Fire Detection and Localization Using Wireless Sensor Networks

Alireza Khadivi and Martin Hasler

Ecolé Polytechnique Fédérale de Lausanne
School of Computer and Communication Sciences
Laboratory of Nonlinear Systems
CH 1015, Lausanne, Switzerland
{alireza.khadivi,martin.hasler}@epfl.ch

Abstract. A Wireless Sensor Network (WSN) is a network of usually a large number of small sensor nodes that are wirelessly connected to each other in order to remotely monitor an environment or phenomena. Sensor nodes use the data aggregation method as an effective tool for estimating the desired parameters accurately and trustfully. In this paper, we have applied a cellular-automata-like algorithm and an averaging consensus algorithm for fire detection and localization with sensor networks. Indeed, when fire is detected somewhere in the network, our algorithm makes aware all the nodes in the network with a very short delay. Afterwards, the algorithm estimates the parameters of the circle surrounding the fire. To simulate the fire outbreak and the reaction of the sensor network equipped with our algorithm, we enabled the data exchange between the fire simulation software FARSITE and the communication software Castalia. The results show that our method detects the fire rapidly and monitors the extension of the fire in real time. The information about the outbreak and the extension of the fire is available from every live sensor in the network, even when part of the sensors are destroyed by the fire.

Keywords: Wireless Sensor Network, Consensus Algorithm, Fire Monitoring.

1 Introduction

Technological advances in microelectronics lead to very low power consumption electronic systems as well as RF modules with reasonable prices [1-3]. This creates new opportunities for environmental monitoring by wireless sensor networks. In a wireless sensor network, the sensors can exchange information repeatedly in order to come to some conclusion concerning the global environmental situation in the area the sensor network covers. In this paper the sensors measure temperature and determine collectively whether there is a fire in the area, and if so, where the fire is localized. This way of coming to a conclusion by a distributed algorithm before sending the information to a base station, instead of sending the information directly from each sensor separately is advantageous from the point of view of energy consumption and robustness, if the network is carefully designed.

N. Komninos (Ed.): SENSAPPEAL 2009, LNICST 29, pp. 16–26, 2010.

There are different strategies to transfer the aggregated data to the base station. In some networks, the sensor nodes first setup a multi-hop path and then send the data directly to the base station. In other networks, the sensor nodes first relay the data to a pre-specified Cluster-Head and then the cluster head sends the data to the base station. In this paper we do not elaborate on this question any further, but we just remark here that the aggregated data are available at each sensor in the network as long as it is alive and therefore many different read-out strategies can be realized, such as e.g. driving a vehicle to the edge of the area and reading from the nearest sensor.

In this paper, two distributed algorithms are discussed, one for fire detection and one for fire localization. Using the first algorithm, the entire sensor node population in the network finds out rapidly that there is a fire somewhere in the network, and by using the second one, one can gain information about the exact place and boundary of the fire from every sensor node in the network. In these algorithms, we assumed that each sensor node is location aware, i.e. it knows its coordinates (but not those of the other sensors) and it has a temperature sensor which measures the temperature of the in its position periodically.

In addition to studying the efficiency of the algorithm in ideal conditions, we also have taken into account non-idealities due to the implementation of the wireless network, such as collisions due to simultaneous media access. For this purpose, we have used the simulation software Castalia[4] which is specifically developed for wireless sensor network simulations. Realistic modeling of the channel is one of the major advantages of this software. The temperature of the environment, i.e. the input data for the sensor nodes, is generated by the FARSITE[5] fire simulator software. This software is well-known for modeling of the spreading of the fire depending on the parameters of the environment. At the end, the output data of Castalia is visualized by MATLAB. The result of simulation shows a rapid alarm spreading in the case of fire outbreak and a good estimation of the circle surrounding the fire. Compared with the behavior of the ideal network simulated by MATLAB, when there is no interference and loss in communication between nodes, a good agreement was found.

The rest of the paper is organized as follows. Section 2 and 3 describes the Fire Detection and Fire Localization algorithms, respectively. Section 4 presents the simulation results and Finally, Section 5 concludes the paper.

2 Fire Detection Algorithm

Since the sensor nodes have limited energy, most of the time, they are in a sleep mode. In this mode, there is a low number of packets traversing the network in order to ensure the connectivity of the network. But, when a node sensed a temperature greater than a pre-specified threshold, it wakes up first and second nearest neighbors and sends them a Fire-Detection message. It is worth to mention that they do not go anymore into sleep mode until the alarm is called off (not modeled here), or until they are burnt.

In order to save energy, the packet lengths of messages that are sent by sensor nodes are short. In each packet, 4 bytes are used for the address of the alarm initiator and 1 bit for the type of the packet. When the sensor network has concluded that there is a fire in the area, and this information has reached every sensor extra information

for finding the location and boundary of the fire is transmitted. The fire detection algorithm is as follows. Each sensor node in the network can be in 3 different states which are: No-Fire: the temperature measured by the sensor is below a pre-defined Temperature-Threshold (T-T) and no or not sufficiently many Alarms have been received from neighbor nodes.

- Fire-Detected: the temperature that is measured by the sensor is more than the T-T for two consecutive measurements and no or not sufficiently many Alarms are received from neighbor nodes.
- Fire-Alarm: there is a fire somewhere in the network that is observed by the sensor itself or by other sensor nodes.

There are two types of packets that are sent in Fire-Detection phase

- Fire-Detection massage: this massage is sent by sensor nodes that have detected the fire.
- Fire-Alarm message: this message is used for disseminating Alarm in the network.

All sensor nodes are initially in the No-Fire state. When a sensor has sensed a temperature above threshold in two consecutive measurements, it goes to the Fire-Detected state. Furthermore, it sends a Fire-Detection message to its first and second nearest neighbors. If a sensor node receives two Fire-Detection messages from two different nodes and also detects the fire itself, it changes its state to Fire-Alarm and sends a Fire-Alarm message to its neighbors. Furthermore, if a sensor node is in the Fire-Detected state and then receives a Fire-Alarm message from another node, it will go directly to the Fire-Alarm state. Also, if a sensor node is in the No-Fire state and its measured temperatures are more than T-T for two consecutive times while it has already received an Alarm message, then it will go instantly to Fire-Alarm state.

The sensor nodes that are in No-Fire state cannot go to Fire-Alarm state unless they receive two Fire-Alarm messages from two different neighboring sensor nodes. This method leads to dissemination of the Alarm to all the nodes in the network. This result is also verified by Castalia simulation software for a realistic communication channel. The state diagram of the algorithm is depicted in figure 1.

First way is adding the node ID of the original sender to its Fire-Detection message. The second option is to send the location of the sender instead of its node ID. We considered the second way in order that the sensors have a good estimate of the Fire-Start point. Having this location is helpful for improving the accuracy of the fire localization which follows fire detection. The way that the sensors estimate this location is as follows.

The sensors that are in No-Fire state and receive two Fire-Alarm messages, they assume the average of the received locations as the Fire-Start point. In general, the Fire-Start point is assumed to be the average of the locations in Fire-Detection and Fire-Alarm received messages before going to Fire-Alarm state. Also, if a sensor node detects the fire itself, it adds its location in the averaging process. At the end, the estimated Fire-Start point on different nodes is slightly different but, it is negligible and does not affect the performance of fire localization algorithm.

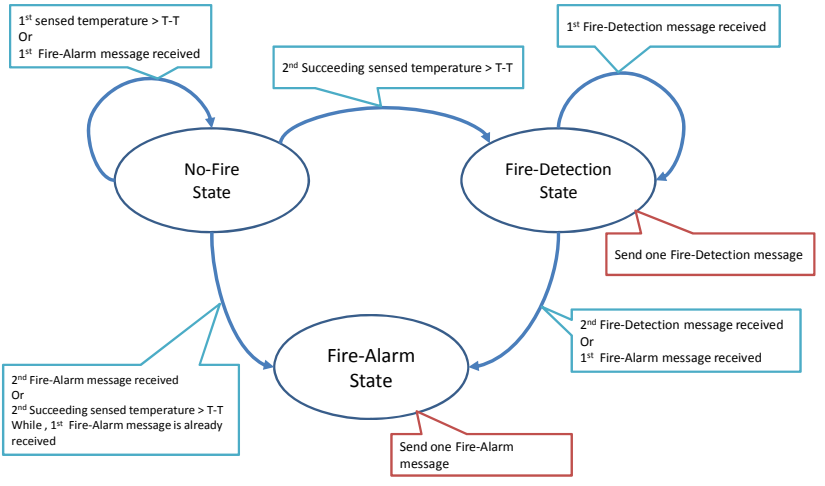

Fig. 1. State Diagram of the Fire Detection Algorithm

3 Fire Localization Algorithm

The Fire localization algorithm starts when the fire is detected in the network. In fact, it starts right after the end of Fire Detection phase. The purpose of this algorithm is to find a circle that surrounds approximately the fire. In order to estimate this circle, three parameters should be derived which are *X* and *Y*-coordinates of the center (*Z*) as well as the radius (*r*). Also, these parameters should be accessible from the all nodes in the network. A simple consensus algorithm [6] is used to estimate the circle parameters in a distributed manner. A similar approach to find parameters of functions by consensus has been described in [7]. The algorithm is as follows.

The sensor nodes that have measured a temperature higher than a pre-defined Low-Temperature-Threshold (LTT) and lower than a High-Temperature-Threshold (HTT) are assumed to be in the boundary of the fire. At each step, the consensus algorithm runs over these sensor nodes and the result is disseminated across the network. In our case, we consider the LTT = 30°C and HTT = 150°C. Also, we assume that if the temperature of a sensor node reaches 200°C, it will be destroyed.

The algorithm finds the circle that minimizes the sum of the squares of the distances $d_i^2 = (\|Z - X_i\| - r)^2$ where X_i is the position of i^{th} node out of the m nodes that participate in the consensus finding. In order to solve this nonlinear least square problem, the Gauss-Newton method is used. Indeed, the objective is to minimize $\sum_i d_i(\boldsymbol{u})^2$, where $\boldsymbol{u} = (z_1, z_2, r)$ is the vector of unknowns and (z_1, z_2) are the *x* and *y*-coordinates of the centre, respectively [8]. This nonlinear optimization problem is solved iteratively by a sequence of linear least square problems. We set $\hat{\boldsymbol{u}} = \tilde{\boldsymbol{u}} + \boldsymbol{h}$ as the next approximate solution if the current approximate solution is $\tilde{\boldsymbol{u}}$.

By expanding $\boldsymbol{d}(\boldsymbol{u}) = (\boldsymbol{d_1}(\boldsymbol{u}), \boldsymbol{d_2}(\boldsymbol{u}), \ldots, \boldsymbol{d_m}(\boldsymbol{u}))$ around $\tilde{\boldsymbol{u}}$ using Taylor series, one obtains:

$$\boldsymbol{J}(\tilde{\boldsymbol{u}}).\boldsymbol{h} = -\boldsymbol{d}(\tilde{\boldsymbol{u}}) \tag{1}$$

Solving (1), we find $\boldsymbol{h}$ in each iteration and update the approximation by $\hat{\boldsymbol{u}} = \tilde{\boldsymbol{u}} + \boldsymbol{h}$. Since $\boldsymbol{J}$ is not a square matrix, the vector $\boldsymbol{h}$ is determined by $\boldsymbol{h} = -\left(\boldsymbol{J}^T\boldsymbol{J}\right)^{-1}\boldsymbol{J}^T\boldsymbol{d}$. For our problem we have:

$$\boldsymbol{J}(\boldsymbol{u}) = \begin{pmatrix} \frac{z_1 - x_1}{\sqrt{(z_1 - x_1)^2 + (z_2 - y_1)^2}} & \frac{z_2 - y_1}{\sqrt{(z_1 - x_1)^2 + (z_2 - y_1)^2}} & -1 \\ \vdots & \vdots & \vdots \\ \frac{z_1 - x_m}{\sqrt{(z_1 - x_m)^2 + (z_2 - y_m)^2}} & \frac{z_2 - y_m}{\sqrt{(z_1 - x_m)^2 + (z_2 - y_m)^2}} & -1 \end{pmatrix} \tag{2}$$

which can be rewritten as

$$\boldsymbol{J}(\boldsymbol{u}) = \begin{pmatrix} \cos(\alpha_1) & \sin(\alpha_1) & -1 \\ \vdots & \vdots & \vdots \\ \cos(\alpha_m) & \sin(\alpha_m) & -1 \end{pmatrix} \tag{3}$$

where α_i is the angle between the horizontal axis and the line through X_i and Z. Then

$$\boldsymbol{J}^T\boldsymbol{J} = \begin{pmatrix} \sum_i \cos^2(\alpha_i) & \sum_i \cos(\alpha_i).\sin(\alpha_i) & -\sum_i \cos(\alpha_i) \\ \sum_i \cos(\alpha_i).\sin(\alpha_i) & \sum_i \sin^2(\alpha_i) & -\sum_i \sin(\alpha_i) \\ -\sum_i \cos(\alpha_i) & -\sum_i \sin(\alpha_i) & m \end{pmatrix} \tag{4}$$

Then we assume that the α of sensor nodes that are in the boundary of the fire are almost uniformly distributed in $[0, 2\pi]$ Hence, we have

$$\begin{aligned} &\sum_i \cos(\alpha_i) \approx 0 \\ &\sum_i \sin(\alpha_i) \approx 0 \\ &\sum_i \cos^2(\alpha_i) = \sum_i \frac{1}{2}\left(1 + \cos(2\alpha_i)\right) \approx \frac{1}{2}m \\ &\sum_i \sin^2(\alpha_i) = \sum_i \frac{1}{2}\left(1 - \cos(2\alpha_i)\right) \approx \frac{1}{2}m \ \sum_i \cos(\alpha_i) \approx 0 \\ &\sum_i \cos(\alpha_i).\sin(\alpha_i) = \sum_i \frac{1}{2}\sin(2\alpha_i) \approx 0 \end{aligned} \tag{5}$$

Therefore, $\boldsymbol{J}^T\boldsymbol{J}$ can be simplified to

$$\boldsymbol{J}^T\boldsymbol{J} = \begin{pmatrix} \frac{1}{2}m & 0 & 0 \\ 0 & \frac{1}{2}m & 0 \\ 0 & 0 & m \end{pmatrix} \tag{6}$$

and as a result,

$$\left(\boldsymbol{J}^T\boldsymbol{J}\right)^{-1} = \frac{1}{m}\begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{7}$$

At last, we find

$$\boldsymbol{h} = \begin{pmatrix} \frac{1}{m}\sum_{i=1}^{m} 2\cos(\alpha_i)\cdot\left(r-\sqrt{(z_1-x_i)^2+(z_2-y_i)^2}\right) \\ \frac{1}{m}\sum_{i=1}^{m} 2\sin(\alpha_i)\cdot\left(r-\sqrt{(z_1-x_i)^2+(z_2-y_i)^2}\right) \\ \frac{1}{m}\sum_{i=1}^{m}\left(\sqrt{(z_1-x_i)^2+(z_2-y_i)^2}-r\right) \end{pmatrix} \tag{8}$$

As can be seen, the components of the vector $\boldsymbol{h}$ are determined by averaging of three quantities across the sensors that are in the boundary of the fire. In fact, sensor *i* computes locally the *i*-th term of each sum and then finds the average value by running a simple averaging consensus algorithm as follows. At step t each sensor node sends the calculated data $\boldsymbol{h}(t)$ to its neighboring nodes. Then it uses the following formula to update $\boldsymbol{h}(t)$

$$\boldsymbol{h}(t+1) = (\boldsymbol{I}-\varepsilon\boldsymbol{L})\boldsymbol{h}(t) \tag{9}$$

In this formula, $\boldsymbol{I}$ and $\boldsymbol{L}$ are the identity and Laplacian matrices, respectively. Also, $\varepsilon > 0$ is the connection weight. It is shown in [6] that if $0 < \varepsilon < 1/\Delta$ then $\boldsymbol{h}(t)$ converges to the vector $\boldsymbol{h}$ of (8). Δ is the maximum degree of the sensor nodes. Alternatively, different weights for each link could be chosen [9].

It should be mentioned that the initial values of (z_1, z_2) constitute the Fire-Start position that as found in the Fire-Detection phase and the initial value of r is set to a default value. We have chosen it 5m in our simulations. The corresponding initial values of the entries of the vector $\boldsymbol{h}$ are determined by using these values.

After reaching consensus, each sensor node in the boundary has vector $\boldsymbol{h}$ and is able to update vector $\boldsymbol{u}$ and as a result, it finds new parameters of the circle surrounding the fire. Also, these nodes send a Fire-Localization message carrying the new parameters of the circle to the sensor nodes that are not in the boundary of the fire.

In the fire localization phase, the sensor nodes that are not in the boundary accept the average of the parameters (center and radius of the circle) in the received Fire-Localization messages as the new parameters of the circle surrounding the fire. Also, they send the results in a new Fire-Localization message to their neighboring sensor nodes. After a while, all sensor nodes have the desired parameters. Every time that the sensor node samples the temperature of the environment, it decides whether to be involved in a new consensus process and the whole process is repeated. If a sensor node receives some Fire-Localization messages and then enters the consensus process, the initial values of the entries of vector $\boldsymbol{u}$ are the parameters that are updated by Fire-Localization messages, instead of those that were found in the Fire-Detection phase. The whole process is repeated periodically with a pre-specified period.

4 Simulation Results

The algorithms are simulated by Castalia which is specially designed for wireless sensor networks. For our simulation, we have considered 1600 sensor nodes that are deployed in a 200m*200m area. The area is divided into 5m*5m sub-areas and in each sub-area a sensor node is deployed in a random position. The transmission power level is such that the transmitted message can be reached only by neighboring nodes. The duty cycle of the listening interval for each node is 0.1 and the temperature sampling interval is 8 seconds. Each sensor node checks the channel before transmission and waits until the channel is free. The wireless channel model is realistic and the effect of interference is considered to be additive. The Path-Loss-Exponent is 2.4 and the channel quality and the effect of fading between a pair of nodes in one direction are not correlated with the opposite direction.

The data of the sensor nodes that were used by Castalia as the data input was generated by the FARSITE fire simulation software which is knows to produce realistic forest fire scenarios. It is worth to mention that the current version of Castalia does not accept any trace file as the input data for sensor nodes. Hence, in order to feed the data of FARSITE to Castalia, we added the possibility of using external trace files to Castalia successfully and thus enabled data exchange from FARSITE to Castalia

Simulation results show that the entire sensor nodes go to the Fire-Alarm state after the fire started with a very short delay below 4 seconds. Figure 2 shows the number of nodes that are in Fire-Alarm mode versus the time difference from the first time of detection of fire in the network. In addition, the states of the nodes in four time instances (a-d) which are marked in figure 2 are depicted in figure 3.

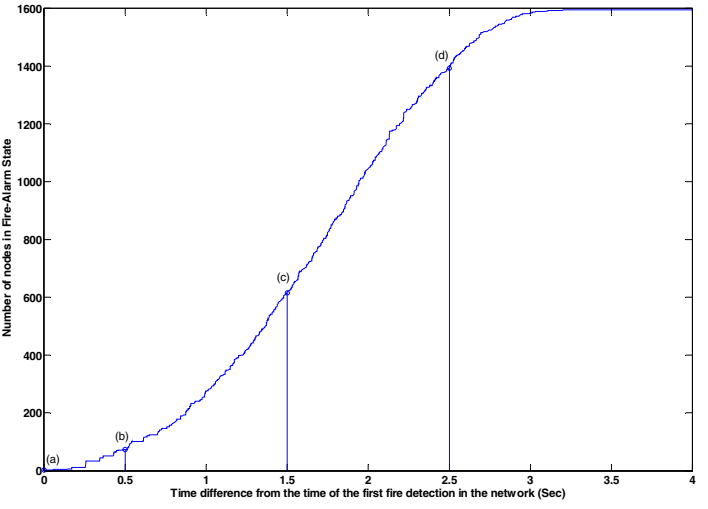

Fig. 2. Number of nodes in Fire-Alarm mode versus the time difference from the first time of fire detection in the network

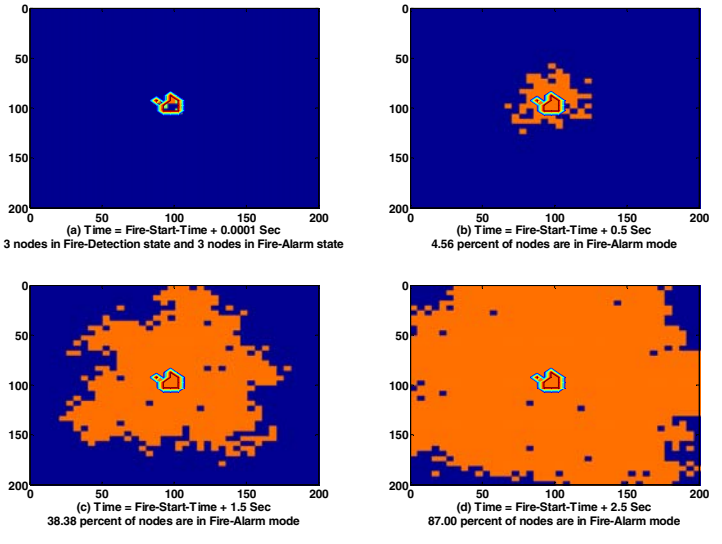

Fig. 3. The states of the nodes in for 4 time instances. BLUE=No-Fire, ORANGE=Fire-Alarm, GREEN=Fire-Detection. In black and white format, BLACK=No-Fire, GRAY=Fire-Alarm, WHITE=Fire-Detection. The contour plot of the fire is shown in the middle of the figures.

As figure 3 shows the spreading of the alarm is quite fast and the Alarm is disseminated in at last 4 seconds which is an acceptable delay. Figure 4 shows the histogram of the difference between the derived geometrical locations of the Fire-Start with real location.

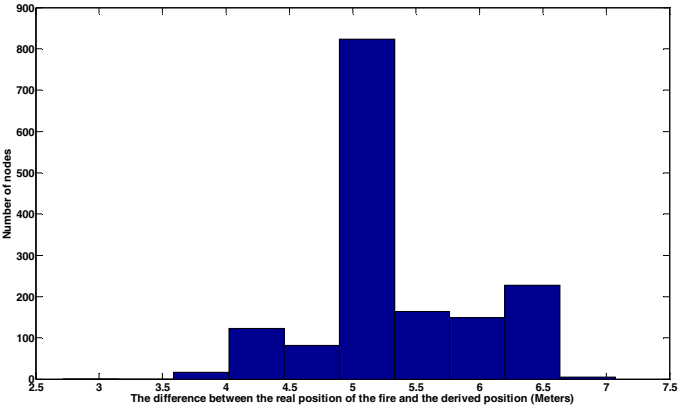

Fig. 4. Histogram of the difference between the real position and estimated positions of the Fire-Start point

As can be seen there is a bias in this difference that is due to the difference between the real position of the Fire-Start point and the position of the first nodes that have detected the fire. Although this bias is inevitable, it does not cause a considerable error in results.

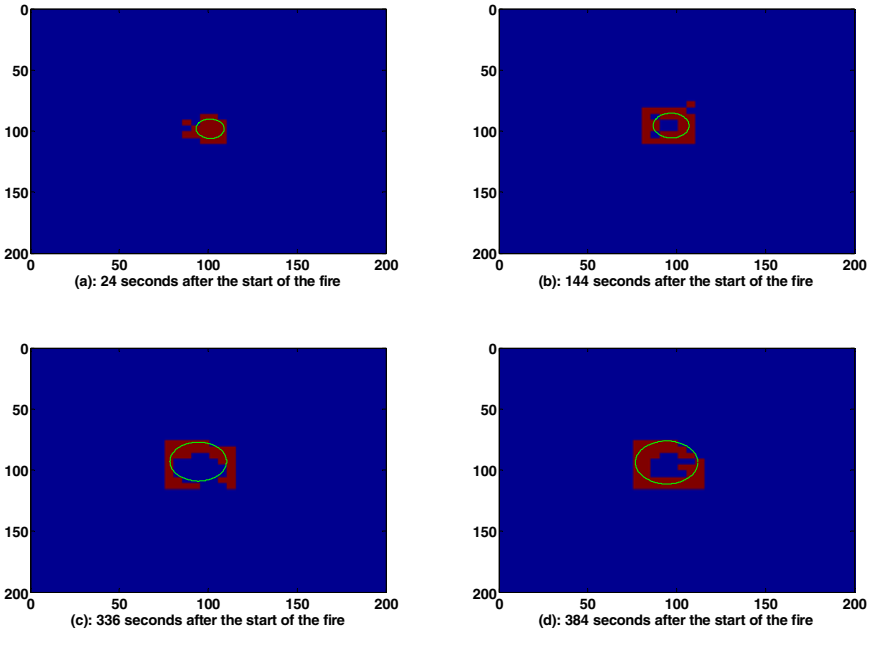

Fig. 5. Four snapshots of the network during Fire-Localization phase. The red (gray in black and white mode) region shows the location of the sensor nodes that are in the boundary of the fire.

The Fire Localization algorithm starts running right after Fire Detection phase. Simulation results show that each node in the network obtains the parameters of the circle after each consensus process. We have taken 80 iterations for each consensus process. With a better choice of the connection weights, the number of iterations could still be reduced. In order that all the sensor nodes have enough time to process the received data, the time considered for each iteration of the consensus process is 0.3 seconds. Thus, the parameters of the circle are updated every 24 seconds. Also, we have chosen $\varepsilon = 0.02$. Choosing a large value for ε leads to the instability of the consensus process.

In figure 5, there are four snapshots of the network. The red units show the places that are in the boundary of the fire. The green circle is the circle that is found by the Fire-Localization algorithm.

This algorithm is also tested in ideal conditions where there is no interference in the communication channel and the buffer size of the radio receiver of the sensor nodes is infinite. The results show that the performance of our algorithm in real conditions is close to ideal conditions. Figure 6 shows the difference in the radius of the circle obtained in real and in ideal conditions in 15 intervals.

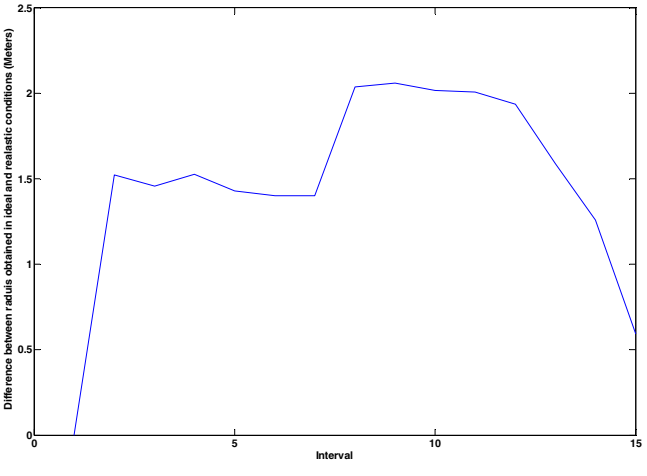

Fig. 6. Difference in the radius of the circle obtained in the real and ideal conditions

This error is negligible with respect to the size of the network. In summary, our algorithms work with a good accuracy and acceptable delay.

5 Conclusion

We introduced two algorithms for WSNs for the application of forest fire detection and localization with good performance in terms of rapid spreading of alarm in the case fire outbreak as well as a reasonably accurate circle surrounding the fire. Simulation results show that our algorithms react rapidly to the changes in the network with

high reliability. They also show that realistic wireless channel conditions do not deteriorate much the speed and accuracy of the algorithms.

Acknowledgments. This work has been funded by the WINSOC Project; a Specific Targeted Research Project (contract number 0033914) funded by the INFSO DG of the European Commission within the RTD activities of the Thematic Priority Information Society Technologies. The authors would also like to express their thanks to INTRACOM Co. for providing us the FARSITE output data.

References

1. Chandrakasan, A., Amirtharajah, R., Seonghwan, C., Goodman, J., Konduri, G., Kulik, J., Rabiner, W., Wang, A.: Design considerations for distributed microsensor systems. Custom Integrated Circuits. Proceedings of the IEEE, 279–286 (1999)
2. Clare, L.P., Pottie, G.J., Agre, J.R.: Self-organizing distributed sensor networks. In: Unattended Ground Sensor Technologies and Applications, vol. 3713, pp. 229–237. SPIE, Orlando (1999)
3. Dong, M.J., Yung, K.G., Kaiser, W.J.: Low power signal processing architectures for network microsensors. In: International Symposium on Low Power Electronics and Design, Proceedings, pp. 173–177 (1997)
4. Castalia, `http://castalia.npc.nicta.com.au`
5. FARSITE, `http://www.firemodels.org`
6. Olfati-Saber, R., Olfati-Saber, R., Fax, J.A., Murray, R.M.: Consensus and Cooperation in Networked Multi-Agent Systems. Proceedings of the IEEE 95, 215–233 (2007)
7. Pescosolido, L., Barbarossa, S., Scutari, G.: Decentralized Detection and Localization Through Sensor Networks Designed As a Population of Self-Synchronizing Oscillators. In: IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2006, Proceedings, vol. 4, pp. IV981–IV984 (2006)
8. Gander, W., Golub, G.H., Strebel, R.: Least-squares fitting of circles and ellipses. BIT Numerical Mathematics 34, 558–578 (1994)
9. Xiao, L., Boyd, S., Kim, S.-J.: Distributed average consensus with least-mean-square deviation. Journal of Parallel and Distributed Computing 67, 33–46 (2007)