sensor network system that is specialized for forest fire detection. While continuously preserving the goal of early detection of forest fire, we also try to construct a system that regards the low energy capacity of the sensor nodes and difficult environmental conditions that may hinder the network performance. Moreover, our system presents meaningful data to the sink during the occurrence of fire event which will help fastening the forecast operation of forest fire. Starting from the sensor node deployment scheme, several design decisions are evaluated considering the system goals. Regular and homogenous deployment scheme is preferred from the energy consumption and early detection points of view. Sensor nodes run in a clustered hierarchy and a data aggregation methodology is applied for creating less message over head throughout the network. Cluster-heads manage the regular sensor nodes according to varying environmental conditions; the activity level of the sensor nodes becomes higher during critical time durations and the nodes go to sleep when the risk of fire is low. The communication schemes between a cluster-head and the regular nodes, and among the cluster-heads aim forwarding critical data to the sink as soon as possible. Also, since the nodes that are closer to the sink have to forward messages more frequently and consume more energy, communication protocols preserve the goal of balanced-energy consumption and it provides that each node sends about the same amount of messages at each period.

The system is also available for several enhancements. Local data management and data synchronization in cluster-heads, localization of the nodes via GPS or other techniques, estimation of fire ignition location with or without GPS, dynamic route determination at the cluster-head level, dynamic cluster-head selection and forest fire spread estimation at the sink are some topics which can be focused in the future studies.

Bibliography

[1] *Republic of Turkey, Ministry of Environment and Forestry, General Directorate of Forestry*, http://www.ogm.gov.tr/, Last accessed August 2010.

[2] *Washington State University Department of Natural Resource Sciences*, http://www.ruraltech.org/, Last accessed August 2010.

[3] D. Morvan, M. Larini, J. L. Dupuy, P. Fernandes, A. I. Iranda, J. Andre, O. Sero-Guillaume, D. Calogine, P. Cuinas, *Behavior Modelling of Wildland Fires: A State of the Art,* EUFIRELAB: Euro-Mediterranean Wildland Fire Laboratory, 2002.

[4] *National Fire Danger Rating System (NFDRS),* http://www.wrh.noaa.gov/sew/fire/olm/nfdrs.htm/, Last accessed August 2010.

[5] *Aerovision Web Page,* http://www.aerovision-uav.com/, Last accessed August 2010.

[6] *The Advanced Very High Resolution Radiometer (AVHRR) Web Page,* http://noaasis.noaa.gov/noaasis/ml/avhrr.html/, Last accessed August 2010.

[7] *B.C. Fire Lookout Towers Web Page,* http://www.firelookout.com/bc.html/, Last accessed August 2010.

[8] *The Moderate Resolution Imaging Spectroradiometer (MODIS) Web Page,* http://modis.gsfc.nasa.gov/, Last accessed August 2010.

[9] *National Center for Landscape Fire Analysis (NCLFA),* http://firecenter.umt.edu/, Last accessed August 2010.

[10] *FireLib Public Domain Software for the Wildland Fire Community,* http://fire.org/, Last accessed August 2010.

[11] D. M. Doolin and N. Sitar, *Wireless Sensor Nodes for Wildfire Monitoring,* In Proc. of SPIE Symposium on Smart Structures and Materials, San Diego, pp. 477484, 2006.

[12] B. Son, *A Design and Implementation of Forest-Fires Surveillance System Based on Wireless Sensor Networks for South Korea Mountains,* International Journal of Computer Science and Network Security, Vol.6, No. 9B, pp. 124-130, September 2006.

[13] L. Yu, N. Wang, and X. Meng, *Real-time Forest Fire Detection with Wireless Sensor Networks,* In Proc. of International Conference On Wireless Communications, Networking and Mobile Computing (WiMob), pp. 1214-1217, 2005.

[14] C. Hartung, R. Han, *FireWxNet: A Multi-tiered Portable Wireless System for Monitoring Weather Conditions in Wildland Fire Environments,* In Proc. of 4th International Conference on Mobile Systems, Applications and Services, pp. 28-41, 2006.

[15] B. Wenning, D. Pesch, A. Giel, C. Gorg, *Environmental Monitoring Aware Routing: Making Environmental Sensor Networks More Robust,* Springer Science Business Media Telecommunication Systems, Vol. 43, Numbers 1-2, pp. 3-11, 2009.

[16] E. Ngai, Y. Zhou, M. Lyu, J. Liu, *A Delay-aware Reliable Event Reporting Framework for Wireless Sensor-actuator Networks,* Ad Hoc Networks, Vol. 8, Issue 7, pp. 694-707, 2010.

[17] M. Shyam, A. Kumar, *Obstacle Constrained Total Area Coverage in Wireless Sensor Networks,* CoRR abs/1001.4753, 2010.

[18] J. Lloret, M. Garcia, D. Bri, S. Sendra, *A Wireless Sensor Network Deployment for Rural and Forest Fire Detection and Verification,* Sensor Nodes, Vol. 9, Issue 11, pp. 8722-8747, 2009.

[19] M. Hefeeda, M. Bagheri, *Forest Fire Modeling and Early Detection Using Wireless Sensor Networks,* Ad Hoc Sensor Wireless Networks, Vol. 7, pp. 169-224, April 2009.

[20] E. M. Garcia, M. A. Serna, *Simulating a WSN-based Wildfire Fighting Support System,* In Proc. of IEEE International Workshop on Modeling, Analysis and Simulation of Sensor Networks (MASSN), pp. 896-902, 2008.

[21] K. Akkaya, M. Younis, *A Survey of Routing Protocols in Wireless Sensor Networks,* Ad Hoc Networks, Vol. 3, No. 3, pp. 325-349, 2005.