A framework for use of wireless sensor networks in forest fire detection and monitoring*

Yunus Emre Aslan, Ibrahim Korpeoglu, Özgür Ulusoy
Bilkent University,
Department of Computer Engineering,
06800 Ankara, Turkey
Email: {aslany,korpe,oulusoy}@cs.bilkent.edu.tr

Abstract

Forest fires are one of the main causes of environmental degradation nowadays. Current surveillance systems for forest fires lack in supporting real-time monitoring of every point of a region at all times and early detection of fire threats. Solutions using wireless sensor networks, on the other hand, can gather sensory data values, such as temperature and humidity, from all points of a field continuously, day and night, and, provide fresh and accurate data to the fire-fighting center quickly. However, sensor networks face serious obstacles like limited energy resources and high vulnerability to harsh environmental conditions, that have to be considered carefully. In this paper, 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, sensor deployment scheme, and clustering and communication protocols. The aim of the framework is to detect a fire threat as early as possible and yet consider the energy consumption of the sensor nodes and the environmental conditions that may affect the required activity level of the network. We implemented a simulator to validate and evaluate our proposed framework. Through extensive simulation experiments, we show that our framework can provide fast reaction to forest fires while also consuming energy efficiently.

*This work is supported in part by European Union FP7 Framework Program FIRE-SENSE Project 244088.

1 Introduction

Forest fires are 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 very important in fighting against fires. Spread features of forest fires show that, in order to put out a 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]. Besides early detection capability, estimating the spread direction and speed of fire is also important in extinguishing fires.

Unreliability of human observation towers, in addition to the difficult life conditions of fire lookout personnel, has led the development and use of various technologies aiming to make the fire fighters aware of the forest fires 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 a control center [5, 7, 8]. However, the accuracy of such a system is highly affected by terrain, time of day, and weather conditions such as clouds, light reflections, and smoke from innocent industrial or social activities. Another alternative technology for detecting forest fires is the use of satellites and satellite images. Usually, satellites provide a complete image of the earth every 1 to 2 days. This long scan period, however, is not acceptable for detecting forest fires quickly. Additionally, the smallest fire size that can be detected by such a system is around 0.1 hectare, which also prevents fire detection just at the time when the fire starts, and fire localization error is about 1 km, which is not very accurate.

As a promising alternative, wireless sensor networks (WSNs) are an emerging technology that can be used for forest fire detection and related activities [24, 26, 12, 13, 14]. A wireless sensor network consists of small, battery-pwered, and low-cost sensor nodes that have the capability of sensing, processing, and wireless communication [18]. Wireless sensor nodes that are deployed to a forest can collect data such as temperature, humidity, barometric pressure, and deliver this highly important data in raw or processed form to a center via a base station (or sink node), where incoming data can be analyzed automatically. As a result, fires and some other related events can be detected at the center without requiring manual, human-centric operations. There are, however, a lot of issues to consider and resolve in using wireless sensor networks for forest monitoring and forest fire detection. For example, the limited energy resources of sensor nodes and the though environmental conditions can limit the success of forest fire detection systems that are based on wireless sensor networks. Constant surveillance of the whole forest is required and this may cause excessive energy usage if not carefully planned. Therefore, a wireless sensor network for forest fire detection should consider several parameters and trade-offs together.

In this paper, we present a framework for the design of a wireless sensor network for forest monitoring and fire detection considering several goals simultaneously. As a part of our framework, we propose a network architecture and related protocols that will enable both rapid detection of forest fires and cautious use of energy resources. In our design, when there is no fire, the sensor network is not very aggressive in sensing and communicating various sensory data. But when there is a fire threat, the network operates in an emergency mode, and senses and communicates as fast as possible. Similarly, since the risk of fire depends on terrain, season and current weather conditions, our proposed design adapts its operation mode to the current level of risk of fire. In our proposed system design, except for the periods of forest fire, the sensor nodes mostly work in a low-duty cycle mode (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 to run in each sensor node considers fire threat cautiously and in case of an abnormal temperature change, informs the control center about the possibility or occurrence of fire rapidly.

Our framework includes four major components: an approach for deployment of sensor nodes, an architecture for the sensor network for fire detection, an intra-cluster communication protocol, and an inter-cluster communication protocol. We simulated our proposed design to show the validity of the protocols and to evaluate the proposal. We report the simulation results and show how the proposed framework can adapt to changing risk levels and in this way use the energy resources efficiently without harming the effectiveness of the system in detecting fires quickly.

The remainder of this paper is organized as follows. Section 2 discusses related studies on forest fire detection with wireless sensor networks. Section 3 describes the proposed framework that includes four major components mentioned above. Section 4 presents our simulation environment and experimental results. Finally, Section 5 concludes the paper and provides a discussion on future work.

2 Related Work

During the last decade, quite extensive research work has been performed on wireless sensor networks, their protocols and algorithms, and their applications [24, 25, 26, 28, 27]. Although these studies are not targeting specifically fire detection application, the approaches proposed are adaptable to various applications, including fire detection and monitoring. In our work, we adapt some of these existing methods (like clustering) and integrate various approaches in the literature to come up with a WSN design specifically targeting energy-efficient and effective forest fire detection.

For fire detection application of wireless sensor networks, specifically, there has been a considerable amount of work carried out as well. In one study, Doolin and Sitar [12] provide experiments through controlled fires in San Francisco area. Their system is composed of ten sensor nodes with GPS capability. The sensor nodes are deployed with ranges up to one kilometer and they sense and forward temperature, humidity and barometric pressure values to a base station. The system was implemented and real-world observations were gathered from the field. However, because of the long distances between sensor nodes, the data arriving to the sink is not valuable enough to detect a fire quickly 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 could fail in delivering the data from all sensor nodes to the base station.

Lloret et al. [19] use a wireless local area network (WLAN) together with sensor-node technology for fire detection. The system they propose mixes multi-sensor nodes with IP-based cameras in a wireless mesh network setting in order to detect and verify a fire. When a fire is detected by a wireless multi-sensor node, the alarm generated by the node is propagated through the wireless network to a central server on which a software application runs for selecting the closest wireless camera(s). Then, real time images from the zone are streamed to the sink. Combining sensory data with images is the most important contribution of this study.

Hartung et al. [15] developed a multi-tiered portable wireless system for monitoring environmental conditions, especially for forest fires. Integrating web-enabled surveillance cameras with wireless sensor nodes, the system can provide real-time weather data from a forest. Three different sensor networks are deployed to different parts of a forest and the communication between the networks is enabled by powerful wireless devices that can send data up to ten kilometers range. The objective of the study is to determine the behavior of forest fires rather than their detection. With a wireless sensor network around an active fire, they measure the weather conditions around the 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 a base station which has the capability of providing long distance communication using satellites. Periodically, the sensor nodes measure the temperature, relative humidity, wind speed and direction, and web-cams provide continuous visual data to the base station.

In all the studies discussed above ([12, 19, 15]), the sensor nodes are deployed to have quite large distances between each other and the sensory data gathered at a center is supported with visual data obtained with cameras. Our proposed system, however, considers a denser deployment strategy where the distances between neighboring sensor nodes are quite short. In this way, we are aiming to detect forest fires in a much faster way and send the related information to a center as quickly as possible.

Son et al. [13] propose a forest fire surveillance system in South Korea in which a dynamic minimum cost path forwarding protocol is applied. After gathering data, a sink node makes several calculations regarding the relative humidity, precipitation and solar radiation, and produces a forest fire risk level. Different from this study, we propose to do in-network processing in cluster-head nodes rather than doing calculations only at a sink node. In this way, in our system, a sink node gathers filtered and processed data, not just raw data. Additionally, [13] 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, applies a low and fair energy consumption strategy by use of appropriate intra- and inter-cluster communication protocols which take the remaining energy levels of sensor nodes into account.

Yu and Wang [14] present a method which applies neural network techniques for in-network data processing in environmental sensing applications of wireless sensor networks. Several data fusion algorithms are presented in their study. Maximum, minimum and average values of temperature and humidity data are calculated by the cluster-heads. Data are propagated to the sink only if a certain threshold is exceeded. The main focus of this study is data aggregation methods, hence energy consumption and forecast capability issues are not discussed.

Ngai et al. [17] provide a general reliability-centric framework for event reporting in wireless sensor networks which can also be used in forest fire detection systems. They consider the accuracy, importance and freshness of the reported data in environmental event detection systems. They present a data aggregation algorithm for filtering important data and a delay-aware data transmission protocol for rapidly carrying the data to the sink node.

Wenning et al. [16] propose a proactive routing method for wireless sensor networks to be used in disaster detection. The protocol is developed to be aware of a node's destruction threat and it can adapt the routes in case of a sensor node's death. The method can also adapt the routing state based on a possible failure threat indicated by a sensed phenomenon.

Hefeeda and Bagharei [20] developed a wireless sensor network for forest fire detection based on Fire Weather Index (FWI) system which is one of the most comprehensive forest fire danger rating systems in USA. The system determines the spread risk of a fire according to several index parameters. It collects weather data via the sensor nodes, and the data collected is analyzed at a center according to FWI. A distributed algorithm is used to minimize the error estimation for spread direction of a forest fire.

Garcia et al. [21] present a simulation environment that can create a model for a fire by analyzing the data reported by sensor nodes and by using some geographical information about the area. The use of topography of the environment distinguishes the study from some other solutions. The estimation of the spread of a fire is sent to hand-held devices of fire fighters to help them in fighting against the fire in field.

The studies described above ([13, 14, 17, 16, 20]) consider and handle a single aspect of environmental monitoring and forest fire detection. In our proposed system, on the other hand, we deal with multiple parameters and trade-offs. We consider and aim both energy-efficiency and early-detection. We also incorporate environmental conditions, obstacles and features in our protocols.

There has been also a significant amount of work performed for clustering in wireless sensor networks [31, 29, 30, 32, 33, 34, 35, 36]. These works, however, consider mostly how clusters are formed and maintained for various applications. They are not focusing on use of clusters for fire detection application. Therefore, their focus and scope are much different than the cluster communication protocols we are proposing in this paper. Moreover, in this paper we are not concentrating on how clusters are formed, but on how clustered hierarchy is utilized in a most efficient and effective manner for detecting and monitoring fires.

3 Proposed Fire Detection System Design

In this section, we describe our WSN-based fire detection system. We first identify the following as some of the important design goals and features that a wireless sensor network should have in order to be able to successfully monitor a forest and detect fires.

1. *Energy efficiency*: Sensor nodes are powered with batteries, therefore a wireless sensor network deployed for fire detection should consume energy very efficiently. Energy consumption should also be balanced fairly among nodes. Usually the deployment area is very large and thousand of sensor nodes may be needed, and therefore replacing batteries may be too costly, impractical or even not possible.

2. *Early Detection and Accurate Localization*: It is important to detect a forest fire as early as possible and to estimate the fire location with high accuracy. A forest fire usually grows exponentially and it is crucial that the fire should be detected and interfered in about six minutes to prevent the fire from spreading to a large area [4]. Accurately estimating the fire position is important to send the fire fighting personnel to the correct spot in the shortest possible amount of time.

3. *Forecast Capability*: Being able to forecast the spread direction and speed is important for planning fire fighting, being proactive in mobilizing resources, and warning the surrounding area. Accurate forecasting requires accurate and fresh sensory data to arrive at the decision and control center from all points of the forest, especially from and around the region where the fire has occurred (i.e., critical zones).

4. *Adapting to Harsh Environments*: A sensor network for forest fire detection will operate usually in harsh environments and therefore should be able to deal with and adapt to harsh conditions. It should be able to recover from node damages, link errors, high temperature, humidity, pressure, etc.

Our aim in this work is to consider the above goals as much as we can in designing a wireless sensor network for fire detection. Besides these goals, there may be some other crucial requirements for a WSN designed for fire detection, such as providing security, coping with vandalism, incorporating self-healing mechanisms, being able to self-organize, etc. We do not consider these requirements in this work and leave them as future work issues.

Our proposed framework involves the design of four main parts: 1) a sensor deployment scheme, 2) a clustered network architecture, 3) an intra-cluster communication protocol, and 4) an inter-cluster communication protocol. Next, we describe the design of each of these parts in more detail.

3.1 Sensor Deployment Scheme

The sensor node deployment scheme can affect the design and performance of all aspects of the system. In a deployment scheme, there are two major decisions to make: 1) What should be the average distance between neighboring sensor nodes? 2) What should be the deployment pattern or distribution (random or a regular pattern)? The requirement for low and balanced energy consumption, early detection, desire to achieve low channel contention, properly covering the region, the terrain and other parameters of the forest should be considered in making those decisions.

The average deployment distance between neighboring sensor nodes is an important parameter that affects the performance of a wireless sensor network deployed for fire detection. The time to detect a temperature increase at a node due to a fire is related with the distance of the node to the fire ignition location. Therefore, in order to reduce the expected fire detection time, the average distance between neighboring sensor nodes should be reduced. But this may contradict with the goal of reducing collisions which is expected to happen more when a network becomes denser. Hence, there is a trade-off between reducing the fire detection time and collision probability.

Some studies about spread characteristics of forest fires show that the time required for a sensor node to be aware of fire depends also 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]. The effects of such environmental factors on forest fires are investigated in National Fire Danger Rating System (NFDRS) [4]. NFDRS calculates a fire spread component (SC) value for a forest, which represents the forward spreading rate of a fire in meters per minute and which depends on fuel model of the forest, wind speed and slope of the zone.

Inspired by NFDRS, in our system, while determining the appropriate average distance between neighboring sensor nodes, we consider an *importance value* (I) for the forest as a parameter. The I value of a forest depends on how important the forest is to protect from fires. For example, a forest surrounding a cultural heritage site may be considered to be more important than a forest that is on top of a mountain. The importance value also depends on the spread component of the forest. A forest with higher fire danger rate, i.e. with a larger SC value, is considered again to have a larger importance value. The required maximum fire detection time (T, in seconds), the initial energy of sensor nodes (E, in Joules), and the required network lifetime (N, in seconds) are some other parameters that may affect the decision of what the average distance between neighboring sensor nodes should be. Considering all these different parameters into account, we propose the following approximate formula to determine the average distance d (in meters) between neighboring sensor nodes:

$$d = \alpha \frac{ET}{NI^2} \tag{1}$$

where α is a normalization factor determined empirically. As seen from the formula, we propose the average distance to be proportional to initial energy level of nodes (E) and the required fire detection time (T), and to be *inversely* proportional to the required network lifetime (N) and the square of the importance value (I) of the forest. I is a unit-less parameter that can have a value between 1 and 10 (1: not important at all; 10: of maximum importance), and we propose squaring the I value to have more effect on the result compared the other factors. The unit of α is meter/Joule. Note that, the value of α is to be found experimentally, which is not focused in this paper.

As mentioned earlier, another important factor that affects the performance of a fire detection WSN system is the deployment pattern of sensor nodes. Two general approaches can be considered to define the deployment pattern: 1) regular deployment, or 2) random deployment. In case of regular (homogeneous) deployment, nodes are deployed according to a regular pattern and we have nearly equal distance between neighboring nodes. Therefore, all sensor nodes transmit their messages to similar distances. This leads to balanced transmit energy consumption throughout the network. In random (non-homogeneous) deployment, nodes are deployed randomly (from a plane maybe) without following a regular pattern, hence the distance between two neighboring nodes is a random value, which may or may not be uniformly distributed. In this case, some sensor nodes may have quite distant neighbors and therefore may have to transmit to longer distances. Since the transmit energy consumption increases exponentially with the distance, those sensor nodes will consume much more energy due to transmissions and therefore will run out of energy earlier. Transmitting to longer distances to reach to some neighbors may also increase the interference on other nodes and may cause an increase in the collision probability. Additionally, an increased distance between neighboring nodes at some locations of the forest may increase the fire detection time at those locations.

As regular deployment pattern alternatives, two popular layout models are proposed: square layout and hexagonal layout [19]. In square model, the region is considered to be divided into squares (a grid of squares) and sensor nodes are placed at the corners of squares and cluster-heads are placed in the centers of squares. In such a deployment, the maximum distance between the fire ignition location and the closest sensor node will be $\frac{a}{\sqrt{2}}$, where a is the side-length of the squares.

In hexagonal layout, the region is considered to be divided into hexagons. Sensor nodes are placed at the corners of the hexagons and cluster-heads at the centers. In this case, the maximum distance between a fire ignition location and the closest sensor node will be $\frac{b}{2}$, where b is the distance between two far most corners of a hexagon. Sample deployments according to square and hexagonal patterns are shown in Figures 1 and 2, respectively. The square layout has less sensor nodes per cluster compared to the hexagon layout. Therefore, each cluster-head is less loaded, but for a fixed number of sensor nodes, it needs more cluster-heads. With less sensor nodes per cluster, the congestion will be managed better in the square model. It is also more robust layout due to having more cluster-heads. Therefore, we prefer square layout and use it in our simulations.

With irregular (random) deployment, we can not guarantee a maximum distance between a fire ignition location and the closest sensor node. Therefore, we expect the distance between a fire ignition location and the closest sensor node to be higher in random deployment. So, regular deployment is preferable if it is possible to do so.

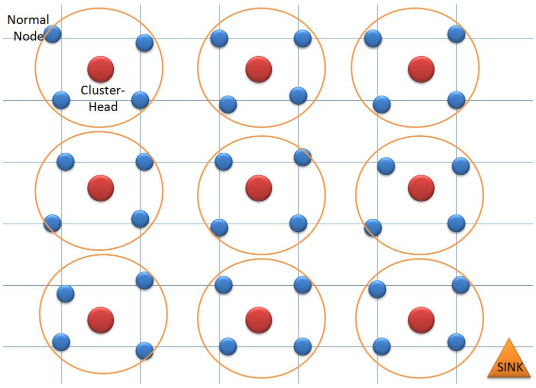


Figure 1: A sample square layout network architecture

Even though we may want to deploy according to a regular pattern, however, considering the geography of the region, it is highly possible that in some cases we may not be able to deploy all sensor nodes with a regular grid pattern. There will be some nodes which have to be deployed to distant locations from other sensor nodes because of the geography of the area (for example, because of a small lake inside the forest). Those distant sensor nodes will have to send their messages to longer distances and therefore will consume more energy than the other nodes. In order to remedy the problem to some degree, those sensor nodes may be deployed with higher initial energy levels if possible.

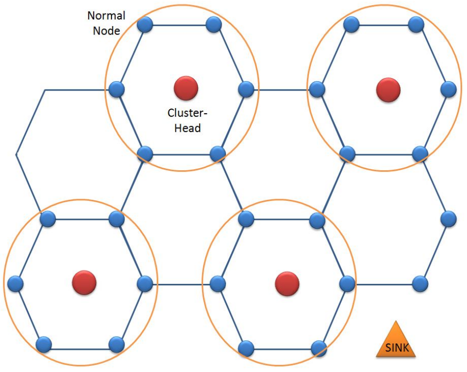


Figure 2: A sample hexagonal network architecture

3.2 Network Architecture and Topology Design

Efficient and effective operation of a WSN depends also on the architecture and logical topology of the network. We designed the architecture and logical topology of our WSN considering the goals of a fire detection system and limitations of wireless sensor nodes.

There are two possible alternatives for the network topology: flat and hierarchical. In flat topology, sensor nodes run in a totally distributed manner with equal responsibilities. In a hierarchical clustered topology, some nodes are designated as cluster-heads with more responsibility to control other member (ordinary) nodes. We performed several tests (described later in Section 4.2.2) and observed that use of a clustered topology provides important advantages for fire detection application of sensor networks. Hence, we propose a clustered logical topology for the network to properly and adaptively control the sensor nodes under various conditions. Clustered topology has benefits in terms of achieving effective control of nodes depending on changing conditions, rapid reaction to fire threat, and energy and bandwidth efficiency. It also enables data aggregation or data fusion [23] to be performed at well-designated nodes, i.e. cluster-heads. In this way, the volume of traffic carried inside the network can be reduced and faster reaction to urgent events can be done. This is especially useful for fire detection applications, because most of the time the maximum temperature from a region is needed instead of individual temperature values from all sensor nodes. Moreover, cluster-heads can apply smart scheduling and adaptive transmissions to reduce the load on sensor nodes closer to the sink.

In a clustered topology, a specific number or percentage of sensor nodes (where this depends on some system parameters and deployment) will form a group (a cluster) and connect to a cluster-head which will have some additional responsibilities. The cluster-heads may have superior physical capabilities as well, such as being equipped with a GPS module or having larger memory, processing, and energy resources. They should also have the capability to adjust their transmit power to transmit to longer distances when necessary. An example illustration of the clustered network architecture is shown in Figure 1.

3.3 Environment Aware Intra-Cluster Communication Protocol

In a clustered network architecture, protocols for intra-cluster communication and inter-cluster communication have to specified. In this section we describe our intra-cluster communication protocol (communication inside a cluster), and in the next section our inter-cluster communication protocol (communication among the cluster-heads).

WSN protocols should be designed to be adaptive to the current environmental conditions, like the current season or the current daily average temperature, and also to whether there is a fire threat at the moment or not. In times when there is no fire and the risk of fire is quite low, the network 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. This should be done, of course, without compromising the fire detection capability. In a possible fire threat time or as the fire spreads, however, energy optimization will be a less critical goal for the network, and reacting to fire rapidly and delivering data to sink as fast as possible will be a more critical issue. Therefore, we designed our cluster communication protocols to be adaptive to changing environmental and weather conditions and whether there is fire threat at the moment.

Our cluster communication protocols are different than the clustering work in literature, because our clustering protocols are designed specifically to be effective and efficient for fire detection and monitoring. We are more concerned about how a clustered topology is employed, operated and utilized rather than how topology is formed. Our communication protocols have unique features designed for fire detection applications, such as having adaptive mechanisms to react to fires quickly and energy-efficiently.

Our intra-cluster communication protocol, that provides communication in a cluster among the cluster members and the cluster-head, consists of four phases: initialization phase (which involves also defining message sending sequence), risk-free time (regular time) phase, fire-threat (fire-time) phase, and progressed-fire phase. Each phase is implemented via a set of messages exchanged between a cluster-head and its member nodes. Next, we detail the actions performed in each phase.

When booted up, sensor nodes start in the *initialization phase*. In this phase, member nodes of a cluster are initialized and set up to connect to their cluster-heads. Since the focus of our paper is not formation of clusters, we assume that the clusters are statically formed and configured. We leave a dynamic clustering approach for fire detection to be out of scope of the paper and as a future work.

When a cluster-head has all members connected, it assigns a data message sending sequence to be followed by the member nodes to coordinate access to the shared wireless channel and avoid collisions. This sequence (time slot information) of each member is sent to the member along with a frequency (duration) parameter which indicates how frequently a sensor node will send data messages to its cluster-head. This frequency is a dynamically adjusted parameter that depends on the current *fire danger rate* calculated by each cluster-head. It is a time and space dependent parameter. The current value of the fire danger rate at a cluster-head indicates the risk of fire at that time and at that location. A higher rate will cause more frequent sending of data messages from the sensor nodes to the cluster-head.

Additionally, a cluster-head sends information about fire threshold levels to each of its connected nodes. Using these thresholds sensor nodes can determine the current risk level of fire. After a cluster-head sends all the required initialization information to the connected nodes, the next phase starts at the cluster, which is the risk-free time phase.

Nodes are in *risk-free time phase* during the times when the fire risk is low. At those times, the system adapts itself by decreasing its activity level, so that it can achieve energy efficiency without compromising the fire detection capability. The frequency of sending temperature data from sensor nodes to a cluster-head is lowered.

Additionally, in this phase sensor nodes can be put into sleep mode for a while in order to save more energy. The nodes in a cluster can be put into sleep in a Round Robin fashion, so that a balanced energy consumption is achieved. In sleep more, the sensing frequency of a sensor node is set to be very low. Sleep period and activity level can be made location-dependent, i.e., adaptive to the fire danger rate of a region.

When the temperature or humidity level exceeds the configured threshold level at a sensor node, the *fire threat phase* is started at the node. In this phase, a node interrupts the normal Round-Robin message sending sequence, and immediately and aggressively sends alarm messages, until its cluster-head sends back a message.

The cluster-head hearing an alarm message from one of its connected nodes takes the required actions to handle emergency situation. This involves issuing more time slots to the node that sent the alarm message. Additionally, besides average and min/max values of temperature and humidity, more information can be sent by a cluster-head for the sink to analyze the progress of the fire.

In *progressed-fire phase* is entered when there are some already damaged sensor nodes due to fire. When an ordinary sensor node has died due to fire, this can be detected by the cluster-head and appropriate action can be taken. What is more serious is death of a cluster-head. In this phase, the network copes with these kinds of situations. The system should be prepared for the incident where a cluster-head may not be able to perform its critical duties. There are two cases to consider: 1) a cluster-head can recognize the danger it faces and can take some actions before becoming non-functional; 2) a cluster-head suddenly dies because of fire.

In the first case, the cluster-head discerns the potential risk and selects the most suitable sensor node as the new cluster-head. The most appropriate member node can be selected by utilizing a risk-level (r) metric we propose:

$$r_i = \frac{T_d - t_i}{\triangle t_i} \times e_i. \tag{2}$$

Here, r_i is the risk level of a node i, T_d is the death temperature level of regular nodes, t_i is the current temperature of the node, $\triangle t_i$ is the change in the temperature of the node in the last period, and e_i is the remaining energy level of the node.

In the second case, a cluster-head may suddenly die because of a very quick temperature increase or a weather incident like lightning. To cope with such a case, at the beginning of fire-threat phase, a cluster-head sends a cluster-information message to its member nodes so that important information about the cluster is replicated in the ordinary nodes as well. Then, when a cluster-head suddenly dies, the ordinary nodes detect this by utilizing the lack of ack messages (in response to regular information messages) sent from the cluster-head. The first sensor node that detects the failure of the cluster-head informs all the other nodes in the cluster about the situation. Then, the nodes exchange some parameter values among themselves (like remaining energy levels) to make a decision about the next cluster-head and use the metric defined in Equation 2 to decide about the most eligible node for being cluster-head. The most eligible node selects itself as the new cluster-head and informs the other nodes about the selection.

3.4 Environment Aware Inter-Cluster Communication Protocol

Our inter-cluster communication (i.e., cluster-head level communication) protocol is used to carry data messages obtained at the cluster-heads to the sink node via multi-hop forwarding. The protocol has two main goals: balancing energy consumption among the cluster-heads and forwarding the 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.

In the *initialization phase*, cluster-heads first determine the (routing) paths that will connect them to the sink node. Then, a message forwarding time table (transmission schedule) is created by using these paths. The table provides the time when each cluster-head will send its aggregated data to the next (upstream) cluster-head. Up to that time, a cluster-head may collect data from the downstream (children) cluster-heads. It is possible that in a given time, more than one cluster-head can transmit its data to its next cluster-head. In this way, we have concurrency in the network to carry data towards the sink node, which reduces the overall time required in a round to collect all data. There are various ways of determining routing paths and time scheduled. One of these schemes proposed in the literature can be used. Here we are not considering a specific algorithm. Upon finishing the initialization, the network goes into the risk-free time phase.

In the *risk-free time phase*, the operation of a cluster-head is quite simple. It receives data from the member nodes and other cluster-heads, performs processing and aggregation, and sends data to the next upstream node on the way to the sink node. For fire detection, the maximum level of the temperature and the minimum level of the humidity are important indicators of a possible forest fire, therefore aggregation is done using max and min functions. From time to time, however, the instantaneous temperature values and min/max temperature values may also be sent by a cluster-head to the sink node so that the center can generate a temperature map of the forest.

When a cluster-head receives an alarm message from one of its connected nodes, it goes into the *fire-threat phase*. In this phase, a cluster-head tries to deliver such an alarm message to the sink node as soon as possible, if it is the first cluster-head to detect the occurrence of the fire. Additionally, it informs all cluster-heads about the fire threat. This can be achieved by a network-wide broadcast.

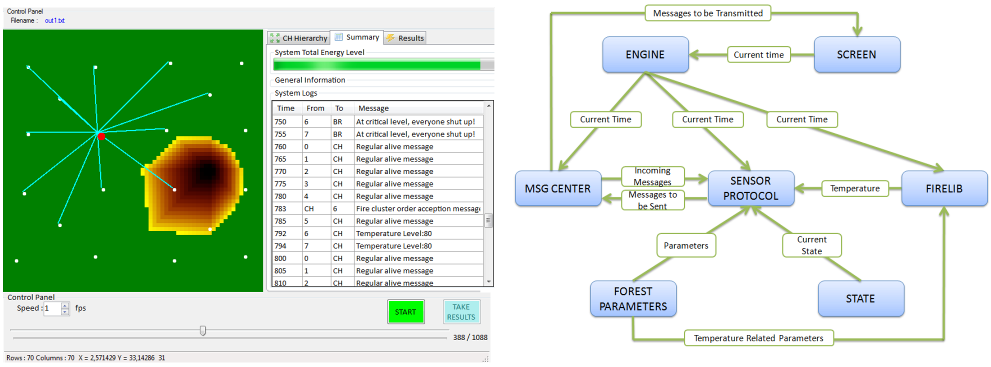
A cluster-head continuously evaluates the local fire-threat level by monitoring the sensed values it receives from the cluster members. Besides, it can receive a broadcast (global) fire threat message from the other cluster-head nodes. The cluster-head can compare the local threat level with the global threat level and adjust the priority of its messages. The priority also depends on how the local threat value is changing over time.

4 Simulation Experiments and Results

To evaluate our framework, we designed and implemented a custom simulator and performed extensive simulation experiments. In this section we first describe our simulator and then provide our experimental results and discussions.

4.1 Our Simulator

We developed a custom simulator that can simulate a wireless sensor network monitoring a forest and detecting fires. The simulator is developed in C# using Microsoft Visual Studio 2008 development environment. Figure 3(a) shows a sample screen-shot of the simulator. It can accept a set of input parameters from the user before a simulation run. When a simulation run is started, the simulator can simulate the actions of the sensor nodes, the protocol operations, the transfer of sensory data and events to a center, the occurrence of a fire and spread of fire, and the operations of the network to detect the fire and inform the center about the fire and fire spread. When a simulation run is completed, results are produced into a file that can be used to evaluate the proposed system components and protocols.



Time	From	To	Message
750	6	BR	At critical level, everyone shut up!
755	7	BR	At critical level, everyone shut up!
760	0	CH	Regular alive message
765	1	CH	Regular alive message
770	2	CH	Regular alive message
775	3	CH	Regular alive message
780	4	CH	Regular alive message
783	CH	6	Fire cluster order acception message
785	5	CH	Regular alive message
792	6	CH	Temperature Level:80
794	7	CH	Temperature Level:80
800	0	CH	Regular alive message
805	1	CH	Regular alive message
810	2	CH	Regular alive message

(a) A sample screen-shot from the simulator

(b) Components of the simulator

Figure 3: Our Simulator

Our simulator consists of the following components (Figure 3(b)):

1. *FireLib:*

   This component uses and processes the output of a fire simulation library called FireLib [11] that we integrated into our simulator. The FireLib library is already available and is open source. It is a C library. It estimates the spread behavior of forest fires. It presents an application programming interface (API) for fire growth modeling. Firelib library produces as output an ignition time table (i.e,. an output map) showing when fire starts at each cell (the region is divided into cells). This output map, however, just contains time and location related information. It does not provide temperature information. Calculating the temperature level of each cell at different times is the job of our FireLib component. For this calculation it uses two additional parameters: required temperature to start a fire, and temperature increment value with respect to a fixed time interval. Through these parameters and the start time of fire at each cell, the FireLib component can produce the temperature values for each cell depending on time. In this way, it generates a time-varying temperature map for a given region.

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

3. *Message center*: When the sensor nodes generate data, they forward their messages destined to one or more destinations to the message center. The message center makes the respective destination sensor nodes aware of their incoming messages.

4. *Sensor node*: The sensor node component performs most of the important actions in the simulator by following a protocol that can be implemented as a separate component.

5. *State:* This component keeps state information about the sensor nodes. States are decided and changed according to the protocol. A state includes many variables.

6. *Parameters of the simulator:* There are several inputs that have to be specified for the simulator engine. Some of these parameters are deployment related such as the layout and the average distance among sensor nodes. The parameters related with sensor node properties are the maximum range of a node and the initial energy level. The remaining parameters are the power consumption values and time/tick values.

After specifying the values of these parameters, a simulation run can be started.

4.2 Simulation Results

We performed extensive simulation experiments to evaluate our framework. Since our framework consists of many schemes, we evaluate each scheme and its design decisions in a separate sub-section below.

4.2.1 Sensor Deployment Scheme

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

Effect on energy consumption:

We have regular or random deployment choice which closely affects how energy is consumed in the nodes and in the network. To study the difference quantitatively between these two main deployment approaches, we did experiments by simulating these approaches with 20 sensor nodes (Figure 4). All nodes start with the same initial energy level. The average distance between neighboring nodes is varied between 5 and 35. For testing scalability, we also performed experiments with larger numbers of nodes; the above experiment, for instance, is repeated with 500 nodes and the results are shown in Figure 5. Comparing Figure 4 and Figure 5, it can be stated that the trends are the same no matter the network size is.

Our simulation results show that regular (deterministic) deployment scheme is preferable when low-energy consumption is considered as seen in Figure 4 and Figure 5, which show the average remaining energy in the sensor nodes at a certain time after the simulation is started (initially all nodes have the same energy). As the figures show, the average remaining energy is higher when sensor nodes are deployed using a regular pattern. In other words, regular deployment causes less energy to be consumed in the network when compared with random deployment. In random deployment, even though some sensor nodes may be closer to a cluster-head compared to the regular deployment and hence may consume less energy, there will be usually distant nodes whose high energy consumption will outweigh the advantage obtained from these low energy consuming close-by nodes. This is because, the energy consumption at a node is inversely proportional with at least the square of the distance to where the node makes transmissions. We are assuming that nodes have transmitters that are capable of power-adjustment, hence consume just enough energy to reach to their receiver. Therefore, the energy consumed becomes higher in random deployment compared to regular deployment. Additionally, we can see from the same figure that as the average distance between neighboring sensor nodes increases, the difference between the energy consumption of the approaches increases as well.

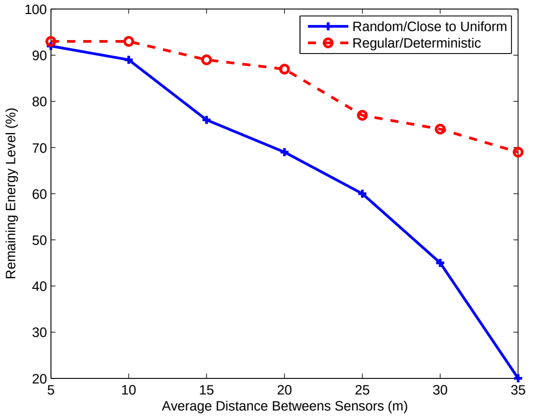


Figure 4: Remaining energy levels of regularly deployed and randomly deployed sensor nodes. There are 20 nodes in the network.

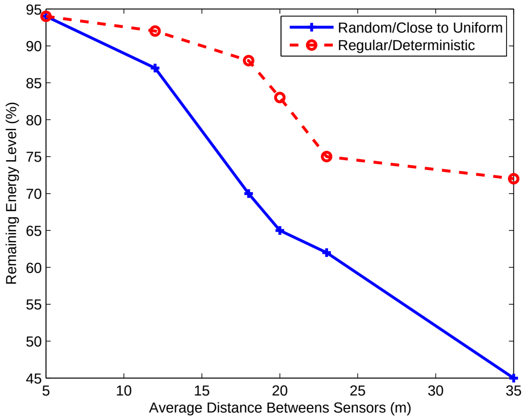


Figure 5: Remaining energy levels of regularly deployed and randomly deployed sensor nodes. There are 500 nodes in the network.

Another experimental result presenting the relationship between a sensor deployment scheme and energy consumption is shown in Figure 6. In this figure, it is analyzed how balanced the energy consumption is among the nodes of a cluster for both approaches. The figure shows how the difference between remaining energy levels of two nodes of a cluster changes over time.

One node is selected to be a close-by node, the other one to be a far-away node. In the deterministic case, the distance of each sensor node in a cluster to the cluster-head is nearly the same (not exactly the same, since in practice it is not possible to place all sensor nodes in exact grid-corner locations), and therefore they consume nearly the same amount of energy with each transmission. In random deployment, on the other hand, a cluster member-node that is far away from the cluster-head consumes much more energy compared to a member-node that is closer to the cluster-head. Therefore, the difference between the remaining energy levels of nodes of a cluster increases with time. As the figure shows, the regular deployment causes a more balanced energy consumption among the members of a cluster.

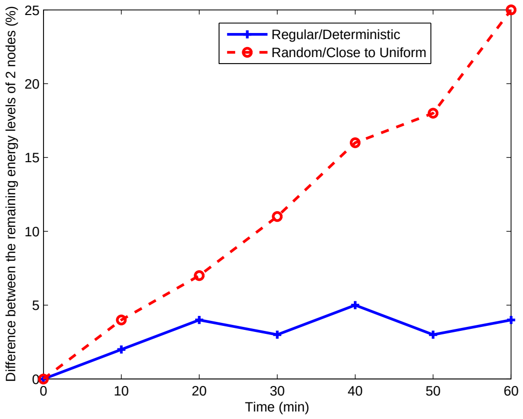


Figure 6: Difference between the energy levels of two nodes in the same cluster.

Effect on fire detection time:

When early detection goal is considered, again regular deployment scheme is more successful as can be seen in Figure 7. The figure shows the average distance between a fire ignition location and a closest sensor node for various values of the average deployment distance between neighboring sensor nodes which varied between 5 and 35. 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 that 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 neighboring sensor nodes increases, the difference between the performance of the two approaches (regular and random) gets larger as well.

The distance between a fire ignition location and a sensor node affects the time required for the heat waves to reach to the node. Figure 8 provides

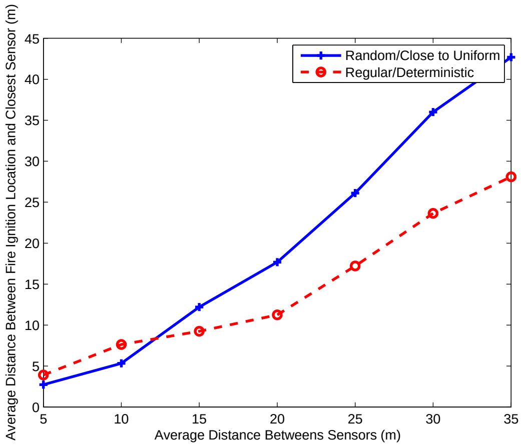


Figure 7: Distance between fire ignition and closest sensor in regular and random deployment schemes

the results of our experiment performed to see how much time is required for a sensor to detect a fire after it has started depending on the distance between fire ignition location and the node position. We can observe from the figure that it takes more than 10 minutes for a sensor node to sense the fire threat if the distance is greater than or equal to 20 m. The figure suggests that, in order to detect a fire as early as possible, the sensor nodes should be deployed in such a manner that the distance of any point of the field to its closest sensor node should be less than 20 m.

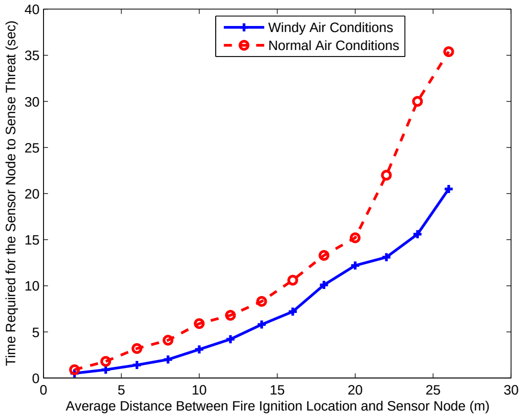


Figure 8: Time required for the sensor nodes to sense the fire threat for various distances.

4.2.2 Effects of Logical Topology Choices

A clustered network architecture is proposed in our framework for the purpose of more efficient 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 proposed to have a hierarchical, clustered topology for a sensor network for fire monitoring and detection. We compared the clustered topology with flat topology (non-clustered topology) in terms of total traffic volume and essential traffic volume carried in the network. As mentioned earlier, clustered architecture facilitates in-network processing and data aggregation, and therefore we expect unnecessary (not essential) traffic volume to be dramatically reduced when clustered architecture is used. For this purpose, we measured the ratio of essential (critical) messages to all the messages carried in the network. We made these measurements for both clustered topology and flat topology. Figure 9 provides the results. As seen in the figure, clustering not only reduces the total volume of traffic carried in the network dramatically, but also increases the percentage of the essential traffic inside the total volume.

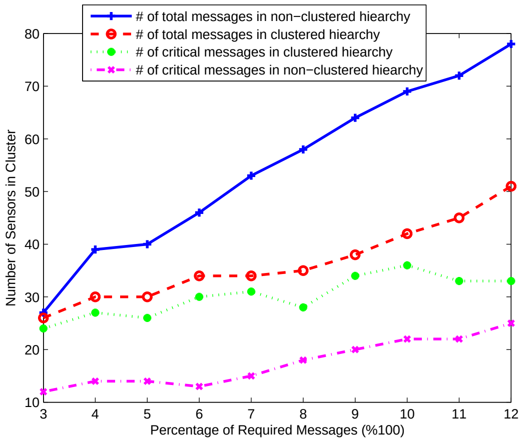


Figure 9: Number of messages sent to the sink when local computation at the cluster level is applied compared to when no local computation is performed

4.2.3 Environment/Situation Aware Communication Protocols

We designed our protocols to be environment aware. That means, design decisions, such as the protocol actions, how frequently data is sensed, etc.

depend on the geographic location of the forest, the climate and the season. We now study the effect of our environment-aware protocols on energy consumption and fire detection time.

Effect on energy consumption:

We compare our approach against a base approach that is not considering the current environmental conditions (such as month of the year, etc.) while adjusting the frequency of sensing and data communication. Such a scheme is not adjusting the activity level in the network.

Figure 10 shows the energy consumption with our scheme and the base scheme. As the figure shows, the energy consumed in the base scheme remains at similar levels throughout the year. However, in our scheme the energy consumption changes depending on the season. Our scheme keeps the activity level of sensor nodes low in months when the risk of fire is quite low, like the months of winter season. Usually, we do not have fires in winter times. At those times our scheme reduces the activity level to very low values without harming the effectiveness of the network to detect fires. This is achieved, for example, by sending regular inform messages less frequently to cluster-heads. In summer times, however, our scheme keeps the activity level quite high and therefore consumes more energy.

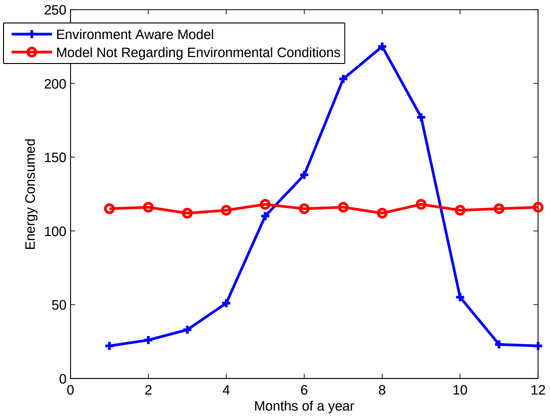


Figure 10: Energy consumption levels of environment aware and base models throughout the year.

This high energy consumption of summer times is compensated with low energy consumption at winter times as shown in Figure 11. The figure provides the cumulative energy consumption throughout the year for both our scheme and the base scheme. As shown in the figure, the total energy spent throughout the year is less with our approach, even though it causes more energy consumption in summer times.

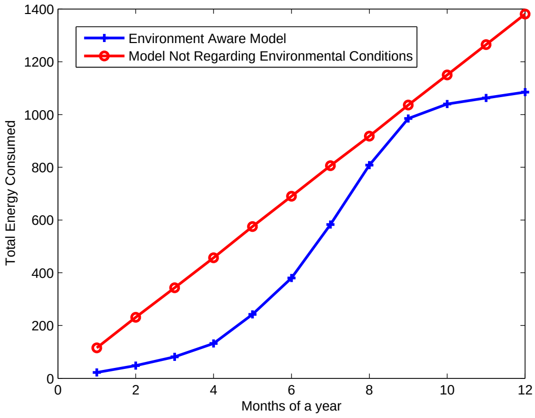


Figure 11: Cumulative value of energy consumption level of environment aware and base models throughout the year.

To evaluate the benefit of our scheme in a more concrete way, we define and use a *weighted energy consumption* metric (WE) that considers not only energy consumption in a month of the year, but also the risk level of fire in that month. Each month of year has a different fire risk level. Our new metric weighs the energy consumption in a month with the risk-level of the month. The formula used to compute the new metric value is as follows:

$$WE_i = FT_i \times E_i, \qquad 1 \leq i \leq 12 \tag{3}$$

where FT_i is the fire risk level of month i, and E_i is the total energy consumed by sensor nodes in month i.

Figure 12 compares the two schemes using this metric. In the base scheme, the WE value is very high for risk-free months. This is an indication of the unnecessary high energy consumption of the base scheme in risk-free months. As the figure shows, the benefit of our scheme is emphasized more when this weighted energy consumption metric is used.

Effect on fire detection time:

In our scheme, during a high-risk period, the activity level of nodes is increased to sense and react to a possible fire threat more quickly. Figure 13 shows the effect of our scheme on fire detection time. The figure plots the fire detection time depending on the month of the year for our approach and for the base approach which is not using adaptive activity adjustment. The figure shows that the base approach causes a nearly constant fire detection time throughput the whole year. But the detection time of our scheme changes

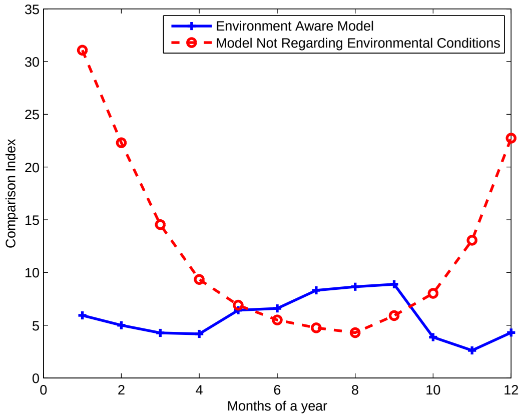


Figure 12: Energy consumption level × the fire risk level values of environment aware and base models throughout the year.

over the year. It is larger in winter times when the risk of fire is very low, but much smaller in summer times when the risk of fire is quite high. Hence, our approach can react to a possible fire threat much faster in summer times.

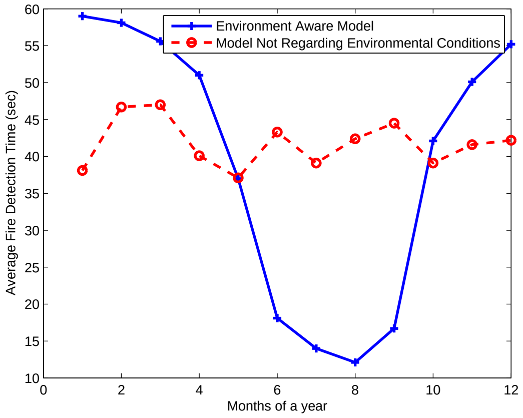


Figure 13: Fire detection durations of environment aware and base models throughout the year.

For a more concrete evaluation of the benefit of our scheme again compared to the base scheme, we use a new metric, *weighted fire detection time* (WT) which is weighting the fire detection time in a month with the risk-level of that month. Below is the formula for the computation of the metric:

$$WT_i = FT_i \times T_i, \qquad 1 \leq i \leq 12 \tag{4}$$

where FT_i is the average fire threat level of month i, and T_i is the average fire detection time of regular nodes in month i.

Figure 14 provides the values of WT metric for both approaches for each month of the year. A smaller metric value means better performance. As can be observed from the figure, our environment-aware scheme performs much better than the base scheme. During risk-free months, the WT value is nearly the same for both schemes. However, in high-risk months, the value of this metric is much smaller (i.e. much better) for our scheme.

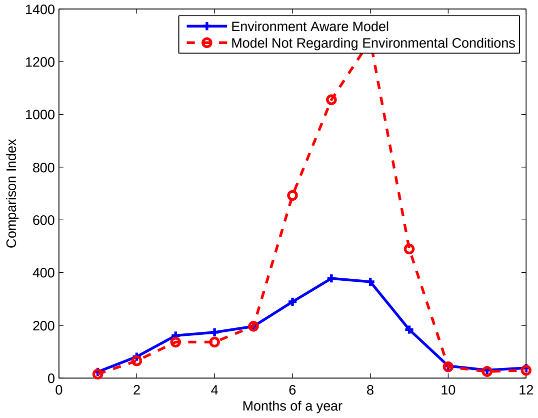


Figure 14: Fire detection duration × the fire risk level values of environment aware and base models throughout the year.

Our scheme is also acting adaptively when a fire occurs. That means it is situation aware. Both inter- and intra-cluster communication schemes start behaving differently with a fire occurrence to react to fire more quickly. For example, our inter-cluster communication scheme is designed in such a way that cluster-heads rapidly propagate the fire alarm messages to the sink in case of a fire threat. And in our intra-cluster communication scheme, when a sensor node senses a fire threat, it immediately broadcasts an emergency packet. Therefore, the cluster-head becomes aware of the threat quickly and intra-cluster scheduling is adapted accordingly. We now compare our scheme with a base version that does not have those features. We again measure the fire detection time. The measurement results are reported for various values of two clustering parameters: 1) number of clusters in the network, and 2) number of nodes per cluster.

Figure 15 compares our adaptive (environment-aware) scheme against the base (non-adaptive) scheme for various network sizes, in terms of the number of clusters. We performed experiments with up to 1000 clusters. We observed that as the number of clusters in the network increases, our scheme provides more improvement in the performance. That means, for large networks, the advantage of our adaptive scheme against a base scheme is pronounced more. This is because cluster-heads rapidly propagate fire alarm messages to the sink in case of a fire threat. Additionally, we observe that even when the network becomes quite large (i.e., 1000 clusters), the detection time of our environment-aware scheme is around 3 minutes at most. This is nearly half of the detection time of the base scheme which is around 6 minutes.

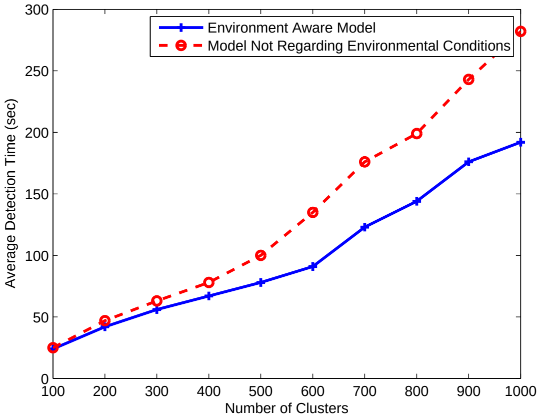


Figure 15: Fire detection durations of environment-aware and base models as the number of clusters in the network varies.

Finally, Figure 16 compares our adaptive scheme against the base scheme for various cluster sizes. Again our scheme performs much better when the cluster size is increased. As mentioned above, in our scheme, in case of a fire, a sensor node immediately broadcasts an emergency packet without waiting for its turn to come according to the normal packet transmission schedule. Therefore, the respective cluster-head learns about the threat as quickly as possible.

5 Conclusions and Future Work

In this paper, we present a general framework for a wireless sensor network to be used for forest fire surveillance and detection. Our framework considers all parts of the life cycle of a wireless sensor network system that is specialized

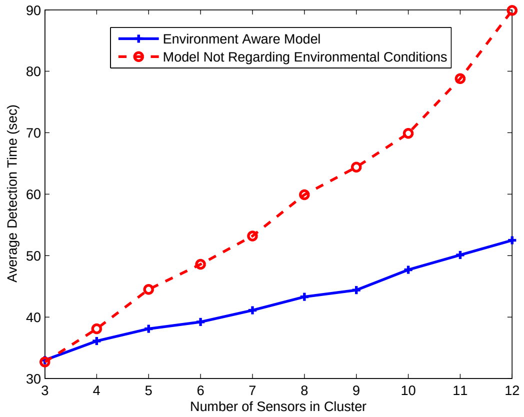


Figure 16: Fire detection durations of environment aware and base models as the number of sensors in a cluster varies.

for forest fire detection. While considering the early detection of forest fires as the major goal, we also aim to construct a system that regards the low energy capacity of sensor nodes and the difficult environmental conditions that may adversely affect the network operation and performance.

Our framework incorporates the design of four main components of a wireless sensor network: the deployment scheme, the logical topology and architecture of the network, the intra-cluster communication scheme, and the inter-cluster communication scheme. Considering the system goals, several design decisions are evaluated for each part, starting from the sensor node deployment scheme and ending with cluster communication protocols.

We evaluated our proposed scheme in terms of energy consumption and effectiveness in detection fires. We observed that our system can provide both effective and efficient operation: consuming less energy without sacrificing the quick reaction capability. We conclude that season, environment, and weather adaptation is very important for a wireless sensor network and can significantly reduce energy consumption. We additionally conclude that clustered hierarchy has benefits in terms of data aggregation, management capability, energy efficiency and better coordination. Moreover, whenever possible regular deployment should be used, but when it is not possible, random deployment can be used with careful consideration of energy consumption at the nodes that are located at distant locations.

The system is open 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 of the topics which can be investigated in future studies.

Acknowledgments

We thank European Union for partially supporting this work with FP7 Framework Program Project FIRESENSE-244088.

References

[1] *Republic of Turkey, Ministry of Environment and Forestry, General Directorate of Forestry*, http://www.ogm.gov.tr/, Last accessed August 2011.

[2] *Washington State University Department of Natural Resource Sciences*, http://www.ruraltech.org/, Last accessed August 2011.

[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 2011.

[5] *Aerovision Web Page,* http://www.aerovision-uav.com/, Last accessed August 2011.

[6] *The Advanced Very High Resolution Radiometer (AVHRR) Web Page,* http://noaasis.noaa.gov/NOAASIS/ml/avhrr.html, Last accessed August 2011.

[7] *B.C. Fire Lookout Towers Web Page,* http://www.firelookout.com/bc.html/, Last accessed August 2011.

[8] B. U. Toreyin, Y. Dedeoglu, U. Gudukbay, A. E. Cetin. *Computer Vision Based Method for Real-time Fire and Flame Detection*, Pattern Recognition Letters, 27(1):49-58, 2006.

[9] *The Moderate Resolution Imaging Spectroradiometer (MODIS) Web Page,* http://modis.gsfc.nasa.gov/, Last accessed August 2011.

[10] *National Center for Landscape Fire Analysis (NCLFA),* http://firecenter.umt.edu/, Last accessed August 2011.

[11] *FireLib Public Domain Software for the Wildland Fire Community,* http://fire.org/, Last accessed August 2011.

[12] 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. 477-484, 2006.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

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

[19] 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.

[20] 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.

[21] 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.

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

[23] Dave L. Hall, James Llinas, *Introduction to Multisensor Data Fusion*, Proc. of IEEE , Vol. 85, No. 1, pp. 6 23, Jan 1997.

[24] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci Wireless sensor networks: a survey, Computer Networks, Volume 38, Issue 4, 15 March 2002, Pages 393-422.

[25] Giuseppe Anastasi, Marco Conti, Mario Di Francesco, Andrea Passarella, Energy conservation in wireless sensor networks: A survey, Ad Hoc Networks, Volume 7, Issue 3, May 2009, Pages 537-568.

[26] Jennifer Yick, Biswanath Mukherjee, Dipak Ghosal, Wireless sensor network survey, Computer Networks, Volume 52, Issue 12, 22 August 2008, Pages 2292-2330.

[27] Alan Mainwaring, David Culler, Joseph Polastre, Robert Szewczyk, John Anderson, Wireless sensor networks for habitat monitoring, Alan Mainwaring, David Culler, Joseph Polastre, Robert Szewczyk, John Anderson, September 2002.

[28] Chi-Fu Huang, Yu-Chee Tseng, Hsiao-Lu Wu, Distributed protocols for ensuring both coverage and connectivity of a wireless sensor network, ACM Transactions on Sensor Networks (TOSN), Volume 3 Issue 1, March 2007.

[29] Stanislava Soro, Wendi B. Heinzelman, Cluster head election techniques for coverage preservation in wireless sensor networks, Ad Hoc Networks, Volume 7, Issue 5, July 2009, Pages 955-972.

[30] N. Dimokas, D. Katsaros, Y. Manolopoulos, Energy-efficient distributed clustering in wireless sensor networks, Journal of Parallel and Distributed Computing, Volume 70, Issue 4, April 2010, Pages 371-383.

[31] Ameer Ahmed Abbasi, Mohamed Younis, A survey on clustering algorithms for wireless sensor networks, Computer Communications, Volume 30, Issues 14-15, 15 October 2007, Pages 2826-2841.

[32] Chuan-Ming Liu, Chuan-Hsiu Lee, Li-Chun Wang, Distributed clustering algorithms for data-gathering in wireless mobile sensor networks, Journal of Parallel and Distributed Computing, Volume 67, Issue 11, November 2007, Pages 1187-1200.

[33] Wendi Rabiner Heinzelman, Anantha Chandrakasan, Hari Balakrishnan, Energy-efficient communication protocol for wireless microsensor networks, HICSS, 2000.

[34] Min-Woo Park, Jin-Young Choi, Young-Ju Han, Tai-Myoung Chung; An Energy Efficient Concentric Clustering Scheme in Wireless Sensor Networks, NCM '09, Page(s): 58 - 61, 2009.

[35] Song Ci, Mohsen Guizani, Hamid Sharif, Adaptive clustering in wireless sensor networks by mining sensor energy data, Computer Communications , Volume 30 Issue 14-15, October 2007.

[36] Renita Machado, Wensheng Zhang, Guiling Wang, Sirin Tekinay, Coverage properties of clustered wireless sensor networks ACM Transactions on Sensor Networks (TOSN) , Volume 7 Issue 2, August 2010.