FED: FUZZY EVENT DETECTION MODEL FOR WIRELESS SENSOR NETWORKS

HadiTabatabaee Malazi[1],Kamran Zamanifar[1] and Stefan Dulman[2]

[1]Department of Computer Eng., University of Isfahan, Iran
{tabatabaee, zamanifar}@eng.ui.ac.ir
[2]Embedded Software Group, Delft University of Technology, The Netherlands
s.o.dulman@tudelft.nl

ABSTRACT

Event detection is one of the required services in sensor network applications such as environmental monitoring and object tracking. Composite event detection faces several challenges.The first challenge is uncertainty caused by variety of factors, while the second one is heterogeneity of sensor nodes in sensing capabilities. Finally, distributed detection,which is vital to facilitate uncovering composite events in large scale sensor networks, is challenging.We devised a new fuzzy event detection model which is called FED that benefits from fuzzy variables to measure the intensity as well as the occurrenceof detected events. FED uses fuzzy rules to define composite eventsto enhance handling uncertainty. Moreover, FED provides a node level knowledge abstraction, which offers flexibility in applying heterogeneous sensors. The model is also applicable to a clustered network for distributed event detection. The simulation results show that FED is less sensitive to environmental noise and performs better in terms of percentage of detected eventscompare to a similar approach.

KEYWORDS

Wireless sensor networks, Composite event detection, Fuzzy event detection (FED), Heterogeneity, Uncertainty

1. INTRODUCTION

Event detection is a popular service in environmental monitoring [1]–[3] and object tracking [4] applications.Ambulatory medical monitoring [5], vehicle tracking [6], [7] and military surveillance [8] are some sample applicationsthat event detection plays a key role. The popularity of this service is not limited to the applicationlayer. Several wireless senor network middlewares [9]–[14] provide the required primitives, such as event notification to facilitateevent detection tasks in various applications. “Event detection is a way to dig meaningful information out of thehuge volume of data produced” mentioned S.Li in [15].

Events are generally categorized into *simple* (*atomic*) and *composite* (*complex*) ones. Simple events can be detectedby an individual sensor type, whenever the sensed value is above/below the predefined threshold, while compositeevents (CE) are those that cannot be detected by a single sensor type and require collaboration among various types.

Composite event detection poses several challenges. One of the issues is the effect of uncertainty in the detectionprocess. Environmental noise, message collision, and hardware malfunctioning are some of the factors that maycause uncertainty. Uncertainty sources not only affect the detection of simple events at node level, but they alsoaffect CE detection in fusion points causing both false negatives and false positives. The node density also introducessome challenges. A low node density increases the chance that none of the notifications reach the

fusion point,while a higher density introduces a collision problem when nodes attempt to transmit simultaneously.

Event detection applications may use variety of different sensor types to uncover composite events using heterogeneoussensor nodes. The main reasons for applying heterogeneous sensor nodes are hardware constraints and energyconsiderations. Therefore, each node may not be able to detect a composite event based on its local observations.Consequently heterogeneous nodes are required to collaboratively detect composite evens. For example they maysubmit the detected simple events to an aggregation point to detect composite events [16].

The growing trend toward cyber-physical systems [17]–[19] introduces new desired properties such as *knowledgeabstraction* and *in network processing*. Event notification part of the traditional event detection systems does notcompletely fulfil these properties. The basic form of an event notification is a tuple consist of *event name*, *reportingnode ID* and *detection timestamp* which only reports the occurrence of an event in the binary form of *true* / *false*.To provide more information on the detected event, the sensed value field can be added to the notification tuple. Theproblem with the latter form of notification format is that the fusion point should have the knowledge to interpretthe sensed value to estimate the intensity of the reported event which leads to spread the interpretation knowledgeof sensing values. Consequently, it results in reducing flexibility especially in heterogeneous sensor networks sincemodifying the threshold of the sensed values should be applied in many nodes. Moreover, it puts the burden ofinterpreting the sensed value of the event fusion point which leads to decrease of inside network processing.

Energy efficiency is also one of the main challenges for most of the sensor network services and applications.Traditionally, nodes submit the sensed values of interest to the base station for information fusion. The centralapproach is prone to several shortcomings such as overspending bandwidth and higher energy consumption, sincenearby nodes transmit the same event redundantly. Reducing the number of message transfers has a considerableimpact on the energy consumption of sensor nodes. The alternative approach is to use distributed event detectionby contributing several fusion points such as cluster heads in a clustered network. Distributed event detection alsofaces several challenges such as dynamic topology and diversity of available sensor types in each cluster.

The last challenge is the scalability and dynamic nature of large wireless sensor networks. Considering a clusterednetwork, various types of sensor nodes may join or leave a cluster making it difficult for a cluster head to keep trackof available sensor types especially in networks with large clusters. A mechanism is required to provide the densityof available sensor types in the cluster. The information will help adaptive event detection, since each cluster head(event fusion point) will make a more accurate decision to either wait for another event type, or report a compositeevent based on previously received events.

There are considerable amount of published papers that tackle the challenges from different perspectives. Wecategorize them into four groups. Application specific event detection approaches [5], [7], [8], [20] are the firstcategory that target issues like energy efficiency, accuracy, and application specific challenges. Their main goal is todevised application-specific tailored solutions. Second category of researches attempts to provide required primitivessuch as, efficient notification service mechanism in the middleware layer [11], [12], [14] to enhance event detectionapplications. They usually consider idealistic models for example in communication and do not address the possibleminor problems such as false positive detection of congestion of communication links. The third group of researchesaddress the uncertainty [11], [15], [21] in event detection, and finally the last category of researches focus ondistributed

approach of event detection [22]–[24]. The main goal is to reach a consensus between detecting nodesin an efficient way in terms of energy and accuracy.

In this paper we devised a generalized model called *FED* for composite event detection. FED benefitsfrom fuzzy modelling in several ways.*FED* applies fuzzy variables to report simple events and their intensity in anabstracted format. Fuzzy membership functions are used for each sensor type to map the sensed values to fuzzyones. Therefore, the fusion points do not need the interpretation knowledge of individual sensor types resulting in asimplification of using heterogeneous sensors. Fuzzy operators are applied to aggregate the reported events. We alsodefine composite events as fuzzy rules. *FED* is fully compatible with our previously designed clustering scheme,called *DEC* [25]. It can also be integrated with our density estimation algorithm [26] to support clusters with theinformation on available sensor types. We evaluated the approach in different node densities, environmental noise,and sensor false detection rate. The results support the idea that *FED* is less sensitive to uncertainty sources. Thedevised fuzzy model can be applied in distributed detection for a clustered network where event notifications areaggregated in cluster heads.

The rest of the paper is organized as follows. In the next section we present the related work. The distributedcomposite event detection problem is formally discussed in Section 3. *FED* model is introduced in detail in Section4. In Section 5 the model is evaluated and finally we conclude in Section 6.

2. RELATED WORK

A wide range of research has been published on event detection in wireless sensor networks. The focus of attention varies from application specific detection to enhancement of middlewares. Some concentrate on uncertainty in event detection while other devise approaches for distributed aspect of detection. In the following we briefly review some of them.

2.1. Application specific event detection

Some of the published research is dedicated to detect events in a particular application such as vehicle tracking,medical diagnosis and military surveillance.

Keally et al. in [7] devised an event detection framework to fulfil user specific requirements mostly on objecttracking. The framework explores the sensing capability of nodes firstly, to perform collaboration between nodes tomeet the required accuracy based on user demands efficiently and secondly, to change detection capabilities basedon runtime observation adaptively.

Hill et al. in [20] reported their experiment in predicting possible events, based on monitoring and analysing a received stream of data sensed by thousands of sensors in an oil field. They introduced an infrastructure foranalysing event detection by real time monitoring in order to detect possible failures. The framework uses fourtiers including, *user tier*, *early event warning tier*, *sensor publisher tier* and *ontology tier* to address the challengessuch as fast response time, maintaining a long history of events, and combining reported events.

Shih et al. in [5] devised an automated approach for detecting seizure in epilepsy patients. Apart from medicalrequirements, building a light weight device with fewer electrodes is considered as requirements for the target system.The use of wireless technology helps in omitting wires which results in lighter devices. They apply machine learningtechniques such as a *support vector machine* (SVM) classifier to construct reduced channel detectors. Consequentlythe seizure is detected with fewer electrodes.

Tian He et al. in [8] design a military surveillance system that enhances a group of sensor devices to detectand track the positions of moving vehicles cooperatively. The main goal is to alert the command and control unitwhenever an event of interest such as moving vehicle happens. Four major requirements are considered for thetarget approach including, longevity, adjustable sensitivity, stealthiness and effectiveness in precision and locationestimate.

The aforementioned research concentrates on a specific application and devises the solutions based on specificapplication conditions. Consequently they do not provide a generic solution which can be applicable to most of theapplications.

2.2. Middlewares for event detection

Some of the researches concentrate on devising middleware [9], [10], [12], [14] to facilitate applications forefficiently reporting the detected events. In the following, we briefly review features devised by middlewares, suchas *TinyDB* [9], *Impala* [10], and *Mires* [12].

Event based query is one of the facilities that *TinyDB* [9] provides for event detection applications. This type ofquery is triggered whenever an explicit event has happened. In other words, based on the sensed value the specifiedevent will raise an interrupt and the query will be executed. In order to use this facility the programmer should writea component to introduce the event and the signals. The defined events can be further used in queries wheneverrequired.

Impala [10] provides an event based programming model for applications. It assigns a specific middleware agentcalled event filter to fulfil the programming model requirements. The event filter agent is responsible for capturingand dispatching detected events to other middleware agents as well as applications.

Souto et al. in [12] devised a *publish/subscribe* middleware called *Mires*. It provides primitives for publishingdetected events for the subscribed nodes. The publish/subscribe approach used in *Mires* provides an asynchronouscommunication between the elements of a network. This is a valuable advantage in a dynamic nature of WSN. Theevent detection mechanism in *Mires* has three phases. In the first phase nodes advertise their sensing capabilitiesas *Topics*. The advertised messages are then sent to the sink node via a multihop routing protocol. In the secondphase user applications connect to the sink and subscribe those sensing capabilities which they are interested in.In the last phase subscribed messages broadcast down the network. Receiving the subscribed messages, nodes cannotify the detected events (*topics*).

Middleware usually addresses the node level event capturing and dispatching. They provide a programming modelto raise events, which are usually simple events, based on the sensed values. The distribution and aggregation ofthese events is the second aspect of these middlewares. On the other hand they do not address the detection ofcomposite events. Besides, they usually do not specify the architecture for distributed detection. Consequently, theseaspects are mainly forwarded to application layer.

2.3. Uncertainty in event detection

Several approaches have been devised so far to cope with the effect of uncertainty in event detection.

Heinzelman et al. [11] has introduced a proactive service oriented WSN middleware called *MiLAN*. One of theinteresting aspects of the middlware is the capability of switching between sensors with different sensing accuracy.*MiLAN* is able to handle heterogeneous nodes with different sensing accuracy (*Quality of Service*). Applicationsrunning above the middleware

layer are powered by the capability to identify their accuracy needs based onapplication states. Generally, uncertainty in event detection contains wider range of issues than *QoS*. *MiLAN* isa remarkable research in dynamically handling accuracy in sensing but it does not address issues such as falsepositive detection, event notification loss and aggregating uncertainties.

S.Li et al. in [15] has defined an event detection service (*DSWare*) using a data centric approach. It supportsdetecting CEs in a sensor network with heterogeneous nodes. An application program can register events bysubmitting an *SQL-like* statement to a group of specified nodes. In order to address CEs, sub-event sets are definedin the statement. The definition of a sub-event consists of several parameters, such as a confidence function anda minimum confidence value for detecting it. To cope with uncertainty *DSWare* uses confidence functions. Aconfidence function takes occurrence of sub-events, in a Boolean data type format, as an input parameter andcomputes a numeric value showing how likely the CE has happened. *DSWare* aggregates the reported events alongthe path to the sink. Consequently, it is not applicable for a clustered network. It also does not provide node levelabstraction in interpreting the sensed values.

Ambiguity in knowledge acquisition for defining composite events is the issue that Manjunatha addresses in[27]. According to the proposed approach, sensors submit their sensed values to an aggregation point. The meanof transmitted values are considered as aggregated value. Then the aggregated value is fuzzified and the inferenceengine looks for any possible composite event, defined as fuzzy rules. Although the approach has some similaritieswith our work, our model differs on several points from [27] in several points. Firstly, Manjunatha in [27] does notaddress false positive detection issues. Secondly, the proposed approach does not consider unreliable communicationand message loss which may cause uncertainty in event detection. Thirdly, sensor nodes transmit the sensed values,which violate node level abstraction in interpreting. Besides, the aggregation method is not appropriate for falsedetection situations. Finally, it seems that the inference engine does not consider time and location correlation indetecting composite events.

Samarasooriya et al. [21] have introduced a fuzzy modelling approach in dealing with uncertainty. The mainfocus is the varying degrees of accuracy in local sensors, specifically the local sensors error probabilities whichare varying in time in a non-random fashion. In other words, they target node level uncertainty in detecting events.They modelled the error probabilities with fuzzy quantities using membership functions. They used a probabilisticapproach to fuse the local sensor decisions and formally prove the performance of their model. Although theytry to model node level error probability, they do not devise a solution at fusion point which includes unreliablecommunication.

2.4. Distributed event detection

From distributed detection point of view,several remarkable researches published so far.

Viswanathan et al. in [22] analyze several distributed detection (*distributed signal processing*) architectures byapplying *Neyman-Pearson formulation*. They investigate the computational complexity in achieving the optimalsolution. Parallel topology with/without fusion center as well as serial and tree topology were studied. They comparethe serial architecture, in which each node makes a decision based on its observation as well as the received decisionsfrom its neighbors and then forwards its decision to the next node in a serial way. One of the important outcomesof the research is, for the case where large number of participate in the distributed detection process, the probabilityof missing event goes to zero with a much slower pace in the serial architecture compared to the parallel one.

Kumar et al. in [28] devised a framework for developing distributed data fusion applications called *DFuse*. Itconcentrates on two main aspects. The first one is providing a wide range of *fusion APIs* to facilitate applicationsin complex information aggregation such as video streams. The second characteristic is the distributed algorithmfor placement of fusion function. The main goal is to find out the optimum placement of fusion point to minimizecommunication cost. *DFuse* provides a heuristic approach to choose a suitable fusion point based on predefinedcost function. The placement process re-evaluated periodically to address network dynamics.

3. PROBLEM DEFINITION

We consider a network consist of m heterogeneous nodes, each node may have different subset of availablesensors.

$$N = \{n_i : i \in m\} \quad (1)$$

Let v be the available sensor types in the network and C_i the capability tuple of node i. Each element in the $C_i$represents a flag for a sensor type. The value of one for s_k indicates that the node is equipped with sensor type kand value of zero shows nonexistence.

$$C_i = (s_k : k \in v) \quad (2)$$

Each node sends its observation upon detection of a simple event. Let y_i be a reported local observation ofsensor node i and u be the global (fusion point) decision on composite event detection. The Eq. 3 shows themapping of local decisions to the composite event detection.

$$u = \gamma_0\big(y_1(.), y_2(.), \dots, y_m(.)\big) \quad (3)$$

Let $\Gamma(\gamma_0, \gamma_1, \gamma_2, \dots)$ be the set of rules that defines composite events, where $\gamma_0(.)$ is the definition of a samplecomposite event. Considering the following probabilities, the goal of the model is to increase $\Lambda(u)$ which is thelikelihood of detection. In the realistic model transmitted observations may fail to reach the destination due tomessage loss. Besides, sensor nodes may have false positive detection due to various reasons including hardwaremalfunctioning.

P_F: $P(u = 1 | H_0)$: global false alarm probability

P_D: $P(u = 1 | H_1)$: global detection probability

P_M: $P(u = 0 | H_1)$: global miss probability

$$\Lambda(u) = \frac{P(u \mid H_1)}{P(u \mid H_0)} \quad (4)$$

In general three conditions are required to detect any CE. These conditions are:

1) In order to detect any CE, a group of predefined simple event types must have been detected. That is, a CE is defined as a set of simple event types.
2) The occurrence timestamp of the reported simple events should be within a limited time period (called *detection window*), which has been defined in the CE definition.
3) The nodes, reporting the simple events should be close enough in terms of geographical distance. That is, in order to infer any correlation among simple events, there should be rational physical distance among reporting nodes. The rational distance between nodes can be calculated based on the sensing range of each sensor. It can also be measured in terms of hop counts.

Given the above detection requirements, the devised approach should fulfil the following goals.

- Detect composite events in the presence of message loss and false positive simple event detection.
- Support heterogeneity of sensor nodes in terms of sensing capability.

- Be expendable for distribute detection in large scale networks.

4. FED MODEL

In this section we describe different aspects of the model. First, we describe the network architecture and in thesecond part, we explain event notification. In the third part, we present composite event detection and finally, weaddress uncertainty problem.

4.1. Network architecture

FED uses clustered network architecture to perform distributed event detection and prevent redundant submissionof simple events to sink node. The cluster heads have the responsibility of aggregating the reported events withintheir clusters and forward the detected composite events to the sink node.

Choosing an appropriate clustering scheme is an important issue in the efficiency of the model. Recall from thedefinition of composite events, a variety of sensor types are required to collaborate in order to detect a compositeevent. Therefore, the clustering scheme should maximize the diversity of sensor types in each cluster to increasethe cluster capabilities in detecting composite events. Traditional clustering schemes such as [29]–[34] fail to fullysatisfy this requirement. They usually do not consider sensor diversity as a clustering parameter.

We have introduced a diversity base clustering scheme called *DEC* [25] to increase the capability of detectingvarious composite events. *DEC* performs four phases, which are *initialization*, *clusterjoining*, *migration*, and*termination*, to generate clusters with maximum possible diversity of sensor types. It applies a cost functionthat uses the residual energy level of a node and the diversity of its neighboring nodes, to elect a cluster head. Tohandle the dynamic nature of wireless sensor networks, *migration* phase of the algorithm has the duty of balancingthe capabilities of the cluster in case where some sensors fail. That is, whenever a scare sensor type fails, thecluster head invites nodes with the same sensor type to migrate. The invitation will be accepted, if it is granted bythe node's current cluster head. The simulation results in [25] show that it produces clusters with higher sensingcapabilities for enhancing event detection applications. For more detailed information on *DEC* please refer to [25].

It should be added that there is also an on-going research to devise a clustering scheme for the mobile nodes toadaptively maximize sensor diversity in each cluster. The first step is to estimate the diversity of sensor types [26]in a mobile network. The estimation will be further used in order to provide the required clustering scheme for themobile network.

4.2. Simple events

Simple event notification is the building block of *FED*. We apply fuzzy logic [35] to provide a node levelabstraction on interpreting the meaning of the sensed values. Based on *FED* model, we fuzzify the sensed valuesinside the sensing node. One of the advantages is to provide a node level abstraction on the meaning of the value.Therefore, the aggregation point is not required to have a full knowledge over all sensor types in the heterogeneoussensor networks.

In *FED* each sensor type is associated with a fuzzy variable and consequently each fuzzy variable has severalfuzzy values. The values are defined based on the specified simple events by the application. Although submittingthe *fuzzy value* of the detected simple event is adequate for applications that only need simple event detection,composite event detection applications need more information on the membership degree of the fuzzy value inorder to aggregate the simple event notifications.

To convert the sensed values into fuzzy ones, a membership function is required. Figure 1 shows a samplemembership function for a heat sensor. The X axis is the temperature degree in Celsius while Y axis shows themembership degree. Values below 26 are out of concern as the threshold is set to be 26°C.

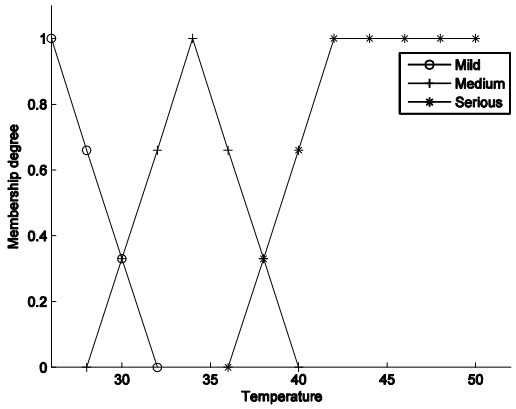

Figure1. Sample fuzzy membership function for heat sensor

The next step is to prepare the event notification message. Simple event notifications in *FED* consist of fivefields. Some of these fields are similar to the traditional event notification format. Equation 5 shows the simpleevent notification in *FED* where e_{name}, n_{id}, t, f_{value} and $d_{membership}$ are event name, node ID, event time, fuzzyvalue and membership degree respectively.

$$y_i = (e_{name}, n_{id}, t, f_{value}, d_{membership}) \quad (5)$$

For example, in an indoor fire detection application the temperature above 26° should be reported. Consider anode with the ID of 43 that has sensed the temperature of 37°at 12:23:39. The following event notification willbe reported. *Temp* shows that a temperature event has happened. The second field shows the ID of the reportingnode while the third element shows the time (more precise time formats can be used to present timestamp field) inwhich the event has been detected. The fuzzy value and related membership degree are the fourth and fifth fieldsrespectively.

$$\text{event} = (\text{Temp},43,12{:}23{:}39\ \text{GMT},\text{serious},0.20) \quad (6)$$

4.3. Composite events

Recall from Section 3, the occurrence of a set of predefined simple events is one of the conditions of detectingcomposite events. Considering the fact that simple events are reported as fuzzy notifications, we define compositeevents as fuzzy rules. Here is a sample composite event that has been defined in a fuzzy rule format.

γ_j:*If Heat is MEDIUM and Humidity is LOW and Smoke is Medium then Fire is SERIOUS*

The IF clause shows the set of simple events required to detect the composite event, and the THEN clausespecifies the composite event name. In the composite event definition, each simple event is associated with a fuzzyvalue. The values are used to estimate the likelihood of the composite event.

FED uses two aggregation methods to fuse the transmitted simple events. The first method is used to fuse thesimple events of the same type and the second one is used for the final aggregation. The fuzzy operator OR is usedto aggregate same type simple events.

The next step is to investigate if the possibly correlated notifications can satisfy any fuzzy rule. In order tocalculate the occurrence degree of the detected CE, notifications are aggregated weighted average function. Theactual weight for each simple event type of a specific fuzzy rule (composite event) is chosen based on application requirements. The more important event types will weight higher compared to less important ones. It is also possibleto adaptively choose the weights based on the redundancy of each simple event that has been received. The result ofthe fuzzy rules will be the composite event, its fuzzy value, and the degree of membership. The fuzzy value showshow serious is the detected event and the degree of membership shows the likelihood of the detected compositeevent. Higher membership degrees indicate that the CE has happened with higher certainty and lower values, viceversa.

4.4. Detection process

The event detection process consists of three main activities, which are *simple event detection*, *composite event detection*, and *event stream maintenance*.

Ordinary sensor nodes are responsible for detecting simple events. The activity starts when a sensor node detectsan event (*event name*) based on the sensed values and predefined threshold values for the simple events. In the nextstep, the node maps the sensed value to a fuzzy one (*fuzzy value*) using the fuzzy membership function, andcalculate the corresponding membership degree. Finally in the last step, the complementary information such as*node ID* and *detection timestamp* are added to the event notification and submitted to the fusion point.

The coordinator is responsible for composite event detection based on the received event notifications from itscluster members. The process triggering mechanism is a design issue aspects of the process. The process of detectingcomposite events can be triggered either by arrival of new event notification, or by a timer. The former providesfaster detection with the price of increased processor consumption, while the detection speed in the latter is sensitiveto the timer value. The occurrence frequency of an event, which is an application specific parameter, is one of thefactors that play an important role in choosing either case.

One of the other responsibilities of the coordinator is to analyze the correlations between the received notifications.In *FED* the event correlation is investigated from two different aspects. The first parameter is the time distance(*detection window*) between the reported simple events and the second one is the physical distance between thereporting nodes. The output of the correlation analysis step is a set of event notifications that should be examined tocover a possible composite event. Then the correlated notifications are aggregated based on the methods describedin Section 4.3.

Maintenance of the received event streams at the fusion point helps to improve the efficiency of the model. Oneof the main responsibilities of cluster heads in *FED* is to keep track of the correlated simple events. Therefore, thereported event stream has to be scanned frequently by the cluster head. It is highly recommended to keep the storedstream as short as possible to save memory and processor resources. One of the ways to keep the stream shortis to set an expiration time for the stored events. Consequently, the expired event notifications will be removedautomatically. Two parameters should be considered in defining the expiration time effectively. The first parameteris the largest detection window for the composite event and the second one is the time synchronization accuracy[36].

4.5. Uncertainty

Generally, a predefined set of simple events should happen in order to be able to detect a composite event. Thereare cases where at least one of the required simple events misses, due to various reasons. To cope with the problem,*FED* calculates the occurrence likelihood of the

composite event. The calculation is similar to aggregation of simpleevents. The output of the calculation is a membership degree of a possible composite event in the absence of arequired simple event. *FED* uses threshold called acceptance ratio, to recognize the reported simple events as acomposite event. Lower values for acceptance ratio threshold will result on higher false positives while highervalues will disable the uncertainty handling of *FED*.

For example consider the fuzzy rule of Section 4.3 and the following event notifications from the two nearbysensor nodes.

$$y_1 = (\text{heat},12,12{:}23{:}\ 40\ \text{GMT},\text{medium},0.90) \quad (7)$$

$$y_2 = (\text{smoke},20,12{:}23{:}\ 39\ \text{GMT},\text{medium},0.85) \quad (8)$$

In the absence of the third simple event, *FED* calculates the average of the membership degrees which is 0.583. Ifthe acceptance ratio for the rule is below the calculated number, the composite event will be detected.

5. EVALUATION

To evaluate the performance of the introduced model on detecting composite events under uncertainty sources,several of simulation configurations were setup. Before analyzing the achieved results, we would like to introducethe simulation tool first.

5.1. Simulation environment

We have developed a simulation environment based a software agent development tool called JADE [37]. *JADE*is a framework for developing multi-agent systems. These are some of the advantages of using *JADE* for simulatingwireless sensor network algorithms.

- The framework encapsulates the network protocol stack and helps to ignore the hardware level details of sensor nodes. Considering each node as a software agent gives us a chance to analyse the higher level behaviour of the algorithm in the network.
- The autonomous property of software agents maps well with sensor nodes' autonomous nature.
- Message passing is considered as the only communication mechanism in *JADE* and wireless sensor networks. That is, there are no method calls or shared memory facilities in both cases. Arrived messages are stored in a FIFO queue in each agent in *JADE* which is similar to a sensor node. Thisalso provides the means of having an asynchronous communication.

The aforementioned features give us sufficient reasons to build our simulation based on *JADE*. But in order to fitthe tool with the problem criteria, we have added five additional features.

1) Event notifications omitted randomly to simulate message loss in the network.
2) To simulate false detection behavior of nodes, a random false detection mechanism is added to sensor nodes.
3) We have added a delay in message transmission in order to simulate propagation delay using a Gaussianrandom generator.
4) Sensor nodes are only allowed to communicate with each other, only if they are within each other's transmission range. All the sensor nodes can communicate with the coordinator in a bidirectional way.
5) An event generator creates CEs by defining the exact location randomly and sends the message to only thosenodes that are within the sensing rage of the generated event.

5.2. Simulation results

To evaluate *FED*, we simulated an experimental case similar to explosion detection application. In our simulation,heterogeneous nodes are distributed in the environment uniformly random. We use three types of sensors(SensorType1, SensorType2 and SensorType3) with equal quantities. In each experiment a sensor field is generatedwith predefined number of nodes that has been located uniformly distributed. 100 CEs with random locations aregenerated periodically across the network area of 600 * 600 unit^2. To reduce the effect of the random distributionwe have run each experiment 50 times and averaged the results.

In the first set of experiments, we investigate the effect of sensing coverage area of a sensor and the message lossrate on the percentage of detected composite events. Figure 2 shows the simulation results for a network consistsof 12 nodes from 3 sensor types (4 sensors from each type) which are deployed randomly. Besides, false simple event reports are produced randomly with rate of 5%. The evaluation metric (Y axis) is the percentage of detectedcomposite events. The performance of *FED* is compared with the approach introduced in [27]. According to theexperiment scenario, we increase the sensing range of sensor nodes from 240 to 380 units. The lower sensing rangerepresents less dense network, while the higher sensing ranges show the dense ones. The reason is, in the highersensing ranges each point in the network field may be covered by more nodes, whereas in lower sensing rangesit will be covered by fewer nodes. The first thing that the simulation results show is that, increasing the sensingrange will provide higher detection percentage of composite events.

To investigate the effect of message loss, we apply two different message loss rates. In the first case 25% ofsimple event notifications are omitted randomly. The results show that *FED* outperforms 15% with the sensingrange of 240 units. As the sensing range increases, both approaches achieve higher percentage of event detection.

The results also show that *FED* is less sensitive to message loss. That is, the difference between the percentagesof detected events in the lowest and highest sensing ranges for the case of 25% of message loss is 35% for *FED*and 42% for [27]. The results for the case of 15% of message loss also support the idea.

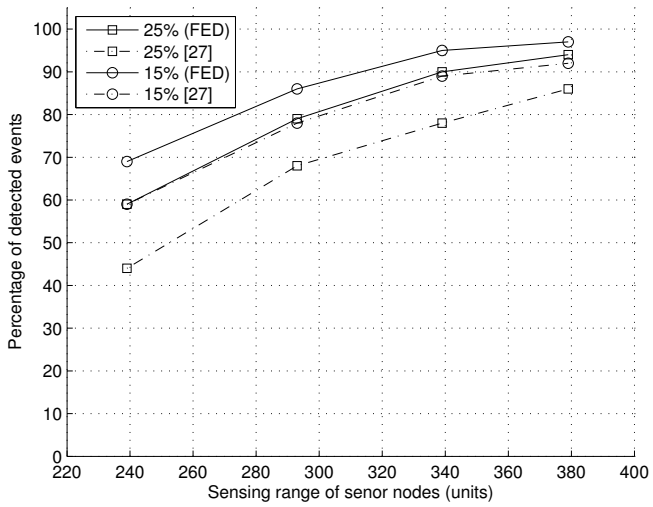

Figure2. CE detection in a network with node density of 3 false alarm 5%.

We run the simulation with similar configuration with 24 nodes. Figure 3 shows the simulation results, whereX axis is the sensing range and the Y axis is the percentage of detected composite

events. The results show that*FED* performs better in all the sensing ranges. For instance, *FED* detects 57% of composite events for the casewhere sensing range is 169, while the approach in [27] detects only 43% in the presence of 25% message loss.The difference between the detection percentages of both approaches reduces in the dense network, due to multipledetection of a simple event. But even in the sensing range of 268, *FED* detects 6% more composite events. Moreover, similar to previous experiment, the results show that *FED* is less sensitive to environmental noise. For example,the difference between the highest and lowest percentage of event detection is 37% and 46% for *FED* and [27]respectively with 25% of message loss. In the network with 15% of message loss, the difference is 30% and 40%for *FED* and [27] respectively.

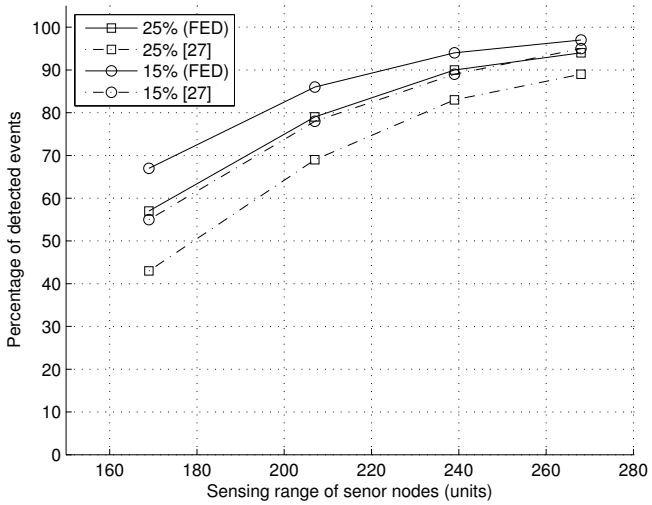

Figure3. CE detection in a network with node density of 3 false alarm 5%.

Figure 4 and Figure 5 shows the experimental results for the network of 36 and 48 nodes respectively. We alsoincrease the false alarm rate to 10%. The outcome of the experiments is similar to the previous analysis.

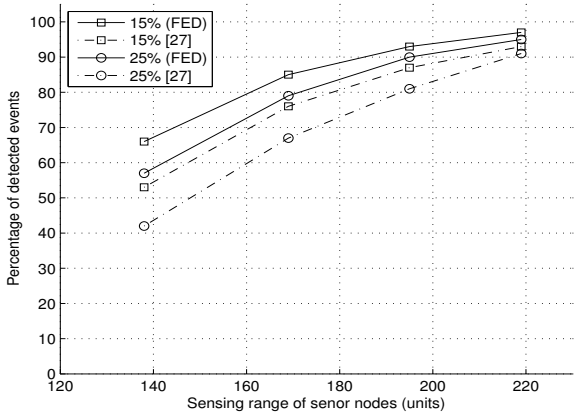

Figure4. CE detection in a network with node density of 4 false alarm 10%.

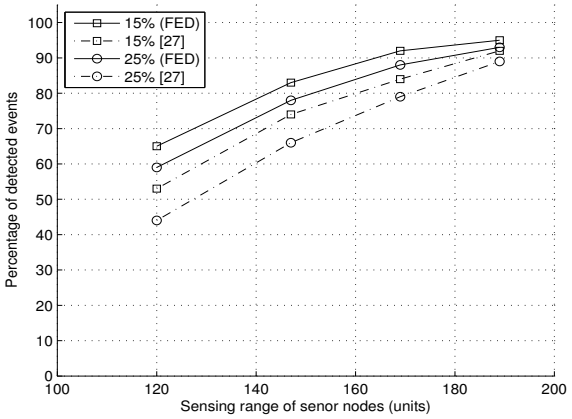

Figure5. CE detection in a network with node density of 4 false alarm 10%.

In the next set of experiments we investigate the effect of message loss in detail. The network surface is the sameas previous experiments and the quantity of each sensor type is equal to $\frac{1}{3}$of network nodes. Figure 6 demonstratesthe percentage of detected events (Y axis) for various message loss rates (X axis). The graph shows the results forthe network of 12 and 24 nodes and false alarm rate is set to be 10%. The sensing range is defined in way that thesenetwork configurations have the same node density. The sensing range for the network of 12 nodes is set to be 239units and for the case of a 24 node network is 169 units. To calculate the proper sensing range we use the followingequation where d, L, W, and n are network node density, length, width, and size (number of nodes) respectively.

$$t_r = \sqrt{\frac{dLW}{2\pi n}} \qquad (9)$$

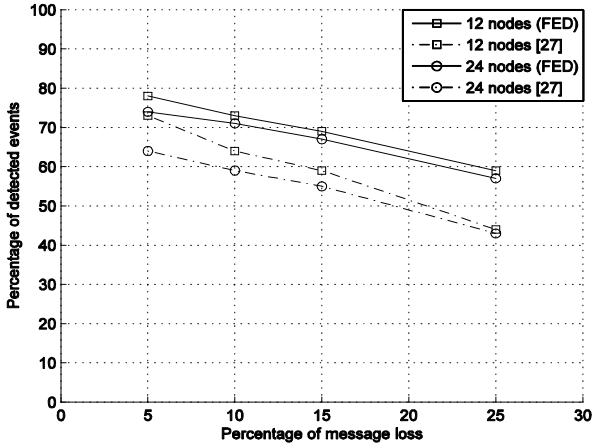

Figure6. *FED* sensitivity to quantity of nodes under 10% false alarm rate.

Similar to the previous results, *FED* detects more events compared to the approach introduced in [27]. For examplein a network of 12 with 5% message loss, *FED* detects 5% of events that is not detected by [27]. As the messageloss rate increase, the percentage of detected events in both

approaches decrease. But the results show that *FED* is more robust message loss. For the case of 25% message loss in a network of 12 nodes, *FED* uncovers more than15% of undetected events by [27], while in the presence of 5% message loss was 5%.

Figure 7 depicts the results for a similar experiment with 36 and 48 nodes. The node density is 3 which means that the sensing range for each node in the 36 node network is 169 units and for the network of 48 nodes is 148units and the false alarm rate is 10%.

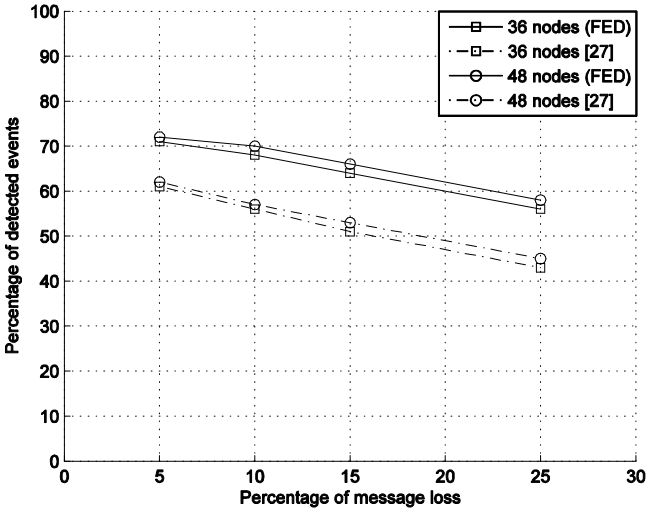

Figure7. *FED* sensitivity to quantity of nodes under 5% false alarm rate.

6. CONCLUSIONS

In this paper we have introduced a fuzzy model called *FED* for distributed detection of composite events. *FED* supports the heterogeneity of sensor types. To perform distributed detection, the model uses a diversity based clustering scheme in which each cluster considers maximizing the sensor diversity of their member nodes.Therefore, the clusters are able to detect a wide range of composite events. Besides, the model provides node level knowledge abstraction which is a valuable characteristic in designing heterogeneous sensor networks. A fuzzy approach is used to uncover composite events in the presence of uncertainty sources such as message loss and false alarms. The simulation results show that *FED* outperforms in terms of the percentage of detected events. The amount of improvement is significant in networks with low node density. Besides, *FED* is suitable for the networks with high message loss rates.

ACKNOWLEDGEMENTS

This research is funded by Iranian Telecommunication Manufacturing Company (ITMC).

REFERENCES

[1] M. Ceriotti, L. Mottola, G. P. Picco, A. L. Murphy, S. Guna, M. Corra, M. Pozzi, D. Zonta, and P. Zanon, "Monitoring heritage buildings withwireless sensor networks: The torreaquila deployment," in Proceedings of the 2009 International Conference on Information Processingin Sensor Networks, ser. IPSN '09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 277–288.

[2] R. Szewczyk, A. Mainwaring, J. Polastre, J. Anderson, and D. Culler, "An analysis of a large scale habitat monitoring application," inProceedings of the 2nd international conference on Embedded networked sensor systems, ser. SenSys '04. New York, NY, USA: ACM,2004, pp. 214–226.

[3] K. Martinez, J. K. Hart, and R. Ong, "Environmental sensor networks," Computer, vol. 37, pp. 50–56, 2004.

[4] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," ACM Comput. Surv., vol. 38, December 2006.

[5] E. I. Shih, A. H. Shoeb, and J. V. Guttag, "Sensor selection for energy-efficient ambulatory medical monitoring," in Proceedings of the7th international conference on Mobile systems, applications, and services, ser. MobiSys '09. New York, NY, USA: ACM, 2009, pp.347–358.

[6] M. F. Duarte and Y. H. Hu, "Vehicle classification in distributed sensor networks," J. Parallel Distrib.Comput., vol. 64, pp. 826–838, July2004.

[7] M. Keally, G. Zhou, and G. Xing, "Watchdog: Confident event detection in heterogeneous sensor networks," Real-Time and EmbeddedTechnology and Applications Symposium, IEEE, vol. 0, pp. 279–288, 2010.

[8] T. He, S. Krishnamurthy, L. Luo, T. Yan, L. Gu, R. Stoleru, G. Zhou, Q. Cao, P. Vicaire, J. A. Stankovic, T. F. Abdelzaher, J. Hui, andB. Krogh, "Vigilnet: An integrated sensor network system for energy-efficient surveillance," ACM Trans. Sen. Netw., vol. 2, pp. 1–38,Feb. 2006.

[9] S. R. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, "Tinydb: an acquisitional query processing system for sensor networks,"ACM Trans. Database Syst., vol. 30, pp. 122–173, March 2005.

[10] T. Liu and M. Martonosi, "Impala: a middleware system for managing autonomic, parallel sensor systems," SIGPLAN Not., vol. 38, pp.107–118, June 2003.

[11] W. B. Heinzelman, A. L. Murphy, H. S. Carvalho, and M. A. Perillo, "Middleware to support sensor network applications," IEEE Network,vol. 18, no. 1, pp. 6–14, 2004.

[12] E. Souto, G. Guimar˜aes, G. Vasconcelos, M. Vieira, N. Rosa, and C. Ferraz, "A message-oriented middleware for sensor networks," inProceedings of the 2nd workshop on Middleware for pervasive and ad-hoc computing, ser. MPAC '04. New York, NY, USA: ACM,2004, pp. 127–134.

[13] T. Abdelzaher, B. Blum, Q. Cao, Y. Chen, D. Evans, J. George, S. George, L. Gu, T. He, S. Krishnamurthy, L. Luo, S. Son, J. Stankovic,R. Stoleru, and A. Wood, "Envirotrack: Towards an environmental computing paradigm for distributed sensor networks," DistributedComputing Systems, International Conference on, vol. 0, pp. 582–589, 2004.

[14] R. Kumar, M. Wolenetz, B. Agarwalla, J. Shin, P. Hutto, A. Paul, and U. Ramachandran, "Dfuse: a framework for distributed data fusion,"in Proceedings of the 1st international conference on Embedded networked sensor systems, ser. SenSys '03. New York, NY, USA: ACM,2003, pp. 114–125.

[15] S. Li, S. H. Son, and J. A. Stankovic, "Event detection services using data service middleware in distributed sensor networks," in Proceedingsof the 2nd international conference on Information processing in sensor networks, ser. IPSN'03. Berlin, Heidelberg: Springer-Verlag,2003, pp. 502–517.

[16] J. Mao, J. Jannotti, M. Akdere, and U. Cetintemel, "Event-based constraints for sensornet programming," in Proceedings of the secondinternational conference on Distributed event-based systems, ser. DEBS '08. New York, NY, USA: ACM, 2008, pp. 103–113.

[17] R. Rajkumar, I. Lee, L. Sha, and J. Stankovic, "Cyber-physical systems: The next computing revolution," in Design Automation Conference(DAC), 2010 47th ACM/IEEE, june 2010, pp. 731 –736.

[18] S. Ren, "Apeser 2010 keynote speech: Shangpingren," in Green Computing and Communications (GreenCom), 2010 IEEE/ACM Int'lConference on Int'l Conference on Cyber, Physical and Social Computing (CPSCom), dec. 2010, p. lvix.

[19] E. Lee, "Cyber physical systems: Design challenges," in Object Oriented Real-Time Distributed Computing (ISORC), 2008 11th IEEEInternational Symposium on, may 2008, pp. 363 –369.

[20] M. Hill, M. Campbell, Y.-C. Chang, and V. Iyengar, "Event detection in sensor networks for modern oil fields," in Proceedings of thesecond international conference on Distributed event-based systems, ser. DEBS '08. New York, NY, USA: ACM, 2008, pp. 95–102.

[21] V. N. S. Samarasooriya and P. K. Varshney, "A fuzzy modeling approach to decision fusion under uncertainty," Fuzzy Sets Syst., vol. 114,pp. 59–69, August 2000.

[22] R. Viswanathan and P. Varshney, "Distributed detection with multiple sensors i. fundamentals," Proceedings of the IEEE, vol. 85, no. 1,pp. 54 –63, Jan. 1997.

[23] V. Saligrama, M. Alanyali, and O. Savas, "Distributed detection in sensor networks with packet losses and finite capacity links," SignalProcessing, IEEE Transactions on, vol. 54, no. 11, pp. 4118 –4132, 2006.

[24] E.Ermis and V.Saligrama, "Distributed detection in sensor networks with limited range multimodal sensors," Signal Processing, IEEETransactions on, vol. 58, no. 2, pp. 843 –858, 2010.

[25] H. TabatabaeeMalazi, A. Khalili, K. Zamanifar, and S. Dulman, "DEC: Diversity based energy aware clustering for heterogeneous sensornetworks," Ad Hoc and Sensor Wireless Networks (Accepted).

[26] H. TabatabaeeMalazi, K. Zamanifar, A. Pruteanu, and S. Dulman, "Gossip-based density estimation in dynamic heterogeneous sensornetworks," in Wireless Communications and Mobile Computing Conference (IWCMC), 2011 7th International, July 2011, pp. 1365 –1370.

[27] P. Manjunatha, A. Verma, and A. Srividya, "Multi-sensor data fusion in cluster based wireless sensor networks using fuzzy logic method,"in Industrial and Information Systems, 2008.ICIIS 2008.IEEE Region 10 and the Third international Conference on, 2008, pp. 1 –6.

[28] A. V. U. Phani Kumar, A. M. Reddy V, and D. Janakiram, "Distributed collaboration for event detection in wireless sensor networks,"in Proceedings of the 3rd international workshop on Middleware for pervasive and ad-hoc computing, ser. MPAC '05. New York, NY,USA: ACM, 2005, pp. 1–8.

[29] H. Chan and A. Perrig, "ACE: An emergent algorithm for highly uniform cluster formation," in European Workshop on Wireless SensorNetworks (EWSN 2004), Jan. 2004.

[30] S. Soro and W. B. Heinzelman, "Cluster head election techniques for coverage preservation in wireless sensor networks," Ad Hoc Netw.,vol. 7, pp. 955–972, July 2009.

[31] O. Younis and S. Fahmy, "Distributed clustering in ad-hoc sensor networks: a hybrid, energy-efficient approach," in INFOCOM 2004.Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies, vol. 1, 2004, pp. 4 vol. (xxxv+2866).

[32] S. K. Singh, M. P. Singh, and D. K. Singh, "Energy efficient homogenous clustering algorithm for wireless sensor networks," International Journal of Wireless & Mobile Networks (IJWMN), vol.2, no.3, August 2010.

[33] Getsy S Sara, Kalaiarasi.R, NeelavathyPari.S and Sridharan .D, "Energy efficient clustering and routing in mobile wireless sensor network," International Journal of Wireless & Mobile Networks (IJWMN) vol.2, no.4, November 2010.

[34] S.Taruna, Kusum Jain and G.N. Purohit, "Power Efficient Clustering Protocol (PECP)-Heterogeneous Wireless Sensor Network," International Journal of Wireless & Mobile Networks (IJWMN) vol.3, no.3, June 2011.

[35] L. Zadeh, "Fuzzy sets," Information and Control, vol. 8, no. 3, pp. 338 – 353, 1965.

[36] B. Sundararaman, U. Buy, and A. D. Kshemkalyani, "Clock synchronization for wireless sensor networks: a survey," Ad Hoc Networks,vol. 3, no. 3, pp. 281 – 323, 2005.

[37] Java agent development framework @ONLINE. [Online]. Available: http://jade.tilab.com/