

Research Article

On Event/Time Triggered and Distributed Analysis of a WSN System for Event Detection, Using Fuzzy Logic

Sofia Maria Dima, Christos Antonopoulos, and Stavros Koubias

Department of Electrical and Computer Engineering, University of Patras, Eratosthenous 6, Rio, 26500 Patras, Greece

Correspondence should be addressed to Sofia Maria Dima; sdima@ece.upatras.gr

Received 14 December 2015; Accepted 15 March 2016

Academic Editor: Matthew J. Whelan

Copyright © 2016 Sofia Maria Dima et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Event detection in realistic WSN environments is a critical research domain, while the environmental monitoring comprises one of its most pronounced applications. Although efforts related to the environmental applications have been presented in the current literature, there is a significant lack of investigation on the performance of such systems, when applied in wireless environments. Aiming at addressing this shortage, in this paper an advanced multimodal approach is followed based on fuzzy logic. The proposed fuzzy inference system (FIS) is implemented on TelosB motes and evaluates the probability of fire detection while aiming towards power conservation. Additionally to a straightforward centralized approach, a distributed implementation of the above FIS is also proposed, aiming towards network congestion reduction while optimally distributing the energy consumption among network nodes so as to maximize network lifetime. Moreover this work proposes an event based execution of the aforementioned FIS aiming to further reduce the computational as well as the communication cost, compared to a periodical time triggered FIS execution. As a final contribution, performance metrics acquired from all the proposed FIS implementation techniques are thoroughly compared and analyzed with respect to critical network conditions aiming to offer realistic evaluation and thus objective conclusions' extraction.

1. Introduction

In recent years, wireless sensor networks (WSNs) have emerged as a promising research field and have been applied to a wide variety of application domains including industrial control, environmental monitoring, and healthcare applications. The primary objective in such WSN applications is the accurate and reliable monitoring of an environment, based on the processing of multiple and diverse sensors values and the identification of irregular situations or dynamic real life events. The collaborative tasks lead to specific action scenarios, so as to control the monitored environment. The process of observing a real phenomenon and evaluating its behaviour in WSNs is known as *event detection* [1].

With respect to the dependency upon the input signals, real events are distinguished in two categories: single modality and multimodality events [2]. The former concerns the examination of the monitored values of each parameter independently, based on the assumption that if any of these

exceed a specific "normal" range, an event occurs [2]. The latter category includes the multimodal events which are based on the correlation of several attributes, the processing of which evaluates the occurrence of an event [3]. Critical challenges of event detection algorithms in WSN include energy saving, data integrity, and in-depth understanding of the monitored environment. In order to meet such objectives, the development of a classification model is essential for the accurate identification of an event, along with the reduction of the communication as well as processing overhead. The classification of an event can be defined as the process of evaluating an event of interest using multiple sensor nodes (multimodal event). Such processing may vary from a trivial rule engine machine to a complex machine learning algorithm, while the final outcome of this process triggers specific action scenarios. Taking into consideration typical WSN characteristics, the classification of an event is strongly affected by the quality and characteristics of the communication channel between the monitoring and actuation units

in order to optimize the monitoring and control of the environment.

Considering an application which is based on single modality events, when a sensor value exceeds an upper/lower threshold (i.e., temperature in an environmental monitoring application), an event generation is indicated (i.e., fire alarm). However, in many cases, such decisions may lead to false alarms, since most of the real life events depend on multiple monitored parameters in a correlated manner. For example, in the fire alarm scenario, an accurate decision should take into consideration the existence of smoke and luminosity level along with the temperature value. Alarm situations trigger specific reactions and, thus, node-to-node communication (actor-to-actor coordination schemes) [4]. Therefore, false alarms will lead to an application's performance degradation, as well as to a network traffic and energy consumption increase. Hence, the need for more sophisticated multimodality classification processes that will maximize the application's accuracy while mitigating resource wastage among the actuation units arises. Additionally, taking into consideration that the communication modules are the most energy consuming components of a sensor node, the lifetime of the node and the network's robustness are anticipated to be accordingly benefited. In that respect, the utilization of classification algorithms [5] (i.e., fuzzy inference systems, FIS) in WSNs regarding the identification of complex events is crucial in achieving the aforementioned objectives.

Towards this goal, existing data mining techniques can offer several algorithmic solutions [6] to this field. However, conventional implementation requires high processing capabilities and abundant memory availability, in order to meet specific execution time restrictions. Such assumptions contradict typical WSN characteristics, where the sensor nodes suffer from limited processing power and available memory. Such characteristics, in combination with the error prone nature of wireless communications, highlight the challenge for designing distributed, highly efficient, yet of low complexity and low resource-demanding data mining algorithms in WSNs. Furthermore, in realistic WSN applications, the distributed implementation of such computational intensive algorithms can be highly beneficial towards balancing CPU load among several nodes. Specifically, the distributed implementation of an algorithm increases on-site processing and can potentially reduce the number of data packet transmissions, leading to bandwidth conservation, network data transfer relaxation, and energy consumption degradation.

Another critical aspect drastically affecting FIS algorithm design when aiming towards WSN application concerns the effect of network characteristics upon the algorithm's performance. An indicative example could concern the case where the outcome of the classification process is extracted using invalid inputs. This could be caused by several factors leading to input values potentially not being up to date because of unpredictable transient problematic network conditions. Indicatively, such conditions include network congestion, resulting in increased packet loss and delay, or node mobility leading to network disconnection. Traditional data mining approaches do not consider these problems. Specifically, the

input data are assumed to be always valid (i.e., in time), while the execution delay is assumed negligible due to abundant processing resources.

Driven by such observations, distinguishing WSN from other traditional network areas is essential. A valuable contribution of this work, compared to relative ones, concerns a comprehensive study on the effect of such conditions in the context of realistic WSN environments. Towards this objective this paper also presents a respective framework enabling application of existing data mining algorithms in such cases, accounting for the distributed processing power, the communication cost, and the algorithm's sensitivity to invalid input data.

In this work, we study the proposed fuzzy logic system in an environmental scenario by simulating a realistic WSN infrastructure characterized by significant communication challenges. In our previous work [7] a centralized implementation of a healthcare FIS was presented, where a TelosB mote was considered as a cluster head, being responsible for the reception of all the generated packets and the FIS execution. In [8] the respective distributed implementation was presented. The evaluation results of both efforts proved the sensitivity of the system's performance to the networking conditions and the cluster head's overloading. Driven by these observations the main goal and contribution of this paper concern the proposal of novel and efficient approaches on implementing data mining algorithms specifically targeting WSNs applications as well as undertaking a respective comprehensive performance evaluation. Towards this objective, this work investigates the FIS's performance under three different scenarios: centralized and distributed time triggered as well as centralized event triggered execution. Respective evaluation highlights in a quantifiable way that the centralized approach burdens the CPU utilization of the FCHN, due to the fact that all data flows are directed to it. Driven by these observations as well as taking into consideration the scarce CPU and wireless bandwidth availability in WSNs, we propose a distributed approach to partition the execution of the FIS among the nodes. In this way we aim to optimally balance the energy consumption between the nodes. Moreover, respective measurements indicate a significant diffusion of network traffic avoiding the all-to-one communication scenarios. Moreover, an exploration is conducted on the way wireless channel conditions and the packet arrival rate affect the overall performance of the FIS and the final outcome of the event detection system. Although the distributed time triggered approach is anticipated to balance the energy consumption, the evaluation revealed additional aspect and interdependencies, needing in-depth investigation. Typically, the FIS algorithm is executed periodically based on a time triggered approach. In case that the occurrence of abnormal events is rare, the periodic nature handles CPU resources and network bandwidth in a nonoptimal way. For this reason, our work proposes an event dissemination plan where only when a sensor's value can potentially trigger the identification of an abnormal situation is the algorithm actually executed. In other words, when all sensors values are in "normal" ranges there is no need for computational and communication effort so as

to execute the FIS, in order to evaluate the probability of fire.

To summarize, the contribution of this paper is threefold. Firstly, to the best of our knowledge it is the first attempt for an online execution of a fuzzy logic system considering the limitations and the restrictions of a WSN infrastructure in an environmental application. In that respect a centralized environmental FIS is implemented in the context of a well known WSN platform, while MATLAB based respective algorithms where ported include Mamdani's model [9], fuzzy max aggregation, max implication, and centroid of the graph defuzzification method [10] to C based functions which are able to be executed on a TelosB platform. It is noted that significant optimization was also necessary so as to take into account the processing and memory resources. Secondly, a comprehensive comparative study as well as evaluation effort is presented, concerning the execution of the FIS following a centralized time triggered, distributed time triggered, and centralized event triggered design and implementation approach. Task distribution is based on adequate objective function proposed in this paper taking into consideration the CPU workload of each task and the communication cost that the network topology introduces. Event dissemination strategy considers a multicast packet transmission, which triggers the sensors to transmit their monitoring value to the cluster head so as to perform the FIS. Thirdly a thorough investigation through respective result analysis is undertaken, showing how the network conditions affect the system's performance.

The remainder of this paper is organized as follows. In Section 2 relevant works are presented. In Section 3 an introduction to the data mining domain is presented and the system model is exhibited, while in Section 4 the necessity of a distributed execution of a high processing algorithm (classification algorithm) through task scheduling is analyzed. The proposed event dissemination plan is described in detail in Section 5. The holistic framework for evaluating the aforementioned concept is analyzed in Section 6. The performance analysis of our work is depicted in Section 7. Finally, in Section 9 the conclusions of our work are presented.

2. Related Work

In order to better highlight the correlation and most importantly the enhancements of our work with respect to relative efforts specifically focusing on wireless sensor networks, this section is effectively structured according to the three basic aspects addressed in this paper: the application of FIS in an environmental scenario under a WSN infrastructure, the distributed execution of an algorithm considering the processing and the communication costs typically encountered in WSN systems, and the logical-event driven execution of a data classification algorithm. In all cases the main objective is to optimize the computational effort.

2.1. Fuzzy Logic in WSNs. Firstly, in current literature there are several applications which utilize the fuzzy logic and

targeting in WSN environments. Nevertheless, these efforts usually fail to accurately consider the impact of the wireless medium, encountered in a WSN, on the performance and on the accuracy of the FIS. In contrast, such aspect is critical to the FIS design presented in this paper highlighting the importance of this work to a typical WSN system. Specifically, in [11], a fuzzy rule based system for fire detection is proposed, based on environmental values, while in [12] a fuzzy logic system for controlling home appliances is proposed, so as to create the smart home environment. The same use case scenario is considered in [13], where the authors offer an interesting analysis, targeting reduction of false alarms in fire detection scenarios, under the utilization of the fuzzy logic framework. In addition they proposed the use of spatial and temporal characteristics into the rule base so as to increase the system's accuracy. Additionally, novel techniques are proposed that reduce the exponential growth of the rule base without compromising the accuracy of event detection. As the authors claimed in their manuscript, the main goal of this work is threefold: (a) to recommend fuzzy logic as an accurate technique for detecting various events, (b) to propose new characteristics that will serve as inputs to the fuzzy engine so as to reduce the number of false alarms, and (c) to propose new techniques for reducing the rule set length, so as to allow its storage in the limited sensor's memory. The evaluation analysis was performed utilizing the FuzzyJ Toolkit software, while the authors used the fire events public database from NIST so as to evaluate their system. At this point it is important to note that this software is not a network simulator so no realistic effect has been evaluated with respect to networking constraints such as sensors disconnection, packet loss, and latency. Moreover, an energy estimation on how this algorithm utilizes sensor's resources is also omitted. Such monitions raise concerns regarding the results practicality and accuracy considering a realistic WSN setup. Enhancing and extending such effort, the effect of FIS design on power consumption as well as distributed implementation plays a critical role in our paper offering significant added value upon considering realistic WSN systems. Furthermore, a congestion fuzzy logic estimation model for QoS in wireless sensor network is presented in [14], while the evaluation results confirm that the proposed FIS is able to efficiently sort out traffic and minimize packet loss, even in critical scenarios. However, the fuzzy system is executed online; there is not any specific application scenario under investigation, while the scheduling of the above system to the sensor nodes is not taken into consideration as well as the event triggering of this system. In addition, the authors of [15] present an activity recognition architecture utilizing multinode collaboration and fuzzy logic, in order to produce a reliable recognition result from unreliable sensor data. Moreover, in [16] it is proven that fuzzy logic can be effectively used to enable location estimation in mobile robots, while the authors emphasize the computational requirements reduction and heuristic determination of combination rules. Nevertheless, none of the above papers present evaluation results with respect to required energy for executing the classification system or comparative metrics of the performance and the sensitivity of the fuzzy algorithm, under critical networking

conditions. Such objectives are specifically tackled in this paper so as to enhance practicality and usefulness of such approaches upon considering real WSN deployments. Also, in [17] the authors propose an effective fuzzy based faulty readings detection model in wireless sensor networks. The proposed model is able to detect the faulty readings and overcome the faulty nodes. More specifically, the difference between the sensed values of neighboring nodes was used as a parameter to determine faulty nodes and was represented by a linguistic variable in fuzzy logic. In the proposed framework, after localization error detection, confidence value of each node was obtained, used to detect faulty readings. Also, the fuzzy inference system was used to detect faulty readings in WSN.

2.2. Task Assignment in WSNs. Secondly the problem of the distributed execution of an algorithm over WSNs, also known as task allocation problem, has been investigated. The authors of [18] emphasize the weakness of the direct implementation regarding existing computational intensive algorithms in WSNs due to limited resource availability and shared communication medium. They propose an application-independent task mapping and scheduling solution for multihop topologies in WSNs able to map tasks of an application, focusing on the minimization of the energy consumption and the delay. Moreover, in [19] the authors note that data fusion techniques are important in order to reduce communication traffic. However such methods introduce additional computation effort and highly inefficient distribution of network traffic as well as energy consumption among the nodes. Therefore the authors propose two models, the communication model to describe the network topology as well as the communication data flows and the task model which describes the sensing and processing tasks and the dependencies between these tasks. Another solution is introduced in [20], where the divide-and-conquer technique is utilized in order to partition a set of real time tasks with dependencies and a Markov Decision Process method is used so as to schedule sequential tasks. The evaluation results showed a better response time and reduced network wide energy consumption. Also, in [21] the authors present an energy efficient task assignment and a distributed task migration for WSNs. As the computation tasks are strongly connected with specific communication tasks, the execution of a processing task may be migrated into a neighbor node if the node's remaining energy is less than a threshold. This collaboration between sensor nodes achieves optimization on the energy consumption. Finally, in [22] a task allocation strategy, called DTAC, is proposed. DTAC consists of two steps: task layering and task allocation. Task allocation is implemented by intercluster allocation and intracenter allocation. In order to ensure parallel processing of task, the ranking domain is proposed based on sensor nodes service abilities to classify the sensor nodes into three levels. Sensor nodes with higher ranking domain can be preferentially selected to perform tasks. Based on the service abilities of cluster members, the service abilities of clusters can be calculated. Then, the task source cluster can choose several

collaborative clusters to complete complex tasks according to their respective abilities. Although the aforementioned efforts highlight the problem of task scheduling, there is no realistic use case scenario on how these methodologies can be applied to a specific WSN infrastructure using a well known computational intensive algorithm for splitting the algorithm into parallel and sequential tasks. So they overlook the fact that any proposed segmentation of FIS algorithm should take into consideration the communication and processing dependencies that are created, which are essential for the task distribution. Aiming at addressing such shortcoming this paper takes into consideration realistic WSN system's characteristics related to both communication and processing domains with this offering objective evaluation of respective novel proposals.

2.3. Event Dissemination Protocols in WSNs. Finally the need for efficient data dissemination protocol has also attracted considerable research interest. Bouhafs et al. [23] introduced a new clustering scheme for wireless sensor networks that uses semantic properties as its foundation. They describe a new cluster formation model that considers relevance of nodes to a specific query and organizes nodes within a cluster in a tree structure and showed using simulation based evaluation that their new clustering scheme is more energy efficient and achieves lower energy consumption. Moreover in [24] the authors introduce the notion of logical neighborhood, which replaces the physical neighborhood provided by wireless broadcast with a higher level, application-defined notion of proximity. The span of a logical neighborhood is specified declaratively based on the characteristics of nodes, along with requirements about communication costs. This paper presents the Spidey programming language for defining logical neighborhoods and a routing strategy that efficiently supports the communication enabled by its programming constructs. Finally in [25] a semantic clustering model based on a fuzzy inference system is proposed aiming to identify the semantic neighborhood relationships in wireless sensor networks in order to both reduce energy consumption and improve the data accuracy. As a case study the authors describe a structural health monitoring application which was used to illustrate and assess the proposed model. Both frameworks are not applied in specific application scenarios, while networking, QoS, and real time metrics are not provided so as to strengthen the main contribution of their work.

To the best of our knowledge, this is the first work offering a holistic framework of applying fuzzy logic event detection systems in a specific application scenario, targeting to investigate three basic problems with respect to real WSN systems: (i) implementation of fuzzy logic systems in real sensor platforms, (ii) task scheduling of fuzzy logic systems in more than one node so as to balance the computational as well as the networking load, and (iii) semantic event triggering of the fuzzy system in order to reduce the waste of sensors' resources. The related efforts, analyzed in the current section, examine such objectives partially or myopically. Aiming at addressing such shortcomings, this paper provides a comprehensive analysis and evaluation study of the above three

objectives, so as to give a guideline regarding each scenario's suitability for each particular application. Additionally, the emphasis devoted to WSN systems, in contrast to the rest of the relative efforts, is highlighted by the consideration of TelosB platform characteristics (comprising one of the most widely used WSN platforms) regarding the development as well as the evaluation and results' analysis presented in this paper.

Therefore, a direct numerical comparison between this paper and the relative efforts presented does not offer any useful indications or it is not possible in various cases. However, based on the previous analysis it is highlighted that our effort aims to enhance and extend relative efforts in order to tackle respective identified shortcomings or/and omissions. In this effort the three aforementioned aspects are merged in an environmental application, efficiently executed in the context of a TelosB based WSN, where the probability of fire detection is estimated using a fuzzy logic system. The workload of this fuzzy logic algorithm is split into specific independent tasks that can be executed in parallel in contrast to tasking having dependencies on the outputs of the independent tasks. These subtasks are allocated to different sensor nodes considering the computation load as well as the communication load which depends on the network topology. Finally a fuzzy semantic neighborhood is created and an event based execution of the proposed fuzzy system is presented based on the system's rule engine, executed on WSN platforms, so as to avoid unnecessary algorithm's executions.

3. Data Classification in WSNs

State-of-the-art data classification techniques offer WSNs designers and developers the capability to apply sophisticated methods enabling processing of excessive data volumes (not possible before due to scarce resource availability). Through such mechanisms, it is now possible to achieve significantly enhanced in-depth understanding of system behaviour as well as highly accurate classification of specific events [26]. The increasing utilization of sensor nodes in several applications and the need for acquiring data and adequate knowledge on the respective applications, as well as the need for occurrence confirmation of an ambiguous event, rises the necessity for developing efficient data mining techniques, based on several modalities. The multimodality events are based on the relational sequence of single modality events and require an adequate fusion process so as to derive a safe conclusion, able to trigger the corresponding actuation units. Efficient respective algorithms can offer multifaceted benefits such as higher accuracy and communication minimization between WSN nodes, thus leading to drastic resource conservation, while enabling the reduction of the communication overhead for sensor actor data exchange.

In a typical deployment, regions of interest are constructed and the local sensors values are processed by a classification algorithm, which can be executed by a node or a subset of nodes (centralized/distributed approach). Latter approaches enable the reduction of the direct transmissions

to the actuation units that are responsible for deciding the reactions to the monitored environment depending upon the outcome of the implemented classification algorithm. Considering the centralized architecture, a specific node assumes the role of the central entity receiving or/and maintaining all sensors' samples. The sensor nodes send their values to the aforementioned central entity, which is able to perform the classification algorithm. In this way, the whole processing effort is placed on the cluster head node, which is responsible for deducing the high level information from the data flows, leading to the efficient monitoring of the system. On the distributed approach the main objective is to efficiently distribute the processing workload to a higher number of processing units, aiming to achieve balanced energy consumption as well as packet transmissions, among network nodes. Another critical categorization of classification algorithms concerns the execution approach and specifically whether it is time triggered or event triggered. In the first case the application of the data classification algorithm is periodically repeated. The algorithm's frequency execution depends on several parameters such as the criticality and the nature of the monitored phenomenon or the sensors' readings frequency. In the second case the algorithm's execution is triggered by an abnormal sensor reading. The classification algorithm's output is not deterministically abnormal if a sensor reports a value out of normal ranges. All sensors' data have to be fused in a sophisticated way in order to provide an accurate, qualitative, and quantitative conclusion about the monitored phenomenon. On the event triggered approach the CPU load is reduced as the high computational classification algorithm is executed only when required, reducing the probability of unnecessary executions.

Several data classification algorithms can provide a solution to the problem of event detection: fuzzy logic systems, Bayes Networks, and so forth. From these solutions fuzzy logic is probably the most well fitted algorithmic framework for multisensor scenarios. It deals with uncertainty and inaccuracy and it can be built based on experts' experience. It is also a low computational cost solution compared to other algorithms such as support vector machines. Hence, it is an ideal approach with respect to the nature of our application. The use of fuzzy systems in classification problems has also been highlighted by several works [13, 27]. Contrary to classification algorithms' approach, relying on crisp values to describe WSN events is not suitable, due to the imprecise sensor's values as well as the complexity of determining precise thresholds when more than one parameter is involved in an event detection scheme often leading to potentially conflicting rules.

In this work an environmental application, able to evaluate the probability of a fire detection, is considered as the case scenario while a fuzzy inference system is utilized so as to classify the fire detection. The environmental area is a critical, for many reasons, domain posing respective challenges. This is due to the fact that heterogeneous sensor values are semantically correlated, and thus they have to be merged so as to derive a safe conclusion. There are two main requirements that should be satisfied in case of environmental applications: first of all the detection of abnormal situations and the

adequate emergency response and secondly the reduction of false alarms' rate.

Driven by the aforementioned characteristics, the use of fuzzy logic comprises a prominent solution as opposed to conventional use of crisp threshold approach [13]. In addition, fuzzy logic can efficiently handle fluctuations of sensors' measurements (due to sensors' hardware imprecisions), thus leading to a reduction of the false alarms' rate compared to crisp logic. However accurate output of the fuzzy logic system requires that the respective set of the input values are always in time and their timestamp deviation to the FIS time of execution is minimum. Such prerequisites, however, cannot be guaranteed in the case of WSNs, as the delay and the packet loss ratio (and in general networking related phenomena) can cause problems to the correct execution of the fuzzy system. Hence, taking into consideration the vital importance of environmental related applications, the necessity for a comprehensive study of the FIS under several networking conditions arises. In addition, techniques that can reduce the effect of the networking quality on the algorithm's performance (i.e., centralized versus distributed approach) should be investigated. The error prone nature of the wireless communication is a critical constraint for all highly sensitive application scenarios. To conclude both the nature of the application (requiring advanced methods of data management, i.e., fuzzy logic system) and the nature of the environment (i.e., wireless sensor networks) pose numerous challenging issues.

3.1. Overview of Fuzzy Logic. In fuzzy logic, numerical values are replaced by linguistic variables. According to Zadeh [28] linguistic variables are "variables whose values are not numbers but words or sentences in a natural or artificial language." The structure of a fuzzy inference system consists of the following four modules: fuzzification unit, fuzzy rule engine, inference engine, and defuzzification unit.

The *fuzzification unit* converts a crisp value into degrees of membership by applying the corresponding membership function. A membership function determines the certainty [0-1] by which a crisp value is associated with a specific linguistic value.

The *fuzzy rule engine* consists of a set of linguistic statements, called rules. These rules are in the form *IF antecedent, THEN consequent* where the antecedent is composed of fuzzy input variables connected by logical functions (e.g., AND, OR, and NOT) and the consequent is a fuzzy output variable. Each set of the inputs should activate at least one rule.

The *inference engine* applies reasoning to compute fuzzy outputs. For every rule that is activated its weight is calculated based on the logical operands (Mamdani model [9]). We categorize the rules based on the linguistic variable of the consequent. Applying the implication method, we use the rules' weights so as to define the overall weight [0-1] of each output's linguistic variable. Using the shape and the weight of each output's linguistic variables a set of fuzzy outputs can be calculated using the aggregation method.

The *defuzzification unit* is the transformation of this set of percentages into a single crisp value. The most common

method for extracting the crisp output is the center of gravity method [9].

3.2. Design of a Fuzzy System. Several methods have been used for acquiring knowledge for fuzzy rule based systems. In general knowledge can be achieved from human experts or from experimental data. Mainly there are three different approaches:

- (i) Indirect knowledge acquisition: when someone designs a fuzzy system required knowledge is extracted from interacting with an expert of experienced operator.
- (ii) Direct knowledge: the designer is also an expert so he can formulate his knowledge so that it reflects the system correctly.
- (iii) Automatic approach: statistical modelling approaches from the domain of machine learning are used so as to extract knowledge from already acquired data.

FIS can be entirely extracted from data. This is known as data modelling and many research papers address this problem. However, this approach cannot be applied to cases where the data are not sufficient. In addition the output of this analysis may not be representative of real life scenarios, where the inputs should be modelled according to a specified membership function, or the thresholds are predefined, or some extracted rules cannot represent logical real phenomena. Data samples cannot be assumed to cover all the possible situations, especially in the case of complex systems. The quality of the extracted rules (based on statistical approaches) highly depends on the training set characteristics. Additionally, higher number of representative cases of the phenomenon lead to more generalized induced rules. In these cases, experts knowledge should be formalized in such a way so as to complement the extracted FIS, which has been derived from statistical data. Experts know the main trends of input variables in the fuzzy system and are able to describe their behaviour using linguistic rules. Expert rules are based on extended experience and present a high level of generalization. A domain expert is able to provide us with a global view of the system behaviour, describing the most influential variables and using them in a few basic rules. Dealing with complex systems, experts face two difficulties: how to define useful variables and how to formalize variable interactions. Cooperation between knowledge and data is possible at various levels, such as in designing a model as well as in using it for prediction or decision support. Expert knowledge and data are likely to cooperate in each step of the FIS design process, especially in input and output partitioning and rule definition. Given the input/output partitioning, the rule base may include induced as well as experts' rules. As an example, expert knowledge may serve to define linguistic terms or rules in areas where no data are available. Either expert rule and model validation with data (FETD) or rule automatic generation and expert assessment of induced knowledge (FDTE) is possible [29]:

- (i) FETD: build an expert knowledge base (KB), and then complete it using the knowledge extracted from

experimental data. Data are used to define the precise meaning, in the numerical space, of an expert linguistic concept.

- (ii) FDTE: experts give a linguistic meaning to rules induced from data, thanks to system interpretability. Build an induced KB, and then look for an expert able to evaluate and refine it.

Specifically fuzzy partition and rule design are utilized where knowledge can complement automatic design, whereas member functions and rules are added in areas where no data are available, FIS parameter optimization, and system validation. In this last step, it is proposed not only to check numerical accuracy but also to analyze induced knowledge. The system's generalization ability can be studied through automatic cross-validation procedures and also analyzed by studying the areas of low performance and the links between data items and inference rules. As the use of the fuzzy formalism increases system complexity it has to be justified. FIS based cooperation between expert knowledge and data is suitable when there exists prior knowledge. Expertise often expresses trends in purely linguistic terms, which have to be completed by data to tune the models. The interpretability constraints restrict this approach to low dimensional problems. In our case the linguistic variables were taken from [11] and the total amount of rules was extracted manually. The outcome of each rule was induced following logic and asking experts opinion. The extracted system was then validated using MATLAB tools so as to test its validity in specific input patterns. It is noted that, on one hand, attention was paid so that the set of the input patterns considered is adequately extensive so as to objectively and clearly expose critical characteristics on the proposed design approaches concerning the task allocation and FIS execution strategies. On the other hand, the induced outcome main objective was to serve as a reference point so as to comparatively analyze the performance and accuracy of proposed approaches.

Much research effort has been exerted so as to simplify the rule base using logic minimization, that is, all those approaches which tend to remove logical redundancies as much as possible. Such approaches try to simplify the Boolean/multivalued relation associated with a fuzzy rule based system, treating it exactly like a Boolean/multivalued function to be minimized, as it happens in the synthesis of VLSI circuits. The logic minimization can be applied if the following three conditions are true: the rules have the same consequents, some conditions in the antecedents are common to all the rules, and the union of the other conditions in the antecedents coincides with the overall term set of some input linguistic variables. Hence the input fuzzy variables can be considered as input to the synthesis of a Karnaugh map, and the simplifications can lead to a reduction in the number of rules. For example, assume the following rules:

if temperature is medium and luminosity is high and carbon monoxide is high and humidity is low then fire is high;

if temperature is medium and luminosity is high and carbon monoxide is high and humidity is medium then fire is high;

if temperature is high and luminosity is high and carbon monoxide is high and humidity is low then fire is high;

if temperature is high and luminosity is high and carbon monoxide is high and humidity is medium then fire is high.

These can be reduced to the following rule:

if temperature \geq medium and luminosity is high and carbon monoxide is high and humidity \leq medium then fire is high.

In addition the problem of rule dimension explosion has been discussed in literature and other techniques have been proposed such as fuzzy curves, genetic algorithms, and interpolation techniques.

Incomplete rule base [13] is also a possible solution for decreasing the number of rules. A rule base is considered complete if there are rules for every possible combination of the input variables. However, only some of these combinations have outcomes that are important to the event detection system. For example, rules containing variables which do not satisfy a logical pattern can be removed. In addition, if we exclude the rules with consequents that are of no interest to the event detection system, such as rules indicating that the possibility that a fire has occurred is low, we reduce the size of the rule base even more. As a result, by lowering the level of completeness of the rule base, we significantly decrease the number of rules that need to be stored on the sensor nodes. This trimming process, however, should be performed very carefully in order to prevent the removal of important consequents.

3.3. Proposed Fire Detection Fuzzy Inference System. Based on the application scenario considered in [11] the FIS input set of values include temperature, light intensity, carbon monoxide, and humidity. The linguistic variables of the input values are considered as low, medium, and high. The number as well as the limits of the membership functions can be adjusted. However, in this work the membership functions are selected according to the previous work, cited above. Applying these characteristics to fuzzy logic the resulting proposed FIS includes the following set of input fuzzy variables:

$$\begin{aligned} \text{Temperature } T &\in \{\text{Low, Medium, High}\}, \\ \text{Light Intensity } L &\in \{\text{Low, Medium, High}\}, \\ \text{Carbon Monoxide CM} &\in \{\text{Low, Medium, High}\}, \\ \text{Humidity } H &\in \{\text{Low, Medium, High}\}, \end{aligned} \quad (1)$$

and the probability of fire (FP) is the resulting output

$$\text{FP} \in \{\text{Very Low, Low, Medium, High, Very High}\}. \quad (2)$$

TABLE 1: Membership functions of the fuzzy inputs.

Membership functions with fiducial points	
Temperature	Low: trapezoidal (0, 0, 20, 30)
	Medium: triangular (20, 40, 60)
	High: trapezoidal (50, 60, 100, 100)
Light intensity	Low: trapezoidal (0, 0, 300, 500)
	Medium: triangular (400, 600, 800)
	High: trapezoidal (700, 900, 1000, 1000)
Carbon monoxide	Low: trapezoidal (10, 10, 25, 40)
	Medium: triangular (30, 50, 70)
	High: trapezoidal (50, 60, 100, 100)
Humidity	Low: trapezoidal (0, 0, 10, 30)
	Medium: triangular (20, 40, 60)
	High: trapezoidal (50, 70, 100, 100)
Fire probability	Very low: trapezoidal (0, 0, 15, 20)
	Low: triangular (15, 25, 40)
	Medium: triangular (37, 45, 63)
	High: triangular (60, 80, 100)
	Very high: trapezoidal (80, 90, 100, 100)

The input variables are collected by the related sensor nodes and are used as input to the FIS. Mamdani's model is considered for this fuzzy system. For each crisp value, the degree by which they belong to a predefined fuzzy set is determined via membership functions. The membership functions are defined on each input variable as trapezoidal and triangular functions [11] and are listed in Table 1. As indicative example the membership functions of the temperature along with the linguistic variables are illustrated in Figure 1(a).

The core of the FIS is the rule system which leads the inference system to produce the fuzzy outputs. Using the number of linguistic variables for each input the total number of rules is defined as 81 ($3 \times 3 \times 3 \times 3$). A complete set of the fundamental rules has been extracted and defined so as to train the proposed FIS (Tables 2–6). The following set of rules represent all the possible combinations of the input linguistic variables. The outcome has been defined according to experts. A MATLAB tool (genfis) has also been used so as to evaluate and verify the validity of the described FIS before applying it to the TelosB motes.

Following all environmental inputs' fuzzification, the minimum implication method is applied to each rule, taking the fuzzified inputs and applying them to the antecedent of the fuzzy rule in order to derive its weight. The outputs of all rules are aggregated using the maximum method. The effect of this union on all of the scaled membership functions of the system's output is an area under a curve, that is, Figure 1(b). The aggregated output fuzzy set is defuzzified using the centroid of gravity method in this area and the output is a crisp number. The aforementioned procedure is presented in Algorithm 1. It is worth noting that considerable effort has been devoted to the optimization of critical loops and optimization of floating points calculations, in order to efficiently execute the FIS on a TelosB mote within acceptable

time limits. In the case of the *for* loops optimization, for the calculation of the centroid of gravity, which is the final outcome, the step parameter in the *for* loops has been selected as 0.1. In general, smaller step value leads to higher accuracy as well as to higher number of loops. However, through experimentation conducted in this work it was observed that accuracy deviation between lower step value (0.01) and the chosen one (0.1) can be considered negligible. Therefore, the selection of the step value is a trade-off between the algorithm's accuracy and the number of loops. In the case of floating point calculations, which are necessary in order to evaluate the weight of each rule, precalculated arrays of the membership functions have been used and accessed when it was necessary, reducing the computational effort. The aforementioned functions of fuzzy logic (aggregation, implication, and centroid of graph) can be used to every FIS (e.g., environmental or industrial applications).

Algorithm 1 can be executed in a specifically selected node which is called Fuzzy Cluster Head Node (FCHN). The temperature, light, carbon monoxide, and humidity sensor node are responsible for acquiring the respective data values. In the centralized approach, these values are transmitted to the FCHN, which also serves as the fusion cell, which performs the FIS algorithm. However if we assume that a realistic size of a sensor network consists of tens of nodes, restrictive limitations are imposed. Networking limitations occur to the concurrent data flows, due to traffic bursts and concurrent data flows that must be effectively managed in a network of limited availability bandwidth of just 250 kbps offered by the IEEE 802.15.4 physical layer. Computation limitations stem from the fact that highly complex operation and calculations must be made by low-end processing units based on 8- or 16-bit architectures and frequency clock of a few tens of Hz typically found on TelosB based platforms. Finally, all software stack (including operating system, network protocols, and FIS) must fit in extremely limited memory structures of few tens of kbytes (10 Kbytes to be exact in the case of TelosB). The number of messages sent to the cluster head as well as the processing load of the data mining tasks demands increased processing capabilities and increases the network congestion level as well. Hence a decentralized application of a data mining algorithm is needed, in order to achieve more efficient operation and minimize network performance degradation, as will be analyzed in the following section. Finally algorithm's execution strategy (time triggered versus event triggered) is also a major issue that should be investigated, so as to minimize computational as well as communicational effort.

4. Efficient Task Allocation among WSN Nodes

4.1. Problem Statement. A critical challenge in the proposed FIS event detection algorithm concerns the process which distributes computationally intensive tasks to several processing elements (sensor nodes) aiming at efficient utilization of sensor resources. In this context it is important to maximize network-wise resource utilization, while minimizing the communication (data flows) between the processing elements. Another reason of paramount importance advocating

TABLE 2: Rule engine.

Inputs IN	Rules																	
	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14	R15	R16	R17	R18
Temperature low	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
Temperature medium	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
Temperature high	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
Light intensity low	√	√	√	√	√	√	√	√	√	x	x	x	x	x	x	x	x	x
Light Intensity medium	x	x	x	x	x	x	x	x	x	√	√	√	√	√	√	√	√	√
Light intensity high	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
Carbon monoxide low	√	√	√	x	x	x	x	x	x	√	√	√	x	x	x	x	x	x
Carbon monoxide medium	x	x	x	√	√	√	x	x	x	x	x	x	√	√	√	x	x	x
Carbon monoxide high	x	x	x	x	x	x	√	√	√	x	x	x	x	x	x	√	√	√
Humidity low	√	x	x	√	x	x	√	x	x	√	x	x	√	x	x	√	x	x
Humidity medium	x	√	x	x	√	x	x	√	x	x	√	x	x	√	x	x	√	x
Humidity high	x	x	√	x	x	√	x	x	√	x	x	√	x	x	√	x	x	√
<i>Fire probability</i>	<i>L</i>	<i>L</i>	<i>M</i>	<i>L</i>	<i>L</i>	<i>L</i>	<i>VL</i>	<i>L</i>	<i>L</i>	<i>M</i>	<i>M</i>	<i>M</i>	<i>L</i>	<i>L</i>	<i>L</i>	<i>L</i>	<i>L</i>	<i>L</i>

TABLE 3: Rule engine.

Inputs IN	Rules																	
	R19	R20	R21	R22	R23	R24	R25	R26	R27	R28	R29	R30	R31	R32	R33	R34	R35	R36
Temperature low	√	√	√	√	√	√	√	√	√	x	x	x	x	x	x	x	x	x
Temperature medium	x	x	x	x	x	x	x	x	x	√	√	√	√	√	√	√	√	√
Temperature high	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
Light intensity low	√	√	√	√	√	√	√	√	√	x	x	x	x	x	x	x	x	x
Light intensity medium	x	x	x	x	x	x	x	x	x	√	√	√	√	√	√	√	√	√
Light intensity high	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
Carbon monoxide low	√	√	√	x	x	x	x	x	x	√	√	√	x	x	x	x	x	x
Carbon monoxide medium	x	x	x	√	√	√	x	x	x	x	x	x	√	√	√	x	x	x
Carbon monoxide high	x	x	x	x	x	x	√	√	√	x	x	x	x	x	x	√	√	√
Humidity low	√	x	x	√	x	x	√	x	x	√	x	x	√	x	x	√	x	x
Humidity medium	x	√	x	x	√	x	x	√	x	x	√	x	x	√	x	x	√	x
Humidity high	x	x	√	x	x	√	x	x	√	x	x	√	x	x	√	x	x	√
<i>Fire probability</i>	<i>M</i>	<i>M</i>	<i>M</i>	<i>L</i>	<i>L</i>	<i>L</i>	<i>L</i>	<i>L</i>	<i>L</i>	<i>M</i>	<i>M</i>	<i>M</i>	<i>M</i>	<i>M</i>	<i>M</i>	<i>L</i>	<i>L</i>	<i>L</i>

TABLE 4: Rule engine.

Inputs IN	Rules																	
	R37	R38	R39	R40	R41	R42	R43	R44	R45	R46	R47	R48	R49	R50	R51	R52	R53	R54
Temperature low	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
Temperature medium	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
Temperature high	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
Light intensity low	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
Light intensity medium	√	√	√	√	√	√	√	√	√	x	x	x	x	x	x	x	x	x
Light intensity high	x	x	x	x	x	x	x	x	x	√	√	√	√	√	√	√	√	√
Carbon monoxide low	√	√	√	x	x	x	x	x	x	√	√	√	x	x	x	x	x	x
Carbon monoxide medium	x	x	x	√	√	√	x	x	x	x	x	x	√	√	√	x	x	x
Carbon monoxide high	x	x	x	x	x	x	√	√	√	x	x	x	x	x	x	√	√	√
Humidity low	√	x	x	√	x	x	√	x	x	√	x	x	√	x	x	√	x	x
Humidity medium	x	√	x	x	√	x	x	√	x	x	√	x	x	√	x	x	√	x
Humidity high	x	x	√	x	x	√	x	x	√	x	x	√	x	x	√	x	x	√
<i>Fire probability</i>	<i>M</i>	<i>H</i>	<i>H</i>	<i>H</i>	<i>M</i>	<i>M</i>	<i>M</i>	<i>L</i>	<i>L</i>	<i>L</i>								

TABLE 5: Rule engine.

Inputs	Rules																		
	IN	R55	R56	R57	R58	R59	R60	R61	R62	R63	R64	R65	R66	R67	R68	R69	R70	R71	R72
Temperature low	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
Temperature medium	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
Temperature high	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
Light intensity low	√	√	√	√	√	√	√	√	√	√	x	x	x	x	x	x	x	x	x
Light intensity medium	x	x	x	x	x	x	x	x	x	√	√	√	√	√	√	√	√	√	√
Light intensity high	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
Carbon monoxide low	√	√	√	x	x	x	x	x	x	√	√	√	x	x	x	x	x	x	x
Carbon monoxide medium	x	x	x	√	√	√	x	x	x	x	x	x	√	√	√	x	x	x	x
Carbon monoxide high	x	x	x	x	x	x	√	√	√	x	x	x	x	x	x	√	√	√	√
Humidity low	√	x	x	√	x	x	√	x	x	√	x	x	√	x	x	√	x	x	x
Humidity medium	x	√	x	x	√	x	x	√	x	x	√	x	x	√	x	x	√	x	x
Humidity high	x	x	√	x	x	√	x	x	√	x	x	√	x	x	√	x	x	√	√
Fire probability	M	M	M	M	M	M	L	L	L	H	H	H	H	M	M	M	M	M	M

TABLE 6: Rule engine.

Inputs	Rules									
	IN	R73	R74	R75	R76	R77	R78	R79	R80	R81
Temperature low	x	x	x	x	x	x	x	x	x	x
Temperature medium	x	x	x	x	x	x	x	x	x	x
Temperature high	√	√	√	√	√	√	√	√	√	√
Light intensity low	x	x	x	x	x	x	x	x	x	x
Light intensity medium	x	x	x	x	x	x	x	x	x	x
Light intensity high	√	√	√	√	√	√	√	√	√	√
Carbon monoxide low	√	√	√	x	x	x	x	x	x	x
Carbon monoxide medium	x	x	x	√	√	√	√	x	x	x
Carbon monoxide high	x	x	x	x	x	x	x	√	√	√
Humidity low	√	x	x	√	√	x	x	√	x	x
Humidity medium	x	√	x	x	√	x	x	√	√	x
Humidity high	x	x	√	x	x	√	√	x	x	√
Fire probability	H	H	VH	H	H	H	H	H	H	H

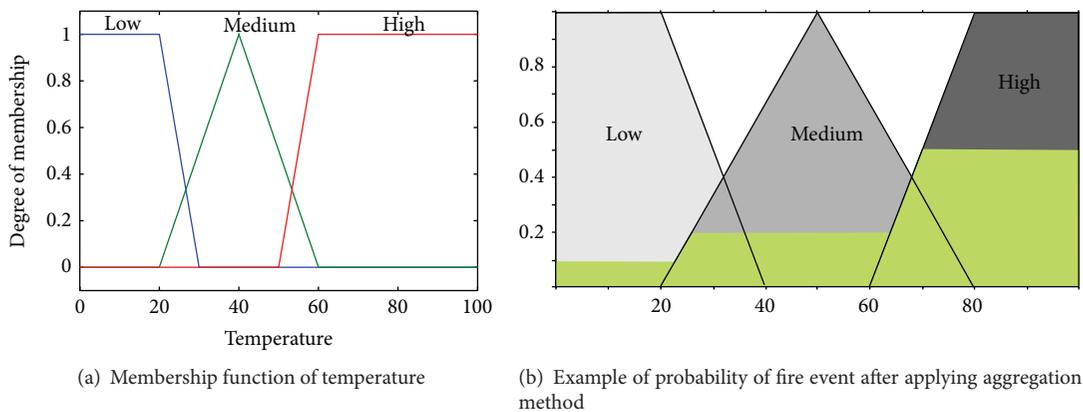


FIGURE 1: Fuzzy logic.

```

Input: Tmpt, Hum, CM, L,
TmptMembFunc, HumMembFunc, CMMembFunc,
LMembFunc, RuleEngine
Output: Fire Probability
FTemp = Fuzzify(Tmpt) // Temperature Fuzzification
FHum = Fuzzify(Hum) // Humidity Fuzzification
FCM = Fuzzify(CM) // Carbon Monoxide Fuzzification
FLI = Fuzzify(Light) // Light Fuzzification
for (i in Len(RuleEngine)) do
    weightrule[i] = findruleweight(RuleEngine(i), FTemp, FHum,
    FCM, FLI)
end
for (i in Len(RuleEngine)) do
    if weightrule[i] == verylow then
        if weightrule[i] > FPverylow then
            FPverylow = weightrule[i]
        end
    else if weightrule[i] == low then
        if weightrule[i] > FPlow then
            FPlow = weightrule[i]
        end
    else if weightrule[i] == medium then
        if weightrule[i] > FPmedium then
            FPmedium = weightrule[i]
        end
    else if weightrule[i] == high then
        if weightrule[i] > FPhigh then
            FPhigh = weightrule[i]
        end
    else
        if weightrule[i] > FPveryhigh then
            FPveryhigh = weightrule[i]
        end
    end
end
area = aggregation(FireProbabiltyMembFunc, FPlow, FPmedium, FPhigh, FPveryhigh)
FireProbability = centroidofgraph(area) // defuzzification

```

implication

ALGORITHM 1: Pseudocode of the fuzzy process.

such approaches is the fact that existing highly demanding processing algorithms cannot be implemented in isolated WSN nodes, due to the limited availability of resources and the shared communication medium. For instance, data mining processing involves highly demanding operations that a single sensor node may not be able to undertake or it may quickly become energy exhausted.

Cluster based schemes have been frequently proposed in current literature, where high level processing tasks are

assigned to cluster heads, without considering the capacity of cluster heads computation power and the concurrent data flows to the cluster head, which will increase the number of collisions. Cooperation among sensor nodes through distributed and parallel mechanisms emerges as a promising solution offering high processing power in resource restricted WSNs. The distributed execution of these algorithms requires task allocation schemes optimally allocating computation tasks to specific sensor nodes, providing multiple benefits as

follows. Contrary to centralized architectures where raw data are transferred and processed to a specific, usually resource-rich, station (i.e., cluster head, base station), task assignment schemes critical objective is to utilize the sensors computing capabilities mainly for local processing (also known as on-site processing). The on-site processing of the data significantly reduces the need for packet transmission and shortens the communication pathways in multihop WSNs. Consequently the application is benefited since the communication overhead and communication delay are reduced, which effectively results into significant enhancement of the network robustness and throughput capabilities of the network. Furthermore, the overall energy consumption regarding processing tasks is better distributed among the sensor nodes. In addition, a distributed implementation of such algorithms allows the possible parallelization of specific tasks' execution offering drastic improvement of the overall algorithm's delay as well as meeting communication time constraints. The application's deadlines are also benefited by the reduced data volume, leading to reduced data communication and delivery latency.

The segmentation of an intensive algorithm to several subtasks is highly challenging due to the dependencies between computation tasks. These execution dependencies are also correlated with communication dependencies, since these tasks are allocated on different sensor nodes. We should also mention that the network traffic is strongly affected by such task parallelization approach, since the traffic flows are not directed to the base station as is typically the case in the centralized approach.

Consequently the distribution of processing tasks among the sensor nodes must be carefully configured, in order to address the aforementioned challenges. Specifically, the parallelization of the main process and the assignment of the tasks can be summarized as follows:

- (i) Calculation of the total processing cost required by the application.
- (ii) Identification of the tasks that can be parallelized as well as of the dependencies among the tasks, in terms of data dependencies, time of delivery to the application, and network traffic.
- (iii) Identification of the respective sensor nodes that will be responsible for the execution of each task.
- (iv) Synchronization among the parallelized tasks and the final application that they will feed with their outcome.

The goal of the task scheduling in WSNs is, on one hand, to balance the energy consumption among the nodes that are related to an algorithm, without increasing the network's total energy consumption, and, on the other hand, to reduce the number of transmissions so as to relax the network traffic. Here we describe the required energy for executing a computational task. The energy required for executing a

specific computational task (t) in a node (n) is given by the following equation:

$$E_{\text{comp}}(t, n) = \frac{N(t)}{F(n)} * I(n) * V(n), \quad (3)$$

where N is the number of cycles for this task, F is the node's CPU frequency, and I is the active CPU current and V is the voltage supply.

At the same time the energy which is needed for the communication task between two nodes which are in the same radio range is defined as follows:

$$E_{\text{tx}}(n_1, n_2) = E_{\text{elec}} * k + \epsilon_{\text{ampl}} * k * d^2(n_1, n_2), \quad (4)$$

where E_{tx} is the transmission energy, E_{elec} is the energy per bit that is consumed on the transmitter's circuitry, ϵ_{ampl} is the energy that is consumed at the transmitter amplifier, k is the packet's data length, and d is the Euclidean distance between nodes n_1 and n_2 .

The energy for receiving a packet is

$$E_{\text{rx}}(n_1, n_2) = E_{\text{elec}} * k. \quad (5)$$

For multihop scenarios with j hops, the energy for transmitting a packet between the nodes n_a and n_b is

$$E_{\text{tx}}(n_a, n_b) = \sum_{j=a:b-1} (E_{\text{elec}} * k + \epsilon_{\text{ampl}} * k * d_{j,j+1}^2(n_j, n_{j+1})). \quad (6)$$

Hence for each node the total energy consumed is defined as the sum of computational energy and communication energy consumption. More specifically each node has a task procedure including data acquisition and potentially processing functionality. The output of this task should be delivered to another node based on the algorithm's dependencies. Regarding the network topology this output may need to be transferred over one or more hops to reach the final destination. Regardless of the exact number of relays, crucial for the network's total energy and for the delay that the output is available at the desirable node, the transmitter node has to spend energy for 1-hop communication (see (4)). We categorize the role of each node, as follows: *pure transmitter*: (sense)-transmit, *raw transceiver*: receive-(sense)-transmit, *high processing transmitter*: (sense)-process-transmit, *high processing transceiver*: receive-(sense)-process-transmit, or *high processing receiver*: receive, processing. Consequently the total energy that each node has to spend for the communication and computational part is the sum of (3) and (4). The network's total energy is extracted considering the energy of each node.

The application of this analysis to the proposed FIS is highlighted in the following subsection.

4.2. Proposed FIS Task Distribution Approach. Driven by the aforementioned analysis in this section, we try to distribute the execution of the proposed FIS among the fuzzy related nodes, with respect to the CPU load of each FIS's specific tasks

and the communication cost among the sensor nodes so as to perform the FIS.

The main objective of the proposed distributed approach is to assign tasks to nodes based on criteria such as data dependencies and nodes' sensing capabilities and locality and energy resources. Hence, for the considered application scenario of fire detection, the sensors' values acquisition can be considered independent of the respective fuzzification processes and thus being executed locally by the sensing WSN nodes. In this way, we accomplish the on-site processing of data. The next step of our application requires the evaluation of the coefficient rules. This task depends on the fuzzified inputs.

Hence it was evaluated that it is more efficient to relocate the task of evaluating the coefficient rules closer to the sensing nodes and particularly to one of them. In this case this node does not need to transmit its own data, leading to packet transmission reduction. The node that was assigned the additional task of evaluating the coefficient of each rule was selected based on the average distance among the others and its energy resources. The other two remaining tasks, the implication and the aggregation method, are the most computationally intensive. However the aggregation method needs as input the weight of each rule. In our case this means that 81 floating point numbers, one for each rule, need to be transmitted. If we merge these values in one packet, the packet size is excessively increased and this affects the duration that a node occupies the medium.

Moreover if these two tasks are not assigned in different nodes, the main percentage of the processing workload will burden the cluster head node, thus invalidating the main objective of the approach. For this reason we assume that the node that is calculating the rule of each weight should also apply the respective implication method. In this way, only the output of this method, which in our case is comprised by four floating point numbers, is going to be transmitted to the cluster head, which is going to apply the aggregation method, as well as the defuzzification of the fuzzy outputs, by finding the center of the gravity. A centralized as well as distributed topology of our network is illustrated in Figure 2. In the centralized approach (Figure 2) the four nodes are responsible only for sensing and transmitting the acquired values to the base station. Therefore, the base station will take over the reception of the sensing data and their processing as described in Section 3.1. In the distributed approach (Figure 2), as we have already mentioned, one node from the FSNs is responsible for evaluating the rules' weights and for applying the implication method. This node is called *coef.FSN*. We assume that for a given topology the temperature sensor is closer to the other fuzzy nodes. Hence, for this topology the temperature sensor is the *coef.FSN*. The remaining FIS related nodes sense their values and fuzzify them according to the membership functions, while the FCHN takes as input the five weights (*very_low*, *low*, *medium*, *high*, and *very_high*), so as to extract the final outcome.

Considering the task assignment that is depicted in Figure 2, a comprehensive evaluation is performed concerning the total energy distribution among nodes as it will be presented in the following section, indicating significant

benefits offered by the proposed approach with respect to the paper's objectives. We assume 1-hop network topology which consists of 5 nodes (i.e., four FSNs and one FCHN). The CPU energy that is consumed for each task is calculated according to the following process.

The application code is written in C, as a module for the Contiki OS [30]. It is then compiled into a firmware targeting specifically the TelosB WSN platform [31]. At the next step we isolated all the tasks, related to our application and based on the profile data extracted by the MSPSIM (an instruction level emulator of the TI MSP430 MCU) [32]; we calculated the average clocks per instruction (CPI) that the MCU requires to perform each task. Additionally, in order to apply (3) it is assumed that the TelosB motes emulated are equipped with the TI MSP430 [33] microcontroller unit (MCU), clocked at 8 MHz with 108 kB RAM; the active current is 2.33 mA, while the voltage supply is 3.3 V [34]. Following this approach, Table 7 presents the calculated processing load, in terms of CPI, for each task of our proposed FIS. Furthermore, the average CPU energy for generating a packet value formulating it to a packet and copying it to the output buffer using the energy timers was offered by the Cooja simulator [35]. The value of the consumed energy for this part of the process was calculated at 0.107 mJ. In addition, we used the energy timers for the radio that the Contiki OS provides. We calculated that the average energy for transmitting/receiving a 2-byte packet is 0.0607 mJ/0.0672 mJ, respectively.

In the centralized approach the whole CPU energy consumption due to processing is located at the base station, while, in the distributed approach, this processing load is shared among the 5 nodes. Based on our previous analysis, a well balanced and distributed network traffic as well as consumption is anticipated. Here we are presenting an indicative numerical example on the energy distribution among the 5 nodes. Specifically the centralized and the distributed approaches are considered using Table 7 and the already calculated energy for acquiring a value and sending a packet and receiving a packet and each fuzzification related task. The average energy of Table 7 is calculated according to (3). As the generation rate is a crucial parameter, we are going to investigate the most stressed and the most relaxed scenario from an energy consumption point of view scenario. In the first case the FIS algorithm is executed every 2 seconds, while the sensing/generation rate ($tf1$) has a value of 0.25 sec. For this configuration, the total energy for one execution of the fuzzy logic algorithm can be extracted as follows: in the time duration of FIS period each one of the FSNs has generated and transmitted 8 packets, which is equal to 1.3416 mJ. The FCHN has received 32 packets from the FSNs and has executed the fuzzy algorithm once, which is equal to 2.806 mJ. Thus, the total energy is 8.1724 mJ, while the CPU energy part is of 4.08 mJ and the remaining 4.0924 mJ is attributed to radio operation. For the same scenario, considering the decentralized approach, the *coef.FSN* receives the fuzzified values from the remaining four FSNs ($tf1 = 0.25$ sec). Then, when the fuzzy timer (2 sec) expires, it fuzzifies its value, it calculates the weight of each rule, and it sends the two weights of the fuzzified alarm/nonalarm output to the FCHN. The aforementioned tasks lead to an

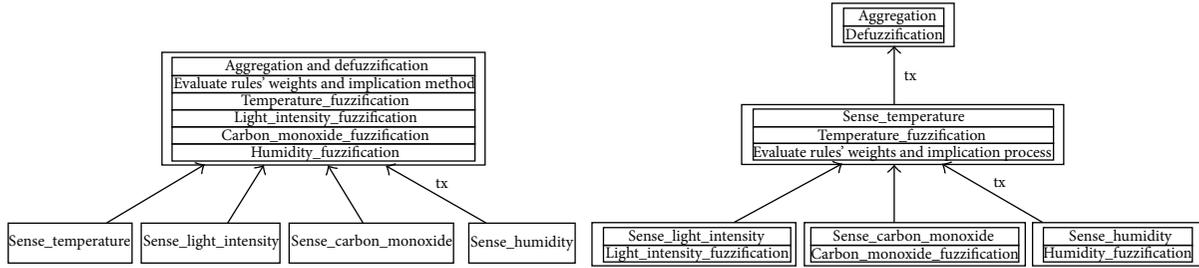


FIGURE 2: Centralized and distributed time triggered approaches.

TABLE 7: CPI for each task of the proposed FIS.

Task	Average CPI	Average energy (mJ)
Temperature_process	1441	0.00138
Light_intensity_process	1443	0.00138
Humidity_process	1440	0.00138
Carbon_monoxide_process	1442	0.00138
Eval. rules' weights & implication	65579	0.063
Aggregation	21806	0.021
Defuzzification	58929	0.566

average CPU energy of 2.6135 mJ. The three remaining FSNs sense, fuzzify, and transmit their value every $tf1$. Based on the energy that is required for the fuzzification of each modality this energy is approximately 1.452 mJ. Finally, the FCHN applies the defuzzification method to calculate the absolute percent of the fire probability (0.6332 mJ). Hence, in the distributed approach the total energy is 7.603 mJ and the radio energy is 3.1975 mJ. At the same time, the absolute number of transmissions is reduced by 21.875%. Moreover, we investigate the energy consumed when $tf1$ has a value of 2 sec. In this case the FSNs transmit when the fuzzy timer expires. In the centralized approach, each of the FSNs spent 0.121 mJ while the FCHN spends 0.924 mJ. For the distributed approach the coef_FSN spends 0.2719 mJ, the remaining FSNs spend 0.1738 mJ, and the FCHN spends 0.5445 mJ. In this case the total energy is approximately the same with the centralized approach; however the overall energy has been balanced among the nodes.

A crucial point is the selection of the most suitable coef_FSN trying to minimize the radio consumption according to (4) and (5). In that respect, xml structure of the utilized topology was extracted and, according to the FSNs position, the total radio energy for performing the coef_FSN role was calculated. The node that satisfied the minimum total radio energy was selected as coef_FSN. A significant added value of this effort is the fact that the proposed two approaches can easily be adjusted to several applications. Additionally, the distributed version of our FIS can be easily extracted for every FIS, if the computational cost, the data flows, and the network topology are known.

5. FIS Execution Strategies

Aiming to further enhance FIS efficiency while minimizing resource wastage it is critical to take into consideration the way FIS itself is executed as well as the behaviour of the sensor nodes correlated to the specific FIS. In that respect semantic correlation among WSN nodes comes into play and in this section an approach is proposed exploiting such correlation in order to optimize each node's functionality. A typical WSN consists of spatially distributed nodes with sensing, processing, and communication capabilities. Several clustering schemes have been proposed in the current literature based on geographical criteria (geographical clusters), in order to reduce communication overhead. However on several realistic scenarios it is of higher significance to focus on conceptual neighboring (expressed by specific semantic correlations) among nodes rather than on the geographical neighboring. For example, in fire detection scenario considering a specific area, values acquired from smoke, temperature, and light sensors are inextricably correlated since these are the values that must be fused so as to accurately indicate the probability of fire. Therefore, it is easily understood that in such case the semantic correlation among these values is much more important to their geographical correlation (if any). Consequently, it is critical, on one hand, to discover semantically correlated neighbors and it is even more critical, on the other hand, to take into consideration the semantic correlation in the design of the functionality of each node.

Furthermore, in contrast to typical WSNs, characterized by a single application of gathering and reporting data, novel demanding application scenarios are composed of many collaborating tasks each affecting only a portion of the system. As an indicative example, a WSN (Wireless Sensor Actor Network) for building control and monitoring can be split into three main tasks: structural monitoring, in-door environment monitoring, and reaction to extreme events such as earthquakes. To provide respective functionality, the nodes that control water sprinklers should use as input data acquired from nearby temperature and smoke sensors and take appropriate actions when and where needed. In such cases, including a central base station in the control loop degrades system performance and reliability without any sensible advantage [24]. Logical neighborhoods can offer a solution to logically partition the system and interact with it.

Taking previous requirements into consideration in this paper an approach is presented, implemented, and evaluated

attempting to exploit semantic correlation among nodes as well as transferring segments of the application logic to the network as follows.

As it is analyzed in Section 3.1 each FSN acquires its sensor's values and it either sends the raw value to the FCHN (centralized approach) or fuzzifies its value and transmits the output to the *coef_FSN*, in specific time periods (distributed approach). However in both approaches unnecessary transmissions may occur in case all FSNs values are in normal ranges and thus no rule is triggered which could lead to a medium or high probability of fire. In other words each node knows when its value can potentially trigger a rule from the fuzzy rule engine, which leads to medium, high, or very high probability of fire. Additionally, in such a case a multicast packet has to be sent to its logical neighborhood (remaining FSNs and FCHN) so as to advertise a possible alarm situation and trigger the FSNs to transmit their current sensed value to the FCHN. Otherwise FSNs do not transmit any value and the FCHN does not perform the FIS.

An indicative example could concern the case where the temperature value is 20 degrees, the humidity is 80 degrees, and the light intensity is 100 degrees, while the carbon monoxide is 15 units. According to the membership functions the linguistic variables of the above values are low, high, low, and low, respectively. As it is indicated in Table 2 the fire probability is very low and hence there is no reason for executing the FIS. It is obvious that each FSN does not have the knowledge of its logical neighborhood so that all nodes belonging to that logical cluster transmit their value when a rule with fire probability medium or higher is triggered. However each FSN knows the normal monitoring ranges that do not cause danger in the monitoring application. Hence if a node exceeds its normal ranges then it is responsible for advertising an alarm situation to its logical neighbors (effectively transmitting a multicast packet). After the reception of this multicast alarm packet, FSNs send their sensing value directly to the FCHN so as to trigger the execution of the FIS. Considering the previous example, let us assume that the temperature node senses a value of 40 degrees. This temperature value is out of the normal ranges. This can be due to weather conditions, due to sensor's failure, or due to a real fire event. However the temperature node is aware of the sensing values inside its logical cluster. In that case the temperature node sends a multicast packet in its logical cluster so as to transmit the sensing values of its logical neighbors values directly to the FCHN. Assuming scenarios where the monitoring environment is out of potential event detection the event triggered execution of the FIS following the proposed approach can lead to a significant reduction of the number of transmissions. In addition in case of potentially dangerous situations the FIS will definitely be executed. Moreover the computational overhead of the FIS algorithm is also reduced, as validated by evaluation effort presented in following sections.

6. Simulation Framework

6.1. Simulation Environment. Aiming to offer an objective evaluation based on state-of-the-art components, the Contiki

OS is selected as a prominent open source operating system for networked wireless devices with limited resources. The simulations were conducted using Cooja, the network simulator provided by Contiki OS. It must be noted that Cooja comprises a valid and widely used emulation environment of Contiki based code offering highly accurate representation of the behaviour and performance evaluation [30, 32]. The application code has been compiled for TelosB sensor platforms which are emulated by the Cooja simulator, while the code optimization that has been discussed in Section 3 targeted the optimal utilization of TelosB's limited resource capabilities. TelosB was developed at UC Berkeley and is widely used in research, although a commercial offspring (Tmote Sky) is used in industry. TelosB is based on 8 MHz 16-bit RISC TI MSP430 processor with 10 kB RAM, 16 kB of configuration EEPROM, and 48 kB flash memory. It offers a digital I/O and I2C and SPI interface in addition to the UART serial communication provided by the FT232BM (FTDI) chip. It draws 1.8 mA in active mode and only 5.1 μ in sleep mode. It is IEEE 802.15.4 compliant having a widely utilized CC2420 radio chip. The communication is compliant with the 2.4 Ghz IEEE 802.15.4 specification used by most dominant WSN platform implementation such as TelosB [31] and Shimmer [36]. The MAC layer protocol considered is the low power CSMA protocol, without duty cycling so as to evaluate the sole effect of the implemented FIS algorithm, which is one of the most common medium access control algorithms. Based on this framework and in order to reach useful conclusions, a comprehensive evaluation is presented with respect to parameterized network characteristics, while considering all required performance metrics.

6.2. Simulation Parameters and Evaluation Metrics

6.2.1. Topology. In all scenarios the network is comprised by twenty-five nodes. In each case one node is considered the cluster head node (Fuzzy Cluster Head Node, FCHN) and four nodes are related to the fuzzy process (Fuzzy Sensor Nodes, FSNs). FSNs values represent temperature, carbon monoxide, humidity, and light intensity, respectively, indicated as *T-FSN*, *CM-FSN*, *H-FSN*, and *L-FSN*. The remaining nodes (20 in total) are the non-FSNs, whose purpose is to simulate concurrent traffic expected in a realistic scenario, where numerous nodes coexist, some of them being irrelevant to the FIS objective and thus effectively interfering with the direct transmissions between FSNs and FCHN. This leads to a low quality, error prone communication channel as anticipated to face in an actual WSN deployment. In the centralized approach the FCHN gathers the data from all the nodes and executes the FIS. In the distributed approach, the node that is responsible for evaluating the rules weights and applying the implication method is called *coef_FSN* and is one of the FSNs. The non-FSNs and the *coef_FSN* send their data to the FCHN, which also derives the final outcome of the algorithm.

6.2.2. Traffic Factor. The traffic factor describes the packet generation rate. The packet generation rate of the 4 FSNs (*tf1*)

is tuned in the range of 0.5–2 sec, while the traffic factor of the remaining nodes ($tf2$) is tuned from 0.25 sec to 1.5 sec. Based on $tf1$, $tf2$, and number of nodes (FSN and non-FSN), different levels of congestion can be configured. As the majority of nodes are non-FSNs the value of $tf2$ has a significant impact upon the aggregated traffic. Thus when $tf2$ has low values the network is quite stressed, while when $tf2$ increases, it leads to relatively relaxed traffic scenarios.

6.2.3. Fuzzy Timer. This parameter represents the execution frequency of the fuzzy algorithm, indicating the time between two consecutive FIS executions. It must be noted that the input data considered are the latest received FSN packets. Therefore, depending on algorithm's frequency and channel's errors, the data utilized may not be up to date, due to network delay or packet loss, highlighting another critical trade-off. However, this parameter may change depending on the criticality of the monitoring event.

6.2.4. Evaluation Metrics. The implemented FIS has proven to accurately detect respective events in [13]. To achieve this level of accuracy, all data (input to FIS) are assumed to be up to date when fuzzification is executed. However this requirement is challenged by the error prone wireless network. Consequently, if the last generated fuzzy inputs have not been received by the FCHN or the coef_FSN when the fuzzy timer expires, then the fuzzification will be executed with respect to previously acquired (in other words old or nonupdated) inputs thus compromising accuracy. Therefore, an evaluation metric of paramount importance is the number of executions of the fuzzy algorithm with nonupdated input data. Such an event is indicated as a nonupdated (NU) fuzzy process. In order to have a comprehensive evaluation the following events are monitored for each simulation:

- (i) The percentage of the nonupdated fuzzy processes.
- (ii) The packet loss ratio.
- (iii) The number of transmissions.
- (iv) The mean response time between an alert generation and the moment the fuzzy execution starts. Each time an alert value is generated by any FSN, the FIS should be triggered. The mean value of the above time intervals is calculated and defined as mean response time. It is noted that this metric effectively highlights the real time capabilities of the proposed FIS since it indicates the time period required from the general alarm (able to trigger a high probability fire event detection) until the algorithm's outcome.
- (v) The ratio of fuzzy executions per alert.
- (vi) The percentage of missed alert generated values. Missed alert is considered an alarm value that the closest fuzzy execution does not take into consideration. This can be due to packet loss or to a generation of a new normal value.

At this point it is important to note that the selection of the particular evaluation environment was made after

careful consideration which indicated Cooja environment as the most appropriate option. As it can be deduced by the number of different parameters and metrics (and as it will be further clarified throughout the evaluation sections) respective evaluation aimed to reveal fine-grained performance aspects at both system level and algorithmic level. Such insight is rather doubtful through the deployment of real system since it would require the use of 25 real sensors difficult to accurately control and monitor. Another disadvantage of real deployment in this case is the inability to accurately reproduce all network conditions which would lead to results and indication of questionable objectivity as well as accuracy. In order to tackle these shortcomings Cooja is ideal, since, besides offering an accurate network simulation environment, it comprises the most accurate emulation environment of the targeted WSN platform, that is, TelosB nodes. This characteristic is of cornerstone importance to the paper's objective, offering the opportunity to tackle all implementation limitations and restrictions of a real implementation. At the same time it allowed us to reproduce all network conditions in high accuracy while logging and monitoring any required performance or/and behavioural aspect of the whole WSN system.

7. Evaluation Results

7.1. Methodology Setup. In this section the results of the experimental evaluation are presented regarding the comparison of the centralized time triggered (CTT), the centralized event triggered (CET), and distributed time triggered (DTT) approach. The topologies of the aforementioned approaches are illustrated in Figure 3, while the state diagrams of the DTT and CET approach are depicted in Figure 4, respectively. In all cases high number of experiments are conducted in a 1-hop topology which consists of 25 nodes. As previously analyzed four of them are the FSNs, and one is the FCHN while the remaining ones are the non-FSNs. The FSNs generation rate is configured via $tf1$, while $tf2$ controls the generation rate of the non-FSNs.

To summarize, the CTT implementation is a straightforward centralized approach where the nodes send their packets periodically ($tf1$) and the FIS is executed also periodically (ft). The frequency of the FIS depends on how often a decision about the status of the monitored environment should be made. Hence the relation between $tf1$ and ft is critical for our analysis. The DTT approach considers the periodic transmission of the FSN's fuzzified values to the coef_FSN. This is one of the FSNs which is responsible for evaluating the coefficient rules and applying the implication methods to them periodically. The coef_FSN sends the output of the implication method to the FCHN, which is responsible for performing the aggregation and the defuzzification method. Thus the distributed approach, according to the previous description, is also time triggered. The state diagram of the DTT approach is presented in Figure 4. Finally the CET represents a centralized implementation, where the FIS is executed on the FCHN, only when the FSNs report a monitoring value which may indicate a possible fire event (Figure 4).

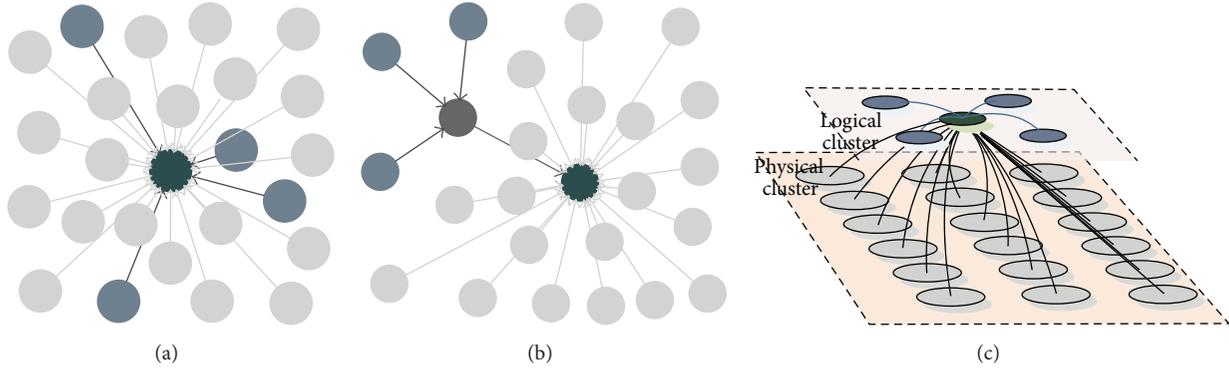


FIGURE 3: CTT, DTT, and CET topology ((a)-(c)).

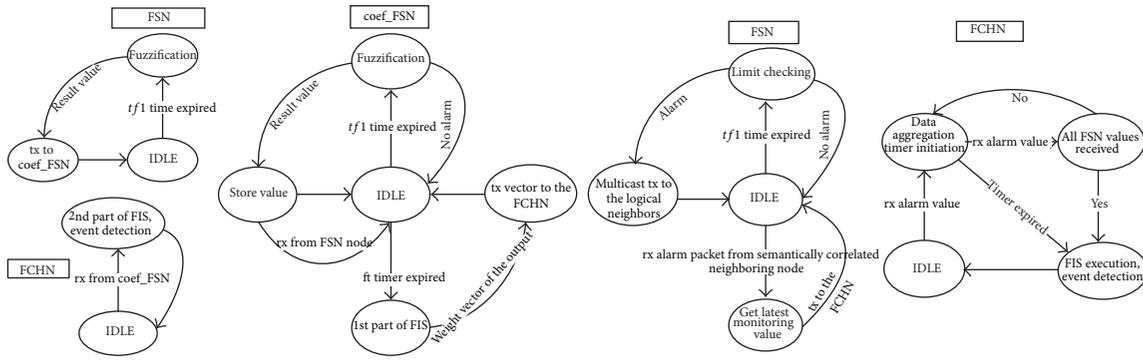


FIGURE 4: DTT and CET state diagrams.

Our evaluation results are analyzed in two subsections. In the first one the CTT is compared with the CET as the first objective of this work is to provide a comprehensive study regarding the FIS execution strategy assuming that the FIS execution will take place on the FCHN. *ft* comprises the cornerstone parameter, as its value affects directly the number of the FIS executions. Hence this analysis mainly focuses on the effect of *ft/tf1* ratio while *tf2* remains constant. The ratio *ft/tf1* is critical as for both stressed and nonstressed scenarios the algorithm's performance is going to be examined for several *ft* period configurations. In the second subsection the three approaches CTT, CET, and DTT are compared, varying *tf2* while keeping *ft* constant. In this way a holistic view is achieved concerning the effect of error prone wireless environment and valuable conclusions are deduced, concerning the resilience of different implementation approaches.

7.2. The Effect of FIS Execution Strategy on Event Detection Algorithm Performance. Our experimental analysis is conducted based on the following assumption: if the input pattern of the FIS does not trigger any rule that has, as a consequence, a medium, high, or very high level of fire probability then FIS should not be executed. In any other case the algorithm must be triggered so as to assess the probability of fire detection (medium, high, and very_high), which effectively depends on the crisp values that each one

of the input parameters has and the consequent output of the fuzzy rule. Thus if the sensors' values are all in normal ranges the fire probability will deterministically be very_low or low. In such scenarios the CTT may lead to excessive CPU and radio resources, as the FSNs transmit their values (without considering their arithmetic value) periodically (*tf1*) to the FCHN, while the FIS is also executed periodically (*ft*). Hence if there is no abnormal value observation in the monitoring environment, the FIS is executed and its output is very_low or low probability of fire detection. On the contrary, the CET is based on the assumption that each time FSN generates a value, it fuzzifies the crisp value and checks whether it triggers any rule with medium or high level of probability. In this case it sends a multicast message to the FSNs and to the FCHN. When the rest of the FSNs receive this alarm message they transmit directly their value to the FCHN in order to perform the FIS based on updated input values. Otherwise no transmission is performed. In this case there is a significant reduction on the number of transmissions as well as on the number of unnecessary fuzzy executions leading to resource waste. It is easily understood that the number of useful FIS executions is effectively a multiparametric problem affected by the frequency of the FIS executions and its relation to *tf1* value as well as the probability of generating values that can trigger any "alarm" rule. Furthermore, in the presented scenarios *tf2* (representing the interpacket time interval for the non-FSNs) assumes two values 1 sec and 2 sec, corresponding to a stressed and

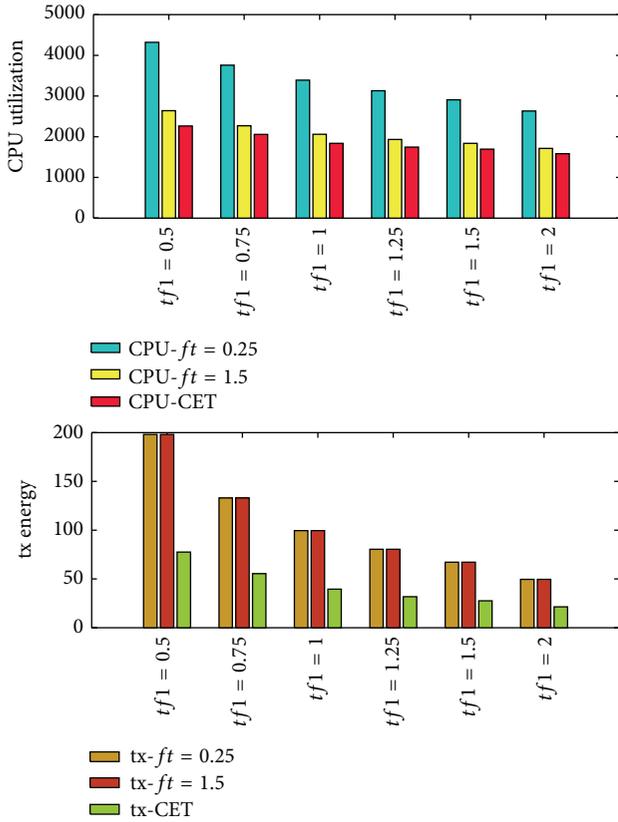


FIGURE 5: tx energy (m) and CPU utilization (mJ) considering CTT and CET.

relaxed case, respectively. Energy consumption is one of the fundamental metrics we focused on to evaluate CTT and CET approaches. The energy consumption gain anticipated by CET implementation is depicted in Figure 5 where the CPU utilization and the transmitted energy are presented. The results of the consumed energy in the CTT approach consider ft value of 0.25 and 1.5 sec, which correspond to extreme values of the fuzzy algorithm frequency execution considered. $tf1$ value varies from 0.5 sec to 2 sec. As it is easily observed CET offers significant benefits regarding overall power consumption. The CPU utilization in the CET is much lower comparing to the CTT ($ft = 0.25$ sec), as, in the latter case, the FIS is executed very frequently and this leads to excessive usage of CPU resources. Specifically, event based implementation leads to 50% reduction of CPU power consumption due to CPU utilization assuming high FSN traffic creation rate ($tf1$ being equal to 0.5 and 0.75 sec) while when the packet creation rate is reduced, respective power consumption reduction of CET compared to CTT ($ft = 0.25$ sec) is approximately 36%. Equally impressive is the power reduction offered by event based FIS concerning radio utilization where 62% reduction is exhibited for $tf1 = 0.5$ sec (Figure 5) which is reduced as traffic rate diminishes reaching down to 50% assuming $ft = 2$ sec corresponding to a drastic power consumption conservation. Overall, in all cases CET is clearly the most power conservative FIS implementation considering both CPU and communication

aspects. Regarding the transmitted energy CTT approaches (tx- ft 0.25, tx- ft 1.5) do not exhibit any deviation as FSNs send their values to the FCHN every $tf1$. On the contrary according to the CET approach the FSNs send their acquired values to the FCHN only when it is needed. Therefore, in case of nonalarm situations unnecessary transmissions are avoided leading to lower consumption due to the radio component.

In the presented event detection scheme unnecessary execution of the fuzzification algorithms comprises one of the most important power wastage factors. Consequently the number of fuzzifications per alert can be used as an objective measure of resource wastage. In this context Figure 6 indicates that event based approach exhibits the minimum unnecessary fuzzification executions which correspond to even less than one execution per alert as recorded. The reason for recording less than one fuzzification per alert is attributed to cases wherein one fuzzification execution corresponds to more than two alerts (i.e., values exceeding respective alert levels). Considering the time triggered approaches respective metric depends solely on the relation between parameters $tf1$ and ft . As depicted, while the frequency of creating packets (and considering a specific alert probability) is increasing compared to the frequency of fuzzy executions (i.e., $ft/tf1$ increasing), it leads to a respective increase of the probability of at least one alert triggering in each fuzzification occurrence. On the contrary when $ft/tf1$ ratio diminishes, it leads to decreased packets creation and thus the probability of an alarm being triggered between two consecutive FIS executions occurring will, respectively, be decreased.

Another critical aspect concerns event detection accuracy which depends greatly on executing the fuzzification algorithms using the latest values of the fuzzy input variables. With respect to such objectives, the error prone nature of wireless communication channels comes into play posing notoriously difficult challenges. Taking this factor into consideration, respectively, highlights a very important contribution of this work with respect to relative efforts omitting accounting for realistic communication environment characteristics. The degree by which this requirement is met is depicted in Figure 7 where the percentage of fuzzy executions not having the latest values of the fuzzy input variables is depicted (through the indication of being not updated). As shown $tf1$ and $ft/tf1$ ratio comprise the most important factors influencing this metric.

In fact only when high $tf1$ (e.g., equal to 2 sec) is accompanied by high $ft/tf1$ (e.g., equal to 1.5) can the measured percentages be considered adequately low (i.e., fuzzification occurs almost in every case with the latest required values). The second critical observation concerns the drastic degradation indicated when $ft/tf1$ ratio is lower than one (1) corresponding to cases where ft expires before the generation of the new FSNs values. In such situations an even relatively low delay of the FSN's packet transmission may lead to nonupdated execution. It is interesting to note the fact that while this ratio is higher than one, nonupdated FIS executions never exceed 30%. On the other hand, in scenarios with $ft/tf1$ lower than one (1), nonupdated fuzzifications are higher than 30% even for $tf1 = 2$ sec while increasing

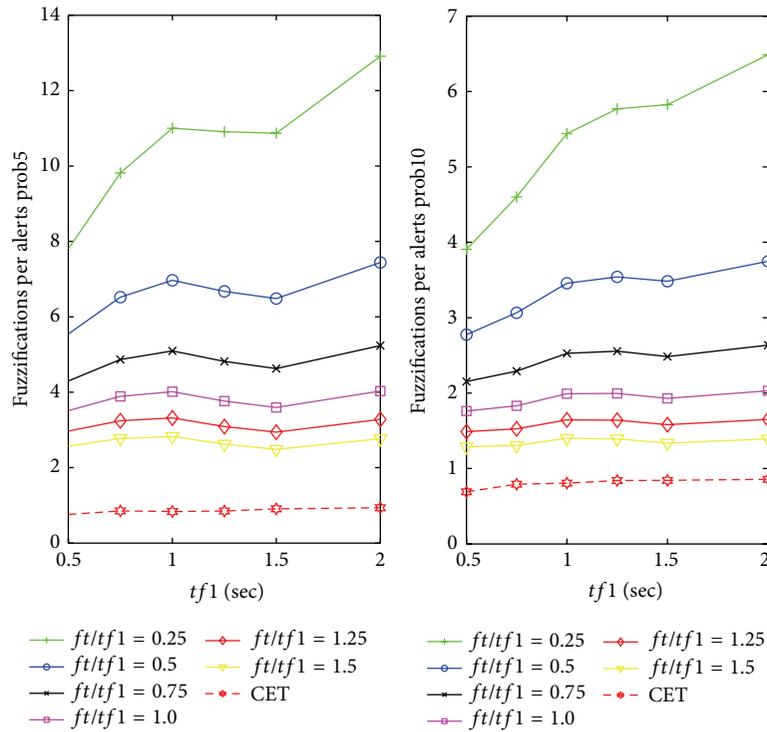


FIGURE 6: Fuzzy executions/alerts $tf2 = 2$ sec.

rapidly inversely proportional to $tf1$ reaching up to 70%–80% and even 90% where effectively most of the fuzzifications use nonupdated values thus compromising event detection accuracy. Event based execution also indicates dependency on $tf1$ values, while the percentage of the nonupdated fuzzy processes ranges from 11% to 30% indicating a considerably enhanced FIS reliability, for a wide range of realistic WSN applications. As it can be easily observed the CTT ($ft/ft1 = 1.25$ and $ft/ft1 = 1.5$) presents a slightly lower percentage of nonupdated fuzzy executions which never exceeds 11%. However in such situations the CTT approach may omit alarm events because of the FIS frequency execution.

As already mentioned the main reason for nonupdated fuzzy executions is the error prone nature of the medium. So it is important to evaluate the communication errors with respect to communication parameter so as to understand the degree of respective effect on the event detection accuracy. The packet loss percentage is measured with respect to communication conditions considered in this evaluation effort. Packet losses are dependant on $tf1$ and $ft/ft1$ in an analogous way as observed in the nonupdated metric measurements in Figure 7. Therefore, a direct correlation can be identified between network conditions (through packet losses) and fuzzy event algorithm’s accuracy (through nonupdated fuzzification executions). However, a significant difference concerns the drastic degradation observed in Figure 7 when $ft/ft1$ ratio was lower than one (1) attributed to the fact that ft expires before the generation of the new FSNs values. In all cases CET implementation offered the lowest packet loss measurements as well as the most agnostic to $tf1$ and $ft/ft1$ parameters.

In Figure 8 the percentage of the missed alerts is depicted, comprising another critical performance metric. The graphs of the CTT have been extracted considering a 10% probability of generating an alarm value. As we have already discussed a missed alert is defined as an alarm FSN value which is not taken into consideration in the timewise closest FIS execution. In other words, assuming FSN monitors a nonnormal value and considering that following a specific time interval (depending on ft value in case of CTT) the FIS will be executed, if the FIS inputs do not include this alert value then a missed alert is recorded. This can be due to network characteristics (loss/delay) or because of a new FSN’s sensing value which leads to another linguistic variable (e.g., temperature has a value of 100) which is considered as high. If ft is high (4 sec) T -FSN will generate a new value which may be 80 degrees, which is not considered as alarm. Hence, the timewise closest FIS will be executed taking into consideration the 80 degrees and the alarm value of 100 degrees is considered as missed alert. In the CET approach, the missed alerts are attributed to networking characteristics. The ratio of the “missed alerts/alerts” represents the missed alerts percentage. As it can be easily observed when the ratio $ft/ft1$ is quite low, a stressed communication scenario is considered. FIS execution frequency is highly relative to packet creation frequency; thus the generation of an alert value may not be in time when the FIS is executed. Stressed network conditions are also formed when $tf1$ assumes values in 0.5–1 sec range. As $tf1$ increases, the number of aggregate transmissions decreases and the network traffic is more relaxed. An observation worth noting is that graphs of $ft/ft1 = 1.25$ and 1.5 present higher percentage of missed alerts

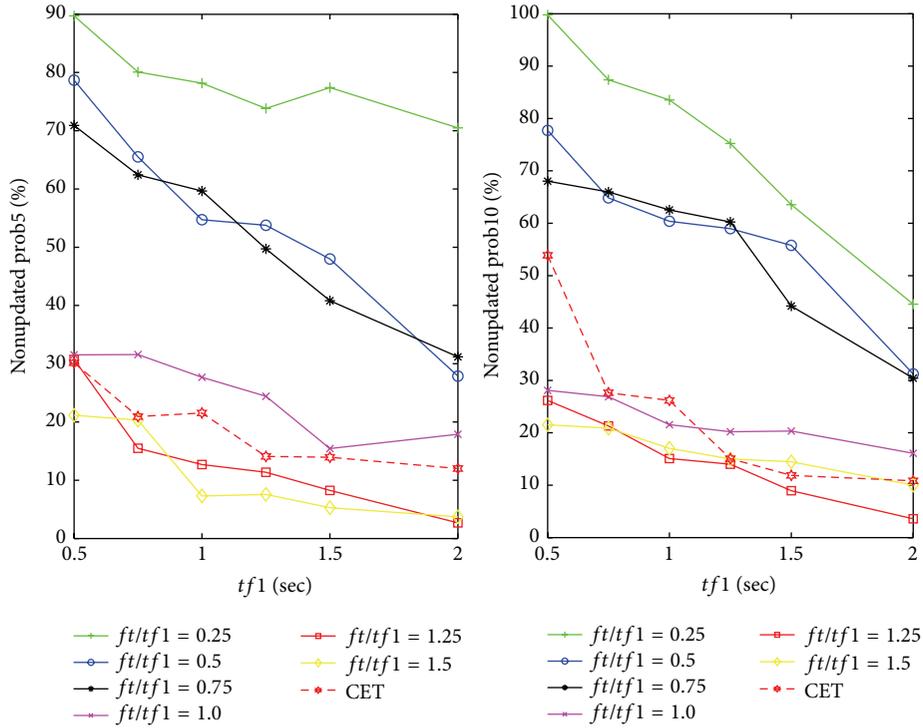


FIGURE 7: Nonupdated fuzzy executions $tf2 = 2$ sec.

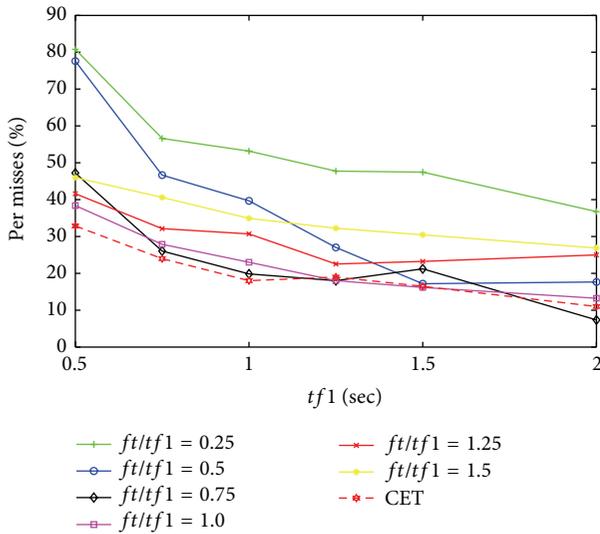


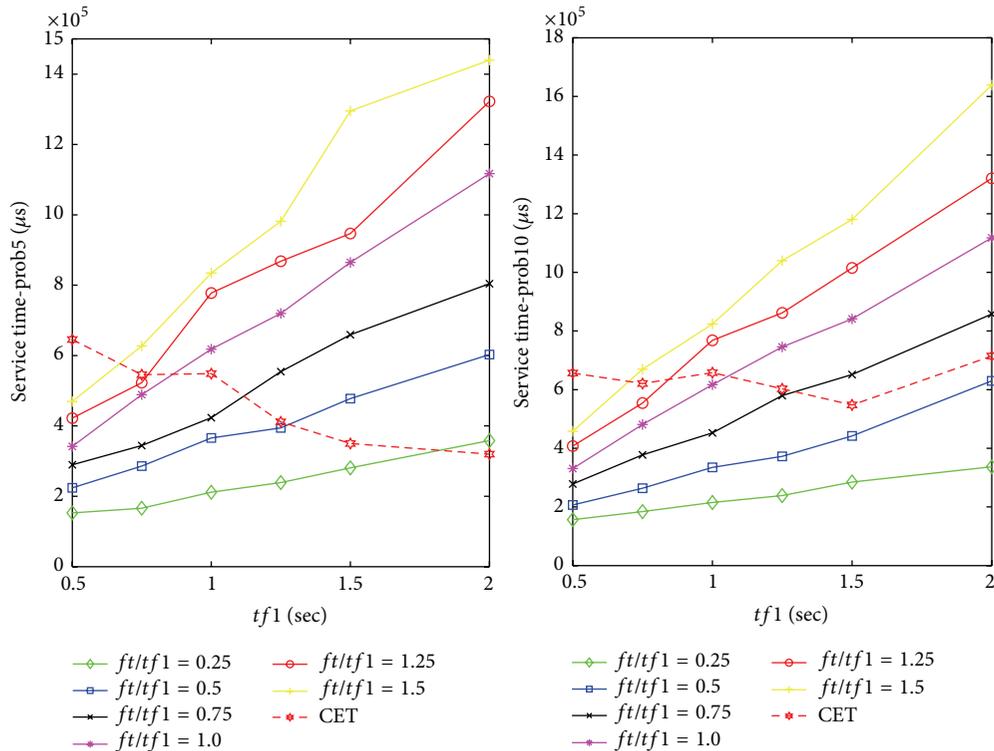
FIGURE 8: Percentage of missed alerts $tf2 = 2$ sec.

compared to those of $ft/ft1 = 0.75$ and 1. This is attributed to the fact that as the FIS's frequency is reduced, an instant alert value may be overwritten by its next generating value. In that case the initial alert value is not taken into consideration and a missed alert occurs.

Considering time constrained applications one of the most important performance aspects of an event detection algorithm concerns the time delay required for an alert to be identified, which effectively relates to the real time

capabilities of the event detection system. Such an evaluation is presented in Figure 9 considering a 10% probability for alert occurrence per packet creation and non-FSNs generate packets every 2 sec. As depicted the $ft/ft1$ ratio represents the most important factor. Specifically it can be observed that when fuzzification occurs faster than data packets are created (e.g., $ft/ft1 = 0.25$) the event is rapidly identified in the range of $2-3.5 \times 10^{-1}$ s. Furthermore, as this ratio increases so does the dependency on $tf1$ parameter itself, thus leading to significant increase of slope for all graphs for $ft/ft1$ ranging from 0.5 up to 1.5. On the contrary event based implementation performance is not affected by $tf1$ value exhibiting considerably steady measurements for the whole range considered. Furthermore, these measurements indicate that both implemented FIS execution approaches can offer significant advantages while the optimum selection depends significantly upon the specific application and algorithm configuration scenario.

7.3. The Effect of Task Allocation Scheme on the Event Detection Algorithm Performance. Following the comparative evaluation of the time triggered against the event triggered approach, this section focuses on the comparison between the aforementioned approaches and the distributed approach so as to gain respective insight concerning the performance of different task allocation techniques. CTT and CET are centralized in the sense that all components of the FIS execution take place in the FCHN. Thus all the data flows are directed to the FCHN, while all the CPU resources allocated to the FIS execution burden exclusively the FCHN. The main objective

FIGURE 9: Mean service time $tf2 = 2$ sec.

of the distributed approach is to mitigate this behaviour by offering a more evenly distributed CPU utilization among the FSNs. This will lead to diffused data flows and to a more well balanced energy consumption distribution. In addition a reduction to the number of packet transmissions is also achieved as the on-site processing (i.e., processing done in each node) will prevent unnecessary transmissions from one node to the node which is responsible for the aggregation and the defuzzification functions. Consequently an exploration of the above approaches is attempted, so as to discuss the respective pros and cons of each approach and the suitability of each approach in conjunction with specific networking scenarios.

The topology of this setup is different from the first one, as the FSNs must be in direct communication with each other, as well as with the FCHN. The packet rate by which the non-FSNs effectively interfere with the direct transmission of the FSNs represents a critical configuration parameter. The probability of each node generating a value which leads to the triggering of at least one FIS rule with consequent median/high level of fire probability detection is configured at 10%. The fuzzy timer is configured to two values, 0.5 sec and 2 sec, corresponding to normal and relaxing monitoring frequency, respectively.

In Figure 10(a) the number of fuzzy executions per alert is depicted for $ft = 1$ sec. For each simulation setup ($tf1 - tf2 - ft$) the total number of alerts is counted with respect to the number of FIS executions. An alert event is defined as a fuzzy value that can trigger at least one rule potentially causing medium or high probability of fire. The first important

observation concerns the number of fuzzy executions per alert considering the CET being steadily around one (1) FIS execution. This behaviour is anticipated since the FIS is triggered only when an alarm is generated. However as it can be seen this absolute value is slightly lower than one. This is due to cases where multiple alarms (i.e., one value triggering a fuzzy rule) occur in one fuzzy execution interval.

On the contrary, time triggered approaches (either centralized or distributed) tend to overuse the fuzzy algorithm. This is depicted by FIS being executed very frequently and thus the number of fuzzy executions per alert being significantly higher. Furthermore as it is depicted as $tf1$ increases, the fuzzy executions/alerts ratio also increases for the same probability of generating alarm packets (10%). This is attributed to the fact that when $tf1$ increases fewer packets are generated for the same simulation time period compared to a lower $tf1$ value. Hence, since the probability of generating alarm packets is predefined, fewer alarm packets are generated, at a specific time interval. However, in the CTT approaches the number of fuzzy executions depends only on ft value. Thus, the number of fuzzy executions per alert increases as $tf1$ also increases.

In Figure 10(b) event based implementation is the most resource efficient option since in all cases mean of fuzzy executions per alert is below one (1). Focusing on time triggered implementation cases both (i.e., centralized and distributed) are heavily dependent upon $tf1$. Specifically, lower $tf1$ (i.e., more stressed scenarios) lead to reduced fuzzy executions per alert, thus attributed to resource conservation. On the contrary when $tf1 = 2$ sec respective measurements

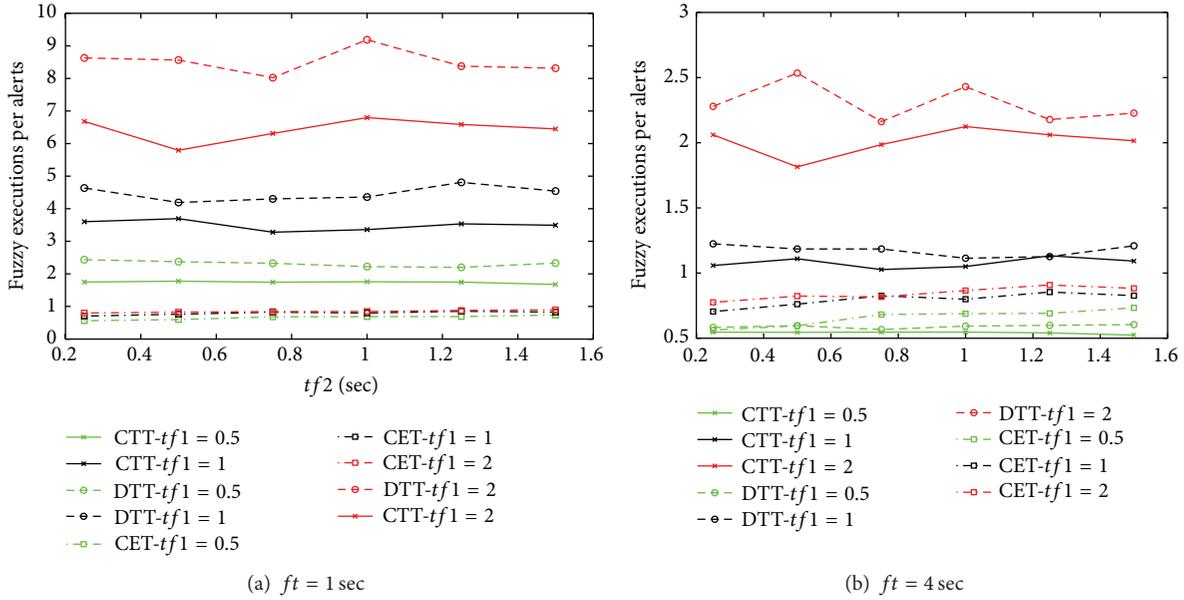


FIGURE 10: Fuzzy executions/alerts.

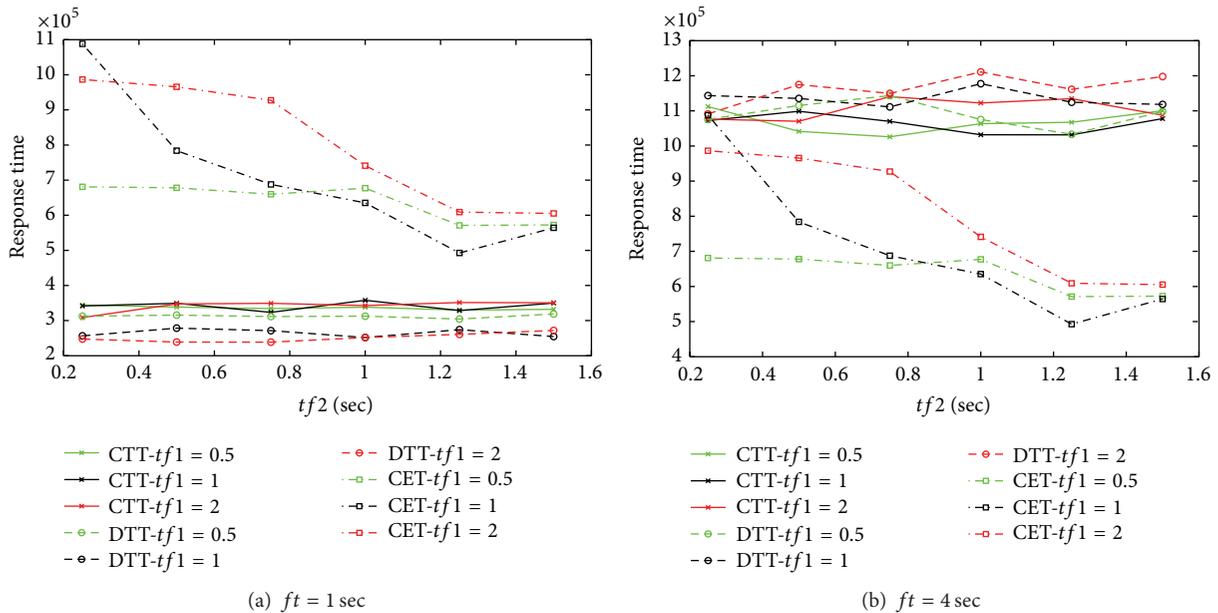


FIGURE 11: Mean service time— μs .

indicate that 2–2.5 fuzzy executions are triggered per alert. In Figure 10(b) the number of fuzzy executions per alert is depicted. As observed there is negligible deviation between time triggered and event triggered approaches. This is due to ft 's value which is quite high and thus the environment is monitored relatively infrequently. The number of alarm packets is similar. Consequently with respect to this metric centralized approach exhibits a preferable behaviour since the number of executions per alert is lower compared to distributed approach.

Finally, this metric offers an objective measure of resource wastage stemming from the fact that FIS is unnecessarily executed which will be quantified in the following figures.

In Figure 11(a) the mean response time between the generation of an alarm packet and the moment the timewise closest FIS execution starts is depicted. Without a doubt, rapid detection of an alarm is of paramount importance to a highly critical application scenario, such as the identification of a potential fire event, since it is directly related to the real time capabilities of the system and thus one can deduce if

specific requirements can effectively be met. On one hand centralized approaches are able to trigger the FIS in response to an alarm packet quite rapidly (few seconds) mainly in correlation to the frequency of the fuzzy timer. On the other hand in the event based approaches the response of the FIS is delayed mainly because of the broadcast transmission and the unicast transmissions of the FSNs to the FCHN. Specifically both time triggered based implementation cases have a clear advantage over event based FIS enabling event detection in the range of 2.5 up to 3.5 sec from the moment the alarm was issued. Among the two time triggered implementation cases, the distributed approach has a small advantage over the centralized one regarding robustness to interference, since event detection delay is quite steady at 3.5 sec. As opposed to such robust and predictable performance event based approach exhibits a significantly fluctuating performance from 6 sec event detection delay up to 11 sec indicating a rather poor time constrained performance. Another critical advantage of time triggered approaches concerns the immunity exhibited with respect to $tf2$ parameter. In fact no variation can be identified varying $tf2$ from 1.6 to 0.2 sec corresponding to a quite stressed traffic scenario. On the contrary due to the significance of broadcast transmissions of alarm values in CET approach the effect of non-FSNs' traffic is quite pronounced. Specifically, in such cases increasing non-FSN traffic (i.e., decreasing $tf2$) from 1.6 to 0.2 sec while assuming a rather relaxed FSN traffic rate leads to response time increase ranging from 60% (for $tf1 = 2$ sec) up to 80% ($tf1 = 1$). Overall it can be argued that time triggered approaches are advocated when real time event detection is required and significant competing traffic is created in the FIS vicinity.

In Figure 11(b) the same metric is presented using $ft = 4$ sec. For the time triggered approaches, the response time increases as ft increases, while the deviation between DTT and CTT approach is negligible. Also, it is depicted that the event based approach offers the best recorded performance. This indicates that the use of the proposed event based FIS scenario in surveillance applications, which are characterized by real time (or in more general terms time constrained) requirements, is advisable. In addition in the CET graphs the response time increases analogously to $tf1$, in the most stressed conditions ($tf2 < 0.9$ sec). Based on trace file analysis this is attributed to nontriggered FIS execution due to losses of alarm or broadcast packets. As $tf1$ is high the interalarm time delay increased leading to a delayed fuzzy execution. Hence, the service times of the previous lost alarm refer to the current fuzzy execution, which effectively lead to significant increase of the overall response time. As a general conclusion it is noted that, in time triggered approaches, response time is effectively controlled by ft period augmented by a constant factor due to data transmission characteristics. Regarding the CET approach the percentage is quite low. It is worth noting that, in case FSN insists on generating alarm values until the fuzzy execution starts, the last generated alarm value is considered as input to the fuzzy execution. However the previous values are considered as missed.

Figure 12(a) indicates the percentage of the fuzzy processes being executed based on nonupdated data. The first observation concerns the advanced performance exhibited

by event based implementation, as the data flows are quite relaxed and the FSN's nodes send their packets only when there is a necessity. The percentage of the nonupdated fuzzy executions is also lower in case of the distributed approach, because of the decentralized data flows.

Furthermore, for FIS detection system's accuracy, it is required that when FIS is executed all relevant input values are updated. Respective measurements indicating the degree by which this requirement is invalidated are presented in Figure 12(b) also offering significant observations. As observed the percentage of nonupdated FIS execution is analogously affected by the traffic rate.

As depicted, the three different implementation approaches offer distinctively different performance measurements. In that respect DTT offers the most accurate FIS execution exhibiting in all $tf1$ cases the lowest nonupdated number of FIS executions. This can be attributed to the fewer number of transmissions in DTT. In addition, the number of fuzzy executions is lower in the CET; hence a potential packet loss or a case of increased delay of FSN packet leads to a nonupdated event. The DTT does not suffer from this kind of problems in this setup, as ft is quite high compared to $tf1$ thus allowing the retransmission of the lost packet before FIS is actually executed. Then the second best performance is offered by the CET implementation whereas CTT approach measurements indicate a rather inaccurate FIS since even in medium stressed non-FSN traffic scenarios (related to $tf2$ parameter) the nonupdated FIS execution percentage is close to or even higher than 50%.

Pertaining to network robustness packet loss represents a critical factor. This is clearly depicted in Figure 13(a) where the packet loss percentage is presented considering the three different implementation proposals, while varying $tf1$ and $tf2$ parameters. The first observation concerns the advantage exhibited by the event based centralized case offering the lowest packet loss percentages compared to other approaches. Indicatively, comparing CET graph against CTT concerning $tf1 = 0.5$, while packet loss difference is approximately 20% for $tf2 = 0.75$, it increases to 30% for $tf2 = 0.25$. This indicates the non-FSNs generation rate is the dominant traffic factor in the network. In addition in case of the CET the networking flows from the FSNs to the FCHN are fewer as the FSNs send their data to the FCHN only when needed. Furthermore, as anticipated, $tf1$ is inversely analogous to packet loss, since it leads to less stressed communication scenarios. Additionally, compared to time triggered approaches, distributed implementation seems to have a slight advantage over centralized one since the networking flows are diffused and the number of concurrent data flows to the FCHN is also decreased. This is reflected in Figure 13(a).

Similar measurements are presented using $ft = 4$ sec in Figure 13(b). As it can be easily seen CTT presents the higher level of packet loss ranging from 13% up to 30%, $tf1 = 0.5$ sec. It is worth mentioning that while $tf2$ increases, DTT and CET have similar behaviour. This is due to the fact that FSNs transmit their value to the coef.FSN diffusing the network flows and leading to more relaxed network conditions, while at the same time the coef.FSN reduces its transmissions to the FCHN because of the high value of ft . However, CET seems

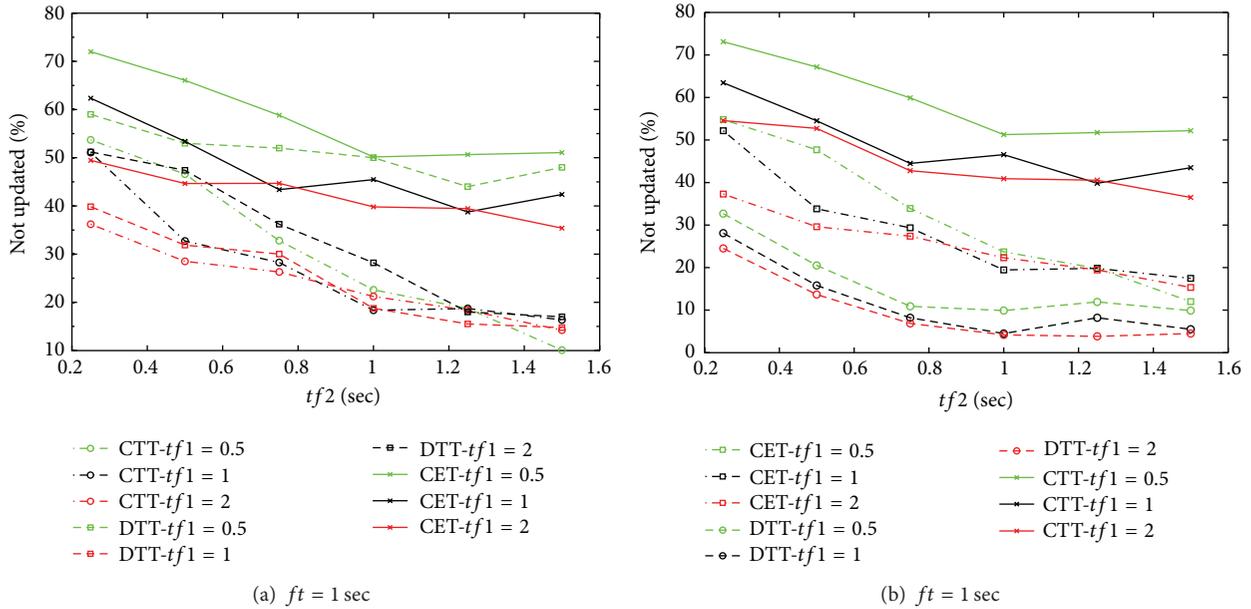


FIGURE 12: Percentage of nonupdated fuzzy executions.

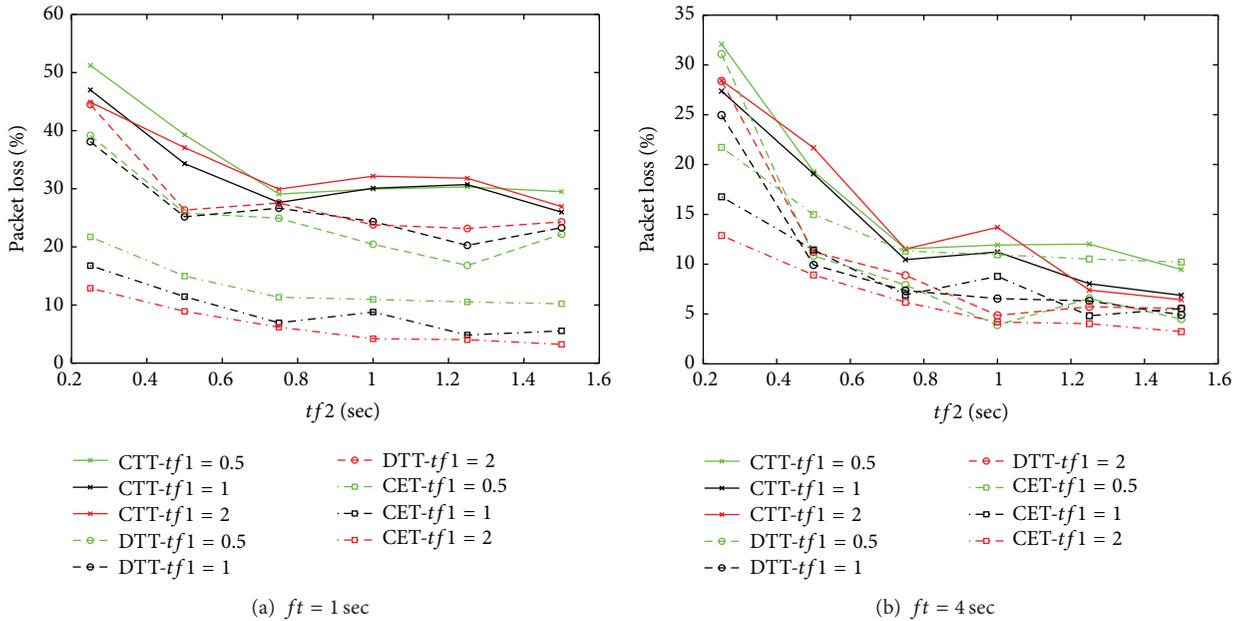


FIGURE 13: Packet loss.

to have the best overall performance particularly regarding low tf_2 values.

Focusing on the FIS's reliability Figure 14(a) presents the missed alerts measurements. A missed alert is defined as the generation of an alarm packet that is not considered as input to the timewise closest fuzzy execution. In the CET this is mainly attributed to packet loss or to network delay. However, in time triggered approaches a missed alert occurs when the generation of an alarm packet (i.e., following the value triggering FIS rule) happens before the execution of the fuzzy process and finally this value is overwritten by the

generation of a new sensing value (i.e., normal value). For example, if an alarm packet is generated but ft is quite high, it is highly possible that new values will be generated. Hence the FIS will be executed with the last received values. In such a case a missed alert can occur if the sensor values turn from abnormal to normal ranges, between two consequent FIS executions. What is most interesting in this case is the fact that distributed approach exhibits the best performance since the missed alert percentage never exceeds 10%. This can be attributed to the decentralized approach, where the packet loss is lower, as well as to the number of the fuzzy executions.

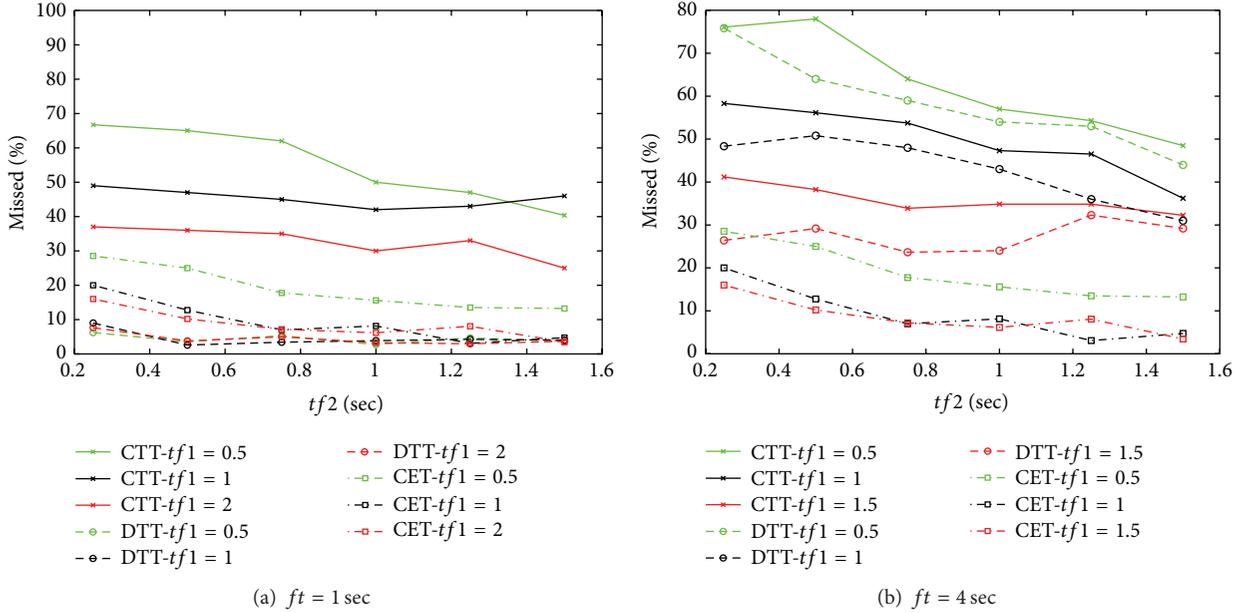


FIGURE 14: Percentage of missed alerts.

Based on such observations, event based implementation also offers a robust behaviour since in all cases respective percentages are approximately 10% except the scenarios of lowest tf_2 (i.e., the most stressed communication scenario for non-FSNs) for which respective percentage is close to 20%. Also it is worth noting that considering distributed implementation tf_1 is effectively negligible, while for the event based triggering considerable deviation with respect to tf_1 can be observed only for the lowest $tf_1 = 0.5$. On the contrary CTT approach behaviour can be considered rather unreliable since for medium tf_1 level approximately half of the alerts are missed. Furthermore, reliability is closely correlated to tf_1 since significant deviation is depicted upon reducing the FSN packet creation interval from 2 sec to 0.5 sec.

In Figure 14(b) the same metric is also presented using $ft = 4$ sec. The difference between a nonupdated event and a missed alert may lead to misunderstanding. Specifically, a “nonupdated” case does not count the number of input values that are not updated in each specific FIS execution; that is, if any value is not up to date then a nonupdated event occurs. Of course this also has to be taken into consideration for the calculation of the missed alerts, that is, assuming a scenario where all sensors generate values outside the normal ranges. One execution of the FIS should take as input the latest nonnormal values. If only a packet is lost then the fuzzy process is considered not updated, while a missed alert occurs. If this happens, for example, 4 times then the percentage of missed alerts is 4 (missed alerts)/16 (alerts) while the percentage of the nonupdated fuzzy processes is 4/4 = 100%. Based on this, Figure 14(b) shows the advantage of CET in terms of true negatives ratio.

The number of transmissions has also been evaluated. The most important overall observation concerns the fact

that in all traffic scenarios event based implementation offers significant benefits.

8. Discussion

In this section, a summary of the evaluation analysis on the three described techniques (CTT, CET, and DTT) is presented, so as to offer valuable insights of each approach, with respect to specific application’s requirements. The three WSN tailored implementation approaches are proposed considering both optimal task allocation among WSN nodes and alternative fuzzy logic algorithm execution strategies. Regarding the former aspect distributed design as well as implementation is presented yielding optimum processing and communication resource utilization. With respect to the latter aspect event triggered algorithm design is analyzed and evaluated aiming to offer significant reduction of resource wastage.

CTT is a centralized approach, in the sense that all algorithmic subtasks of the FIS are executed to the FCHN. The execution of the FIS in one node increases significantly the CPU utilization of this node. Moreover, the centralized nature of this approach means that the data flows from the FSNs are all directed to the FCHN. Taking into account that the non-FSN’s packets are also directed to the FCHN, the radio utilization of this node is also increased. Moreover, because of these concurrent data flows to the FCHN, networking congestion leads to low quality of service metrics (i.e., packet loss). In addition application oriented metrics (i.e., percentage of nonupdated fuzzy processes, mean service time when an alert is generated) are also affected by the network confidence status. The periodic execution of the FIS (fuzzy timer) is also a parameter that can reduce network lifetime, if the FIS is executed, when none of its inputs can

trigger an alarm situation. Therefore, on one hand CTT offers low complexity and adequate performance when concurrent transmissions are limited. On the other hand, however, conducted experiments highlighted significant vulnerability with respect to increased neighbor traffic as well as considerable energy wastage in low frequency event creation cases. DTT is a distributed approach, in the sense that the algorithmic subtasks of the FIS are allocated to all FSNs. The distributed execution of the FIS in the FSNs utilizes nodes' CPU power, prevents possible burdening of the FCHN, and achieves a more well balanced energy consumption distribution. In addition, the distributed implementation diffuses the data flows, leading to a lower network congestion and optimizing network quality of service metrics. Therefore and based on performance metrics presented, although a complexity overhead is posed, DTT exhibits a clear advantage over CTT in scenarios characterized by high congestion probability. Also DTT proposed design is advocated when central node power consumption is of paramount importance; for example, respective device is powered by small, difficult-to-renew batteries. Moreover the on-site processing reduces the number of transmissions between the FSNs, as the *coef_FSN* serves as receiver for the other FSNs as well as the computation unit for performing fuzzification of its data, evaluating the rules' weights and applying the implication process. In that respect and considering applications where the execution of FIS should be performed quite often, it seems that DTT offers a highly efficient solution. However, when the occurrence of abnormal events is rare, the periodic nature of the DTT approach handles CPU utilization in a nonoptimal way. In this case our third proposed design indicated as CET comes into play. CET is a centralized approach, as the FIS execution takes place in the FCHN. This technique overcomes important limitations of the FIS periodic execution based on the fuzzy timer. Each FSN knows the threshold value, which can trigger an alarm situation. Thus, only when FSN detects an abnormal value does it inform its logical neighbors (FSNs) about the necessity of executing the FIS. In that case, the FSNs send their data to the FCHN, where the criticality of the situation is evaluated. In conjunction with the event oriented nature of this approach, CET algorithm leads to a significant decrease of network traffic, offering high network confidence rate and algorithm's performance. Unnecessary communications are avoided and when an alert packet is generated, its service time is almost instantaneous. For applications, where the FIS is not needed to be executed quite often, CET offers a valuable solution. When alarm packets are generated in a high rate, CET approach is similar to CTT.

9. Conclusions and Future Work

Wireless sensor networks are increasingly utilized in a wide range of highly demanding applications where event detection effectively depends on multimodal data fusion. In such cases simplistic approaches based on crisp value evaluation are not able to offer adequate reliability and event detection performance. Therefore WSN developers turn towards more sophisticated classification algorithms like

fuzzy logic and Bayesian Networks as prominent approaches able to address demanding application scenarios. However, though well known and widely utilized in typical networks, design and implementation of such algorithms in WSNs pose critical challenges due to the error prone nature of the communication medium and scarce resource availability of typical WSN platforms.

Therefore, this paper tackles these challenges by designing, implementing, and evaluating fuzzy logic based fire detection algorithms specifically focusing on WSNs. The application of classification models enables the handling of large volumes of raw data, thus achieving better control of the monitored environment, dealing with the uncertainty and minimizing redundant communications between wireless nodes. However, additional processing capabilities are required for the classifications algorithms execution. Monolithic implementation of a classification algorithm in one node, which effectively serves as a fusion cell, leads to energy exhaustion of this node, which will result in the applications failure. In addition the concurrent data flows towards a single node (handling all the processing) reduce the available bandwidth, leading to increased packet loss ratio. Driven by such observations and with respect to WSN characteristics in this work, WSN tailored implementation approaches are proposed considering both optimal task allocation among WSN nodes and alternative fuzzy logic algorithm execution strategies. Regarding the former aspect distributed design and implementation are presented yielding optimum processing and communication resource utilization. With respect to the latter aspect event triggered algorithm design is analyzed and evaluated aiming to offer significant reduction of resource wastage.

Additionally, significant effort is exerted so as to offer highly objective and practical evaluation setup. In that respect all designs presented are implemented in the context of a widely utilized WSN operating system (i.e., Contiki) effectively proving the feasibility of respective solutions. Even more, evaluation is conducted using the respective highly accurate WSN emulator (i.e., Cooja) incorporating communication and processing characteristics in a valid and objective manner. Based on numerous experiments varying a wide range of critical network and fuzzy logic parameters, respective measurements' analysis showed significant advantages offered by distributed implementation regarding energy balance, packet loss, and algorithms performance in terms of nonupdated fuzzy executions. Event based execution of the FIS also showed significant advantages in terms of packet loss, number of transmissions, mean service time of a generated alarm, and better utilization of the FIS algorithm. To conclude, extracted observations revealed valuable insights of each approach with respect to specific requirement and performance metric while as future work such approaches can be further optimized and adapted so as to offer even better performance in more application scenarios.

Most of our effort has focused on providing a comprehensive analysis of the three objectives that this work addresses: (i) implementation of a fuzzy system in a WSN infrastructure, (ii) task scheduling of the fuzzy system into subtasks that are executed by different sensor nodes, so as to balance the

computational load, and (iii) event dissemination algorithm so as to optimize the performance of this system. We believe that there is significant room for further investigation by providing experimental results, from a real WSN deployment without the use of any simulation tool. However, in this work, we provided evaluation results extracted by the well-adapted Cooja simulator, so as to establish a yet flexible framework of the proposed methodology, while avoiding limitations of a real WSN deployment, that is, number of nodes, hardware failure, inference, uncertainty, and difficulty in repeating specific conditions. Consequently, an important future objective of our work is to provide a comparative evaluation analysis between a simulation environment and a real WSN deployment.

Finally our method could be extended by incorporating advanced spatial and temporal characteristics into the FIS so as to deal with other problems such as nodes' uncertainty or nodes' remaining energy. These characteristics will improve FIS accuracy as it deals with the error prone nature of a WSN system. However these parameters have not been taken into consideration in this work, as this paper main focus is on proposing adequate methodologies of applying fuzzy systems in WSNs for event detection.

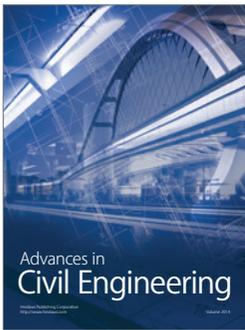
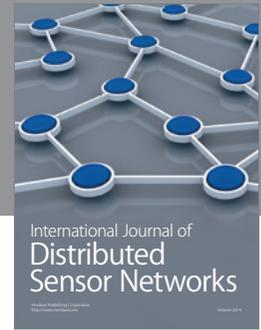
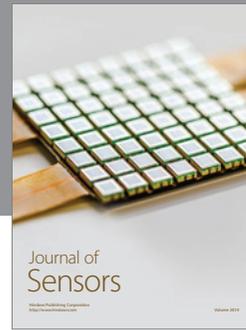
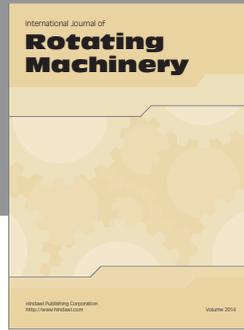
Competing Interests

The authors declare that they have no competing interests.

References

- [1] G. Wittenburg, N. Dziengel, C. Wartenburger, and J. Schiller, "A system for distributed event detection in wireless sensor networks," in *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN '10)*, pp. 94–104, Stockholm, Sweden, April 2010.
- [2] D. Li, K. D. Wong, Y. H. Hu, and A. M. Sayeed, "Detection, classification, and tracking of targets," *IEEE Signal Processing Magazine*, vol. 19, no. 2, pp. 17–29, 2002.
- [3] M. Bahrepour, N. Meratnia, and P. J. M. Havinga, "Automatic fire detection: a survey from wireless sensor network perspective," December 2008, <http://doc.utwente.nl/65223/>.
- [4] J. Barbarn, M. Daz, I. Esteve, D. Garrido, L. Llopis, and B. Rubio, "Rtcoord: a methodology to design wsan applications," *Proceedings of World Academy of Science: Engineering & Technolog*, vol. 21, p. 144, 2007.
- [5] E. F. Nakamura, A. A. F. Loureiro, and A. C. Frery, "Information fusion for wireless sensor networks: methods, models, and classifications," *ACM Computing Surveys*, vol. 39, no. 3, article 9, 2007.
- [6] N. D. Phung, M. M. Gaber, and U. Röhms, "Resource-aware online data mining in wireless sensor networks," in *Proceedings of the 1st IEEE Symposium on Computational Intelligence and Data Mining (CIDM '07)*, pp. 139–146, Honolulu, Hawaii, USA, April 2007.
- [7] S.-M. Dima, D. Tsitsipis, C. Antonopoulos, J. Gialelis, and S. Koubias, "FLOGERA: a fuzzy logic event recognition algorithm in a WSN environment," in *Proceedings of the 8th IEEE International Wireless Communications and Mobile Computing Conference (IWCMC '12)*, pp. 850–855, IEEE, Limassol, Cyprus, August 2012.
- [8] S. M. Dima, C. Panagiotou, D. Tsitsipis, C. Antonopoulos, J. Gialelis, and S. Koubias, "Performance evaluation of a WSN system for distributed event detection using fuzzy logic," *Ad Hoc Networks*, vol. 23, pp. 87–108, 2014.
- [9] E. H. Mamdani, "Application of fuzzy algorithms for control of simple dynamic plant," *Proceedings of the Institution of Electrical Engineers*, vol. 121, no. 12, pp. 1585–1588, 1974.
- [10] R. R. Yager and D. P. Filev, *Essentials of Fuzzy Modeling and Control*, John Wiley & Sons, New York, NY, USA, 1994.
- [11] P. Manjunatha, A. K. Verma, and A. Srividya, "Multi-sensor data fusion in cluster based wireless sensor networks using fuzzy logic method," in *Proceedings of the IEEE Region 10 and the 3rd International Conference on Industrial and Information Systems (ICIIS '08)*, pp. 1–6, Kharagpur, India, December 2008.
- [12] L. Zhang, H. Leung, and K. C. C. Chan, "Information fusion based smart home control system and its application," *IEEE Transactions on Consumer Electronics*, vol. 54, no. 3, pp. 1157–1165, 2008.
- [13] K. Kapitanova, S. H. Son, and K.-D. Kang, "Using fuzzy logic for robust event detection in wireless sensor networks," *Ad Hoc Networks*, vol. 10, no. 4, pp. 709–722, 2012.
- [14] S. A. Munir, Y. W. Bin, R. Biao, and M. Man, "Fuzzy logic based congestion estimation for qos in wireless sensor network," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '07)*, pp. 4336–4341, IEEE, Hong Kong, 2007.
- [15] M. Marin-Perianu, C. Lombriser, O. Amft, P. J. M. Havinga, and G. Troster, "Distributed activity recognition with fuzzy-enabled wireless sensor networks," in *Distributed Computing in Sensor Systems: 4th IEEE International Conference, DCOSS 2008 Santorini Island, Greece, June 11–14, 2008 Proceedings*, vol. 5067 of *Lecture Notes in Computer Science*, pp. 296–313, Springer, Berlin, Germany, 2008.
- [16] A. G. Dharne, J. Lee, and S. Jayasuriya, "Using Fuzzy Logic for localization in mobile sensor networks: simulations and experiments," in *Proceedings of the American Control Conference*, pp. 2066–2071, IEEE, Minneapolis, Minn, USA, June 2006.
- [17] A. Barati, S. J. Dastgheib, and A. Movaghar, "An effective fuzzy based algorithm to detect faulty readings in long thin wireless sensor networks," *International Journal on Technical and Physical Problems of Engineering*, vol. 10, no. 4, pp. 52–58, 2012.
- [18] Y. Tian, Y. Gu, E. Ekici, and F. Ozguner, "Dynamic critical-path task mapping and scheduling for collaborative in-network processing in multi-hop wireless sensor networks," in *Proceedings of the International Conference on Parallel Processing Workshops (ICPP '06)*, p. 222, Columbus, Ohio, USA, 2006.
- [19] L. Zhigang, L. Shining, Z. Xingshe, and Y. Zhiyi, "Energy-efficient task allocation for data fusion in wireless sensor networks," in *Proceedings of the 6th Annual International Mobile and Ubiquitous Systems: Networking Services (MobiQuitous '09)*, pp. 1–6, November 2009.
- [20] J. Jinghua Zhu, J. Li, and H. Gao, "Tasks allocation for real-time applications in heterogeneous sensor networks for energy minimization," in *Proceedings of the 8th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD '07)*, vol. 2, pp. 20–25, IEEE, Qingdao, China, 2007.
- [21] H. Park and M. B. Srivastava, "Energy-efficient task assignment framework for wireless sensor networks," Tech. Rep., 2003.
- [22] F. Wang, G. Han, J. Jiang, and H. Qiu, "A distributed task allocation strategy for collaborative applications in cluster-based

- wireless sensor networks,” *International Journal of Distributed Sensor Networks*, vol. 2014, Article ID 964595, 16 pages, 2014.
- [23] F. Bouhafs, M. Merabti, and H. Mokhtar, “A semantic clustering routing protocol for wireless sensor networks,” in *Proceedings of the 3rd IEEE Consumer Communications and Networking Conference (CCNC '06)*, vol. 1, pp. 351–355, IEEE, Las Vegas, Nev, USA, January 2006.
- [24] L. Mottola and G. P. Picco, “Logical neighborhoods: a programming abstraction for wireless sensor networks,” in *Distributed Computing in Sensor Systems*, P. B. Gibbons, T. Abdelzaher, J. Aspnes, and R. Rao, Eds., vol. 4026 of *Lecture Notes in Computer Science*, pp. 150–168, Springer, Berlin, Germany, 2006.
- [25] A. R. Rocha, L. Pirmez, F. C. Delicato et al., “WSNs clustering based on semantic neighborhood relationships,” *Computer Networks*, vol. 56, no. 5, pp. 1627–1645, 2012.
- [26] L. Gu, D. Jia, P. Vicaire et al., “Lightweight detection and classification for wireless sensor networks in realistic environments,” in *Proceedings of the 3rd ACM International Conference on Embedded Networked Sensor Systems (SenSys '05)*, pp. 205–217, ACM, November 2005.
- [27] K.-X. Thuc and K. Insoo, “A collaborative event detection scheme using fuzzy logic in clustered wireless sensor networks,” *AEU-International Journal of Electronics and Communications*, vol. 65, no. 5, pp. 485–488, 2011.
- [28] L. A. Zadeh, “The concept of a linguistic variable and its application to approximate reasoning,” *Information Sciences*, vol. 8, no. 3, pp. 199–249, 1975.
- [29] J. M. Alonso, L. Magdalena, and S. Guillaume, “Designing highly interpretable fuzzy rule-based systems with integration of expert and induced knowledge,” in *Proceedings of the 12th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU '08)*, pp. 682–689, Torremolinos, Spain, 2008.
- [30] <http://www.contiki-os.org/>.
- [31] http://www.willow.co.uk/html/telosb_mote_platform.php.
- [32] J. Eriksson, F. Österlind, N. Finne et al., “COOJA/MSPSim: interoperability testing for wireless sensor networks,” in *Proceedings of the 2nd International ICST Conference on Simulation Tools and Techniques (SIMUTools '09)*, March 2009.
- [33] <http://www.ti.com/msp430>.
- [34] F. Kerasiotis, A. Prayati, C. Antonopoulos, C. Koulamas, and G. Papadopoulos, “Battery lifetime prediction model for a WSN platform,” in *Proceedings of the 4th International Conference on Sensor Technologies and Applications (SENSORCOMM '10)*, pp. 525–530, Venice, Italy, July 2010.
- [35] F. Österlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt, “Cross-level sensor network simulation with COOJA,” in *Proceedings of the 31st Annual IEEE Conference on Local Computer Networks (LCN '06)*, pp. 641–648, Tampa, Fla, USA, November 2006.
- [36] <http://www.shimmersensing.com/>.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

