

Indoor Fire Detection using Wireless Sensor Networks

Le Dinh Thanh, Nguyen Dai Tho

VNU University of Engineering and Technology, 144 Xuan Thuy, Cau Giay, Hanoi, Vietnam

Emails: {thanhld, nguyendaitho}@vnu.edu.vn

Abstract: *In this paper, we propose an algorithm for early detection of indoor fire based on temperature and light data collected by wireless sensor networks. Our goal is to develop an affordable, acceptable, detection method based on wireless sensor networks with cheap and highly available sensors. Experimental results show that our proposed algorithm is able to detect fire quickly and accurately.*

Keywords: Wireless sensor network, fire detection, dynamic threshold, light fluctuation.

I. INTRODUCTION

There are many effective fire detection methods that have been applied to indoor environments [4, 6, 7, 8]. However, since these methods use expensive devices, they are not cost-effective. Researchers, therefore, are still in searching for affordable, yet acceptable, methods. One of such methods is to use wireless sensor network (WSN) with cheap and highly available thermal and light sensors [13]. The goal is to make a compromise between efficiency and cost. Light detection in [13], however, is only applicable to outdoor environments.

With the same goal of achieving efficiency with low cost, in this paper, we propose an algorithm, namely Dynamic Threshold and Light Fluctuation (DTLF), that effectively detects indoor fire using the same assumption that WSN has only thermal and light sensors. Two key points that make our algorithm different are (1) the automatic adjustment of the threshold by the thermal detector and (2) the usage of light fluctuation as a phenomenon of fire by the light detector. Experimental results show that our proposed algorithm is able to detect fire quickly and accurately.

Related works are reviewed in Section 2. Our proposed algorithm for indoor fire detection is

described in Section 3. Then, experimental results are presented in Section 4. Finally, Section 5 provides our conclusion and future works.

II. RELATED WORKS

Wireless sensor networks have been extensively used in environmental monitoring [5, 10, 12]. Among hundreds of environmental events that should be monitored, fire has been catching much attention. Several WSN-based systems have been proposed for both indoor [2, 14] and outdoor [1, 11] fire monitoring. Narendra et al [14] have designed a fire monitoring and extinguishing system and deployed this system in a hospital building. Sensed data is transmitted from monitoring nodes to extinguishing nodes. If an extinguishing node realizes, based on the received data, that there is an enormous increase in the temperature, it will activate the process of fire extinguishment. Authors in [2] presented a Zigbee based WSN framework for facilitating the monitoring of indoor environments.

Works closely related to ours include State Model [13], Double Sliding Windows [9] and Monitoring Window [3]. State Model [13] detects abnormal conditions based on the ratio between the current temperature and the window average (the average of the temperature measurements in the window). Light intensity is used by this method to detect the presence of sunbeams. State Model is designed for outdoor environments. Double Sliding Window [9], on the other hand, makes its decisions based on the ratio between the averages of two consecutive windows. If it is larger than a given threshold, fire is assumed to occur. Monitoring Windows [3] also operates based on the calculation of window averages. Two thresholds are used during this operation. The current window will be moved forward if its average is less than a lower threshold. Otherwise, an anomaly is detected and a consecutive

window is opened. The anomaly is said to be a global (resp. local) one if the window average is greater than (resp. less than or equal to) an upper threshold. Fire is assumed to be present if two or more global anomalies or so many (at least half of the window size) local anomalies are detected in two consecutive windows.

Our proposed thermal detector exploits neither an enormous increase in the temperature [14] nor some values of recent windows [3, 9, 13]. Instead, it makes use of the accumulative average, namely $mean_t$, or the value of the global window that contains all ever sampled normal thermal measurements. Intuitively, $mean_t$ is the

environmental temperature, and the occurrence of fire makes the sampled thermal measurement, t , significantly higher than $mean_t$.

On the other hand, our proposed light detector does not only rely on the presence of light like the State Model method [13] but also takes into account the fluctuating nature of light emitted by fire. As far as we know, our proposed method is the first one that exploits the fluctuation of light as a phenomenon of fire.

III. PROPOSED ALGORITHM

Temperature and light measurements in different conditions are collected then analysed. We have

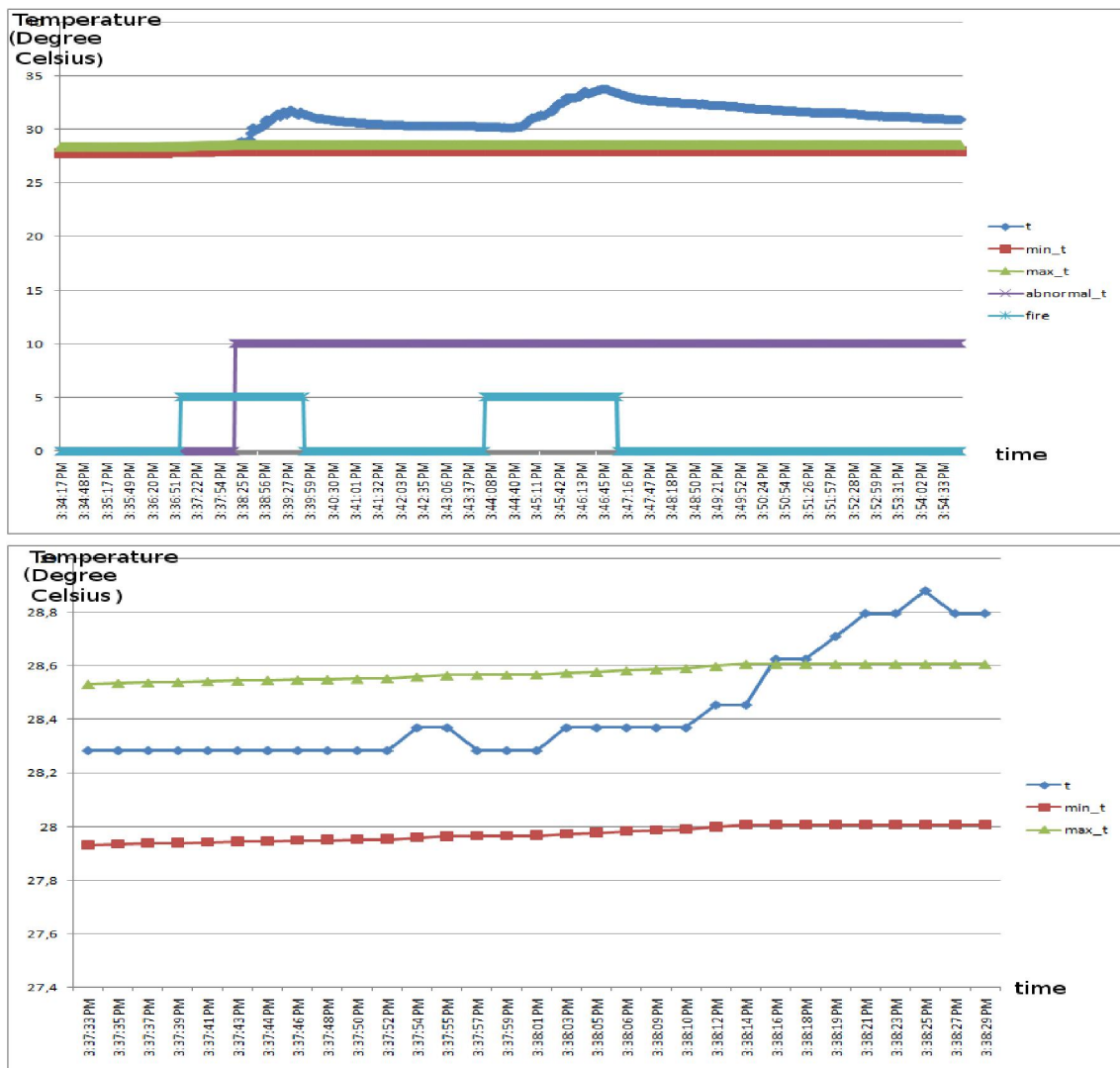


Fig. 1. The adaptation of thermal thresholds to the change of temperature, together with the output of thermal detection: (upper) a data series, (lower) a room in to that data series.

artificially created controllable fire in such conditions. After figuring out how temperature and light change under certain conditions, we propose an algorithm, namely Dynamic Thresholds and Light Fluctuation (DTLF), for indoor fire detection.

III.1. Thermal Detection

The environmental temperature, or the indoor temperature in normal conditions without fire, is monitored and used as the base for fire detection. It changes gradually from time to time. Fire is detected if the sensed temperature becomes significantly higher than the environmental temperature.

Let $mean_t$ be the environmental temperature which is monitored. Then, the upper threshold is defined as $max_t = mean_t + ampl_t$, where $ampl_t$ is a constant. The lower threshold min_t is set to $mean_t - ampl_t$. Basically, there will be a threat of fire if the temperature exceeds max_t .

Our key idea is to adjust $mean_t$ according to environmental conditions. For example, $mean_t$ should be automatically changed from season to season, from day to night, and even from raining to sunny weather. It may be different from one node to another since nodes are placed in different locations (rooms, halls, ...) with different weather conditions. At the beginning, $mean_t$ is set to the first temperature measurement. $mean_t$ is updated whenever the sensor node has a new temperature measurement, namely t . If $min_t \leq t \leq max_t$, i.e. the temperature is normal, then the sensor node will update $mean_t$ as follows: $mean_t = \alpha * t + (1-\alpha) * mean_t$, where $0 \leq \alpha \leq 1$. $mean_t$ is unchanged if $t < min_t$ or $t > max_t$, i.e. the temperature is abnormal.

Thermal indicator, $abnormal_t$, is defined as a variable whose value is TRUE if $t > max_t$, i.e. a fire is detected, or FALSE otherwise. The automatic adjustment of max_t is a simple, yet powerful mechanism. Consequently, our thermal detection can adapt itself according to the changes in weather.

Figure 1 shows a series of temperature measurements, in a condition with artificially created fire, collected by a sensor node, and the according changes of max_t , min_t , and $abnormal_t$. Note, in this figure, that the $abnormal_t$ and fire

indicators do not make sense to the metric (Degree Celcius) of the y-axis. The non-zero (TRUE) value of $abnormal_t$ is already explained above. The non-zero value of fire simply indicates that artificial fire is present. The values of $ampl_t$ and α must be chosen carefully so that fire can be detected as soon as possible and the thermal detector does not make false-positive detection. Choosing the values of $ampl_t$ and α will be described in more detail in Section 4.

III.2. Light Detection

Light fluctuates frequently when fire is present. The fluctuation of light is, then, a phenomenon that indicates, with some probability, the presence of fire. By using the word ‘‘probability’’ we would like to note that few other sources, such as televisions, can emit fluctuating light as well. It is worth noting that light changes gradually in normal condition. Therefore, in order to make our light detection workable, we recommend not to expose sensors directly to such lighting sources.

To detect the fluctuation of light, each sensor node analyses the light measurements in its window. Let k be the window size. Let $l_{c-k-1}, \dots, l_{c-1}, l_c$ be the light measurements in the window, i.e. the k most recent light measurements, where l_c is the last one. $l_{c-i}, i = 1 \dots k - 2$, is called a *local minimum* if $l_{c-i} < l_{c-i+1}$ and $l_{c-i} < l_{c-i-1}$. Conversely, $l_{c-i}, i = 1 \dots k - 2$, is called a *local maximum* if $l_{c-i} > l_{c-i+1}$ and $l_{c-i} > l_{c-i-1}$. A series $[l_{i-j}, \dots, l_{i-1}, l_i]$ is called a *left slope* if l_{i-j} is a local minimum, l_i is a local maximum and $l_{i-j+1}, \dots, l_{i-1}$ are neither local minima nor local maxima. Conversely, a series $[l_{i-j}, \dots, l_{i-1}, l_i]$ is called a *right slope* if l_{i-j} is a local maximum, l_i is a local minimum and $l_{i-j+1}, \dots, l_{i-1}$ are neither local minima nor local maxima. A left/right slope $[l_{i-j}, \dots, l_{i-1}, l_i]$ is called *significant* if $|l_{i-j} - l_i| > thres_{lc}$, where $thres_{lc}$ is a given threshold. The fluctuation of light, then, can be detected if the number of significant left slopes (inc_l) and the number of significant right slopes ($desc_l$) are both greater than another given threshold $thres_{sc}$.

Light indicator, $abnormal_l$, is defined as a variable whose value is TRUE if $inc_l > thres_{sc}$ and $desc_l > thres_{sc}$, i.e. light fluctuation is detected, or FALSE otherwise.

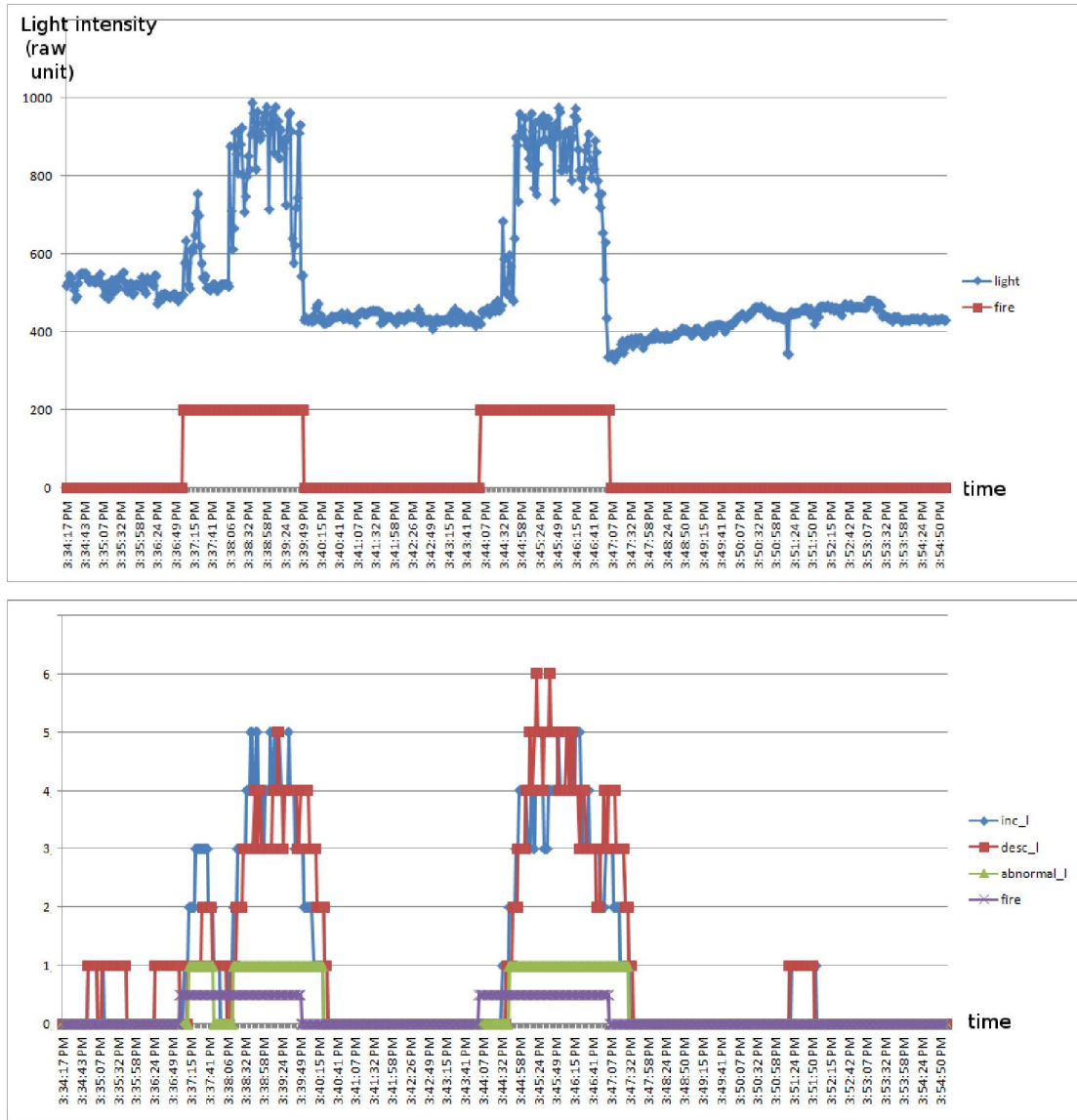


Fig. 2. The fluctuation of light caused by fire (upper) and the output of light detection (lower).

Figure 2 shows a series of light measurements, in a condition with artificially created fire, collected by a sensor node, and the according changes of inc_l , $desc_l$, and $abnormal_l$. Note, in this figure, that the $abnormal_l$ and fire indicators do not make sense to the metric of the y-axis. The non-zero (TRUE) value of $abnormal_l$ is already explained above. The non-zero value of fire simply indicates that artificial fire is present. Note also that light measurements are given in raw unit, i.e. the output of the nominal circuit in the nodes. There is no need to convert these measurements to engineering units.

The values of k , $thres_{lc}$ and $thres_{sc}$ must be chosen carefully so that fire can be detected as soon as possible and the light detector does not make false-positive detection. Choosing the values of k , $thres_{lc}$ and $thres_{sc}$ will be described in detail in Section 4.

III.3. Levels of Risk

Whenever a sensor node has got a new data sample, it calls the thermal detection function and the light detection function described above, then combines the outputs of these two functions to

decide a level of fire risk. Table 1 explains this combination.

Table 1. The levels of fire risk

<i>Thermal indicator</i>	<i>Light indicator</i>	<i>Level of risk</i>	<i>Meaning</i>
FALSE	FALSE	0	No threat of fire
FALSE	TRUE	1	Low threat of fire
TRUE	FALSE	2	High threat of fire
TRUE	TRUE	3	Fire detected

IV. EXPERIMENTS

Our proposed algorithm DTLF is experimentally compared to State Model [13], Double Sliding Window [9], and Monitoring Windows [3]. The experiments are conducted in two separate phases: training phase and working phase¹. The task of the training phase is to set the right values to parameters used by each algorithm. The working phase, on the other hand, provides comparative evidences of the effectiveness of compared algorithms.

IV.1. Training Phase

The objective of this phase is to set the parameters of each compared algorithm to proper values so that the algorithm produces the fewest false-negative (failing to detect fire occurrences) and the fewest false-positive (incorrectly identifying normal condition as fire occurrences). The false-negative index, f_n , is the percentage of occurrences of fire that the algorithm does not detect, i.e. $f_n = o_m/o_a \times 100\%$ where o_a is the total number of occurrences of fire, and o_m is the number of occurrences of fire that the algorithm does not detect. The false-positive index, f_p , is the percentage of wrong detections, i.e. $f_p = d_w/d_a \times 100\%$ where d_a is the total number of times that the algorithm assumes that fire is happened, and d_w is the number of times that the algorithm makes wrong assumptions. The objective of the training phase is then restated as to set the values to parameters of each compared algorithm so that the algorithm produces minimal $F = f_p + f_n$.

¹ Data and source codes are available at <http://bit.ly/22vPnFf>

In order to achieve the above objective, the following 3-step process is conducted during the training phase.

- *Step 1:* Gathering training data: We use several sensor nodes to get a number of series of temperature and light measurements in different environment conditions. A half of such series are taken in conditions with artificially created fire. Each series is then tagged with the occurrences of fire. An occurrence of fire is marked by the beginning and ending points of time.
- *Step 2:* Running each detection algorithm with varying values of its parameters: For each detection algorithm, we develop a program that uses loops to vary the values of the parameters. Then, for each combination of these values, this program simulates the behaviour of sensor nodes, i.e. sequentially reads the measurements in each data series and runs the detection algorithm. The outputs of these runs are written to files for use in Step 3.
- *Step 3:* Setting parameters to selected values: We develop another program that takes the outputs in Step 2, together with the records of the occurrences of fire, as its input. For each output in Step 2, the program computes $F = f_p + f_n$. Then, for each detection algorithm, the program chooses the output in Step 2 and the associated combination of values of parameters that produce minimal F .

After our training phase, each compared algorithm passes fixed values to its parameters. These values are listed as follows:

- DTLF: $ampl_t = 0.1^\circ\text{C}$, $\alpha = 0.65$, $k = 15$, $thres_c = 50$, $thres_{sc} = 6$.
- Sate Model: $window_size = 16$, $ratio_threshold = 1.01$.
- Double Sliding Window: $window_size = 8$, $ratio_threshold = 1.01$.
- Monitoring Window: $window_size = 10$, $lower_threshold = 8071.4$, $upper_threshold = 8081.4$.

IV.2. Working Phase

In order to have the best evaluation of the compared algorithms, we have implemented these

algorithms in real motes and deployed a nearly real production system. The whole system (see Figure 3) is divided into three layers: mote layer, server layer and client layer. The mote layer is the wireless sensor network whose functionality is to sense environmental data (temperature and light) and to detect fire. It is connected to the server layer via a gateway. The server layer is responsible for the storage of sensed data and providing services to

In our implementation, we use the MoteWorks platform provided by Crossbow Technology². In more detail, we use MICAz motes with MDA100 sensor board attached. Each MDA100 sensor board has a temperature sensor and a light sensor by default. TinyOS³ is the operating system for the motes. xMesh routing protocol turns the set of sensors into a multihop wireless network. Compared algorithms are, in turn, implemented on the motes

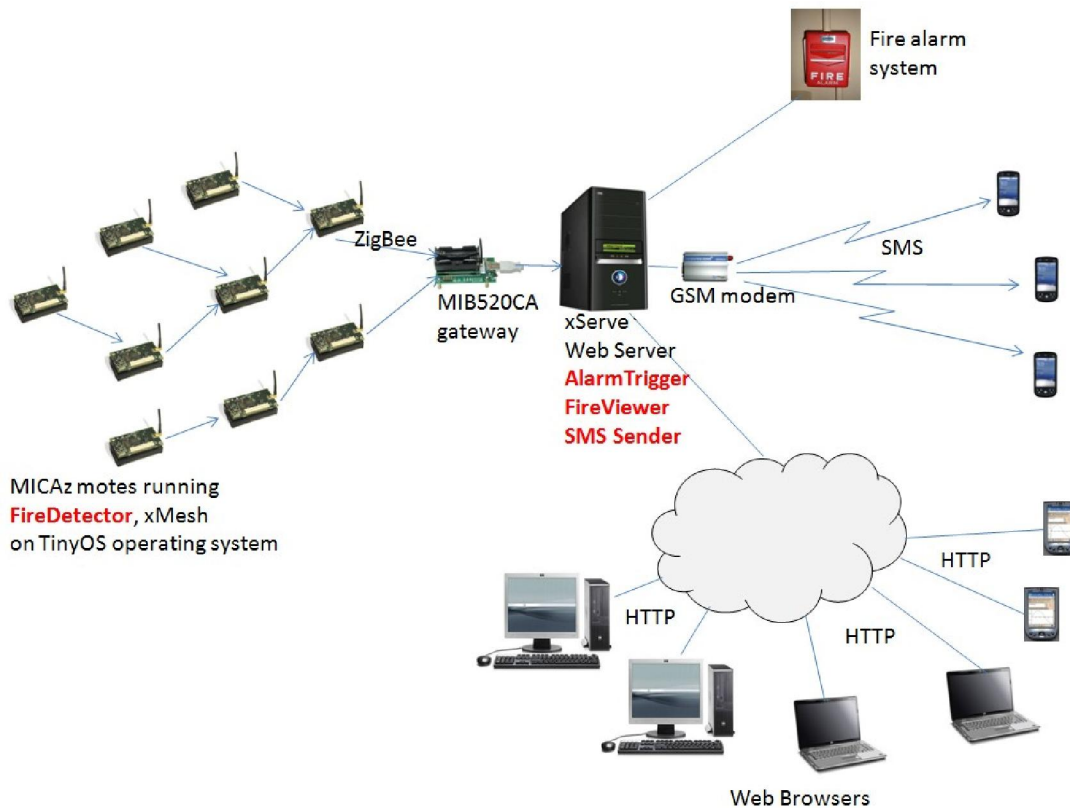


Fig. 3. The architecture of indoor fire monitoring system.

clients in the last layer. Services provided by the system include environment (temperature and light) monitoring and fire alerting. Specially, there are three services offered to the client layer. The first service is to view the sensed data in a remote web browser. The second service is to send SMS messages to responsible people in order to inform them of the fire risk. The last service is to trigger a fire alarm system if fire is detected. The client layer consists of web browsers, cellphones, and fire alarm systems.

using the NesC programming language⁴. Sensed data is sent from motes to the gateway by xMesh. Then, xServe receives the sensed data from the gateway and stores data in a PostgreSQL database. We have developed FireViewer which is a web application for remotely viewing the sensed data while the components for sending SMS messages to

² <http://xbow.com>

³ <http://www.tinyos.net/>

⁴ <http://nesc.sourceforge.net>

mobile phones of the people in charge and for triggering an alarm system are left for future work.

Since energy is consumed mostly by message transmitting, in order to save energy, we let motes send as few messages as possible. A mote needs only to send message toward the server in one of the following cases:

- It has detected that there is a risk of fire (risk level > 0).
- A time period has ended up so the mote needs to report to the server that it is still alive.

Each mote gets temperature and light measurements in every time period called *sensing cycle*. Then, it analyses the sensed measurements in order to decide if there is a risk of fire. If there is, the mote will send an ALARM message toward the server immediately. Otherwise, if a *reporting cycle* ends up, the mote will send an ALIVE message toward the server in order to tell the server that the mote is still working. Both ALARM and ALIVE messages convey sensed data to the server. An ALARM message lets the server know that there is a risk of fire near the source of the message.

Our experimental system was launched during May 2014. Different weather conditions (day, night, sunny, rainy, hot, cool) were experienced by our system. We artificially created many controllable fire incidences. All ALARM messages were recorded for statistics. In addition to false-positive and false-negative indices, average time delay is also reported. Time delay is measured in seconds as the period from the beginning of fire incidence to the point of time by which the corresponding ALARM message has been received at the server. The results are presented in Table 2.

Among the four competitors, our algorithm produces the fewest false-negatives as well as the fewest false-positives. State Model, on the other hand, produces the most false-negatives as well as the most false-positives. Double Sliding Windows has the smallest average time delay.

Experimental results show that the use of windows is effective. State Model depends heavily on the latest sampled data. Because of the technical issues, the sampled data may be incorrect, or error-

prone. Therefore, State Model makes so many false decisions. In order to overcome this, Double Sliding Windows and Monitoring Windows use several local windows, DTLF uses the global window. In other words, the use of windows tolerates the errors in sensed data.

Table 2. Algorithm comparison via experiments

<i>Metric</i>	<i>DTLF</i>	<i>State Model</i>	<i>Monitoring Window</i>	<i>Double Sliding Windows</i>
False-negative f_n (%)	5.88	47.05	35.29	23.53
False-positive f_p (%)	5.88	10.00	8.33	7.14
Average time delay (s)	46	56	44	43

Experimental results also show that the average time delay is not proportional to the percentage of correct decisions. This is true because if fire is hard to be detected, due to the distance or obstacles for example, then it can only be detectable after a long time.

V. CONCLUSION

In this paper, we proposed a new fire detection algorithm based solely on temperature and light data collected by the wireless sensor network. The proposed algorithm is more effective than existing algorithms. In the future, we are going to build a complete system with a large number of sensor nodes.

ACKNOWLEDGMENTS

This work was supported by the project CN.13.07 funded by VNU University of Engineering and Technology.

REFERENCES

- [1] Y. E. Aslan, I. Korpeoglu, and O. Ulusoy. A framework for use of wireless sensor networks in forest fire detection and monitoring. *Computers, Environment and Urban Systems*, 36:614–625, 2012.
- [2] S. S. Bhunia, S. Roy, and N. Mukherjee. Iems: Indoor environment monitoring system using zigbee

- wireless sensor network. In *Proceedings of the ICCCS*, pp. 142–145, 2011.
- [3] G. Boggia, P. Camarda, L. A. Grieco, and M. R. Palattella. Fire detection using wireless sensor networks: an approach based on statistical data modeling. In *Proceedings of the New Technologies, Mobility and Security Conference*, 2008.
- [4] A. E. Cetin, K. Dimitropoulos, B. Gouverneur, N. Grammalidis, O. Gu`nay, Y. H. Habiboglu, B. U. Toreyin, and S. Verstockt. Video fire detection review. *Digit. Signal Process.*, 23(6):1827–1843, Dec. 2013.
- [5] G. H. Choi, G. S. Choi, and J. H. Jang. A framework for wireless sensor network in web-based monitoring and control of indoor air quality (iaq) in subway stations. In *Proceedings of the 2nd IEEE International Conference on Computer Science and Information Technology*, pp. 378–382, August 2009.
- [6] A. E. Cote. Fire Protection Handbook eighteenth edition. National Fire Protection Association, 2000.
- [7] C. J. Fleming. Smoke detector technology research. The World Fire Safety Foundation.
- [8] L. A. Gundel, G. A. Michael, and R. N. Albert. Carbon monoxide detector technology comparison: Response to various gases. Technical Report LBNL-40556, Ernest Orlando Lawrence Berkeley National Laboratory, 1998.
- [9] K. Kapitanova, S. H. Son, and K. D. Kang. Using fuzzy logic for robust event detection in wireless sensor networks. *Ad Hoc Netw.*, 10(4):709–722, 2012.
- [10] A. M. C. Lee, C. T. Angeles, M. C. R. Talampas, L. G. Sison, and M. N. Soriano. Motesart: Wireless sensor network for monitoring relative humidity and temperature in an art gallery. In *Proceedings of the IEEE International Conference on Networking, Sensing and Control*, pp. 1263–1268, April 2008.
- [11] Y. Liu, Y. Gu, G. Chen, Y. Ji and J. Li. A novel accurate forest fire detection system using wireless sensor networks. In *Proceedings of the 7th International Conference on Mobile Ad-hoc and Sensor Networks*, pp. 52–59. 2011.
- [12] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson. Wireless sensor networks for habitat monitoring. In *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, pp. 88–97, 2002.
- [13] O. S. Penha and E. F. Nakamura. Fusing light and temperature data for fire detection. In *Proceedings of the IEEE Symposium on Computers and Communications*, pp. 107–112, 2010.
- [14] P. N. N. Reddy, P. I. Basarkod, and S. S. Manvi. Wireless sensor network based fire monitoring and extinguishing system in real time environment. *Int. J. Advanced Networking and Applications*, 3(2):1070–1075, 2011.

AUTHORS' BIOGRAPHIES



Le Dinh Thanh received his Ph.D. degree in information technology from the University of Engineering and Technology, Vietnam National University, Hanoi (VNU-UET) in 2014. His research interests include wireless sensor networks, distributed computing, and information security. He worked at the Hong Duc University from 2005 to 2008. He is currently with VNU-UET.



Nguyen Dai Tho received his Ph.D. degree in computer science and engineering from the University of Technology Compiegne (UTC), France. He is currently with the University of Engineering and Technology, Vietnam National University, Hanoi. His research interests include network security, peer-to-peer systems, wireless networks, and distributed computing.