

# Event Detection Using Wireless Sensor Networks

Irfana Memon, Fiza Syal, Rafia Naz

**Abstract**—Monitoring of environmental characteristics plays a key role in development of different scientific and engineering applications. To sense and process the data defining the environment sensors are normally deployed in the space under investigations. The data is recorded using Wireless Sensor Networks (WSNs) where the data is processed and communicated via radio without any additional prior backbone infrastructure. WSNs cover a wide range of applications such as battlefield surveillance, telemedicine, event detection. An early and accurate event detection using wireless sensor network is one of the key area focused by the researchers in the present times.

The present work focuses on the application of the existing protocols for WSNs and determine their limitations.

**Index Terms**—Wireless sensor networks; Event detection; Features of event detection schemes; Energy efficiency.

## I. INTRODUCTION

With the advancement of Micro-Electro-Mechanical Systems, wireless communications and digital electronics have facilitated growth of cheap Wireless Sensor Networks (WSNs). WSNs are an emerging special type of wireless ad hoc networks technology for cheap, and an open area WSNs are composed of large number of small, inexpensive devices, called sensor nodes, which are deployed either carefully or randomly over a geographical area to monitor it and these sensor nodes are networked through wireless links. In recent years, the research widened its scope in the area of WSNs from continuous data collection to detection of event.

Event is an incomparable change in environment parameter. Event detection is the process of detection of such a changeable environment phenomenon. There are two types of events: atomic event, and composite event. An atomic event may be identified based on observation of one attribute. Whereas, composite event is detected based on the combination of the multiple attributes.

Very little work has been proposed to address the event detection problem using WSNs. The purpose of this paper is to review the existing event detection protocols.

We have classified the event detection protocols on the basis of techniques followed by each protocol and is discussed in the next Section.

Irfana Memon, Department of Computer Systems Engineering, Q.U.E.S.T, Nawabshah, Fiza Syal, Rafia Naz, Department of Information Technology, Q.U.E.S.T, Nawabshah.  
Email: irfanahameed@yahoo.com

From the literature, it is manifested that in existing protocols suffer from high communication overhead (and thus high energy consumption) and do not address node failure. Through the critical analysis, we highlight the limitations of existing protocols for WSNs. This will make easier for new researcher to understand the problem and propose new solution to address the weaknesses of the existing protocols.

The rest of the paper is organized as follows. Section II gives critical analysis of existing event detection protocols. Section III presents basic requirements for designing event detection protocols. Section IV presents the conclusion and future work.

## II. CRITICAL ANALYSIS OF EXISTING EVENT DETECTION PROTOCOLS

In this section, we review and classify existing event detection protocols. The existing protocols to detect event can be categorized into three broad classes: (A) threshold-based protocols, (B) pattern-based protocols, and (C) fuzzy-based protocols.

### A. Threshold-based event detection protocols

Threshold-based techniques are based on threshold values of the characterizing parameter and use some if-then-else rules for event detection. For example, high temperature: IF temperature > 100 THEN Event (Probable failure) is detected.

Phani Kumar, AVU et al. [1] have proposed a method to detect event using collaboration. They have proposed two algorithms called simple event detection algorithm and composite event detection algorithm. Each protocol works in two steps: (i) initialization step, and (ii) data collection step. At the beginning, an application subscribes events of interest (atomic events or composite events) to the sensor network. Based on these events, an Event Based Tree (EBT) is maintained. In the collection phase, predicate data is collected along EBT and sent to the base station. The base station will take decision of event occurrence based on the received predicate data and then event is reported if all conditions are satisfied.

Vu, C.T. et al. [2] introduced an approach for event detection. The objective of the proposed approach is to assure that deployed environment is monitored by at least k sensors. Fig.1. represent the proposed approach.

Multiple types of sensors are used to sense area so that event can be detected according multiple attributes of event. For example, in Fig.1. 3 types of sensors: Sensor type 1 for

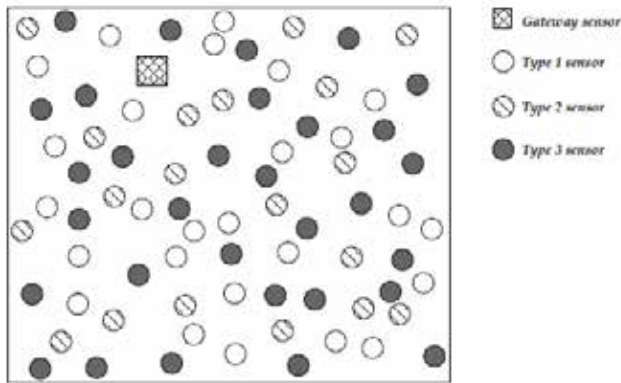


Fig. 1. Event query in a WSN [2]

temperature, Sensor type 2 for smoke density, and Sensor type 3 for light.

The proposed protocol divides all deployed sensors into a number of non-disjoint small set of node called detection sets. These sub set must assures that area is monitored with  $k$  sensors. This is done by base station. Gateway will apply Breadth First Search (BFS) algorithm to construct tree for each detection set. Then, information sensed by sensors is received along the tree. At the beginning, a node will be selected as gateway. Any sensor node with richer energy resource can be gateway. Then, gateway constructs tree with detection sets. For each event (i.e., atomic), a counter to count the number of sensors required to sense is maintained to achieve  $k$ -watching property for desired atomic event.

In this scheme, all the sensor nodes transmit their sensed data to a gateway node, and a decision for occurrence of event according to the received data is made by gateway node. The disadvantage of the proposed scheme is the requirement of global knowledge regarding sensors during the detection set construction.

Composite event detection scheme was developed by Marta, M. et al. [3] considering energy efficiency in wireless sensor network. They proposed a method to connect dominating set using scheduling. Tasks are prearranged in rounds. There are two steps in each round i.e., initialization step, and event detection step. Sensor nodes will take decision for their status (active or sleep status) during initialization step. At the initial step, algorithm apply the sensor scheduling. The proposed algorithm apply scheduling based on received information from neighbors of  $h$ -hops ( $h$  is design parameter). Each sensor node transmits <HELLO> within  $h$ -hops to maintain its  $h$ -neighbors list. The authors assumed some priority for nodes during scheduling, such as  $p(u)$  for node  $u$ . This priority is the pair of node's residual energy and node's identity. In the case, when nodes have similar identity, then node with larger energy will be selected. In the initial step, keep all sensors active.

Node  $u$ , has neighbor set referred as  $N'_h(u)$ :  $N'_h(u) = \{v \in N_h(u) | p(v) > p(u)\}$ . In 1st round, each node will make

decision about its status according to the set of neighbors, and connectivity with neighbors. So that, when sensor node  $u$  go to sleep, the area is monitored by its  $k$ -active neighbors to provide  $k$ -watching property.

The main limitation of this protocol was the data collection by each node from its neighbors, where by each node transmits signal to  $h$ -hop nodes resulting in high cost and probable message collisions.

Yang, Y. et al. [4] focused on composite event detection using scheduling method. The authors considered a network with large number of nodes deployed densely. Each node has multiple sensing components  $\{x_1, x_2, \dots, x_M\}$  for detecting a composite event. Where  $M$  is the number of sensing components. Keeping 'redundant' sensing components to sleep is one way to enhance network. In this work, the authors have proposed two solutions: a grid-based distributed algorithm, and a localized algorithm.

In the distributed algorithm, the deployed area is partitioned into grids. The scheduling method has two algorithms: (1) decide the sensing nodes 'Decide Status ()' and (2) decide the relay nodes needed to ensure a connected network 'Connectivity ()'. In localized approach, sensing components can be in active, pending, and sleep state for some assigned time. Each sensing component selects its status based on residual energy, its communication range, neighboring sensing components; and its identity.

Each node transmits <Hello messages> communicating its status, and its sensing components. In time interval  $t$ , each sensing component with pending state will initially detect the state of neighbor node and their connectivity; and if these are active, it will go to sleep mode; otherwise it will be active for this time interval.

#### B. Pattern-based event detection protocols

In Pattern-based event detection, events are defined with some pattern and data pattern. When pattern of data is same as pattern of event, then a composite event is detected. Events are defined as spatiotemporal patterns of sensory data. Patterns of data or event could be in format of object's movement etc.

Existing techniques typically employ contour maps [5], isolines [6], and gradient maps [7] to build patterns of events. Composite event is detected via comparing event pattern with current sensed data patterns,

Xue et al., [5] presented scheme using contour map matching to detect composite event. They observed that spatio-temporal patterns can be used for sensed data, and using contour map matching, pattern of sensed data and pattern of event can be compared.

Proposed contour map [5] is a topographic map. This map shows the division of data values in the deployed network. The deployed area is divided into small parts. Each part contains nodes with similar readings. These

divided areas are called contour regions and the boundaries of these area are called contour lines. A snapshot of a contour map was defined by authors, as shown in Fig. 2. Where, 2 contour areas with different colors shows different readings for temperature.

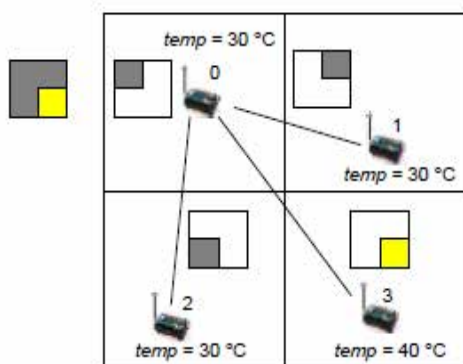


Fig. 2. Contour mapping on a 2x2 grid for temperature readings [5]

In this work, the authors have proposed a technique to make and update contour maps for 2-D number in wireless sensor network. Then, blocks are built using these contour maps. When, defined pattern for event matches with current snapshots, then event is detected.

The assumption of pre-describing capability of event pattern over area and variations with time was the major limitation of the protocol.

Solis and Obrazcka [6] present a solution to produce maps for continuous monitoring of area considering different applications such as temperature, rainfall etc. They produced a contour map with sensed attributes. In this scheme, nodes are grouped with similar data, called isoclusters. In the isoclusters, line which connect nodes with same reading is called isoline. These isolines are discovered using Neighbor-to-Neighbor Protocol (NNP) via collecting knowledge of neighbors.

NNP protocol sends a message with its sensed data, identity and distance to the base station (i.e., hop number) to all neighbors. After receiving knowledge about its neighbors, a node compares its sensed reading with the reading of all neighboring nodes. If the sensed readings is around the different sides of an isoline, then a node will send event report to the base station.

A 3D map and aggregation method for event detection was developed by M. Li et al. [7]. The authors assumed that nodes are located in 3D-environment and therefore modeled the 3D environment in form of a cubic map cell.

Each node is in charge for the environment sensing inside the unit cubic cell. These cubic cells with similar readings can then be aggregated, which form the gradient data map at every interval time for the deployed network. There are different clusters in the gradient map. For each

sampling process, the partial gradient data maps from the sensor readings are created with data aggregation.

Then, these partial gradient maps are aggregated on the way to the base station. Finally, base station built gradient map from partial gradient map and detect event.

### C. Fuzzy-based event detection protocols

Fuzzy-based techniques use fuzzy logic system (FLS) for event detection in WSNs. Fig. 3. Illustrates the basic diagram of a fuzzy logic system (FLS). This system consists 3 parts: (i) fuzzifier,(ii) inference engine, and (iii) defuzzifier.

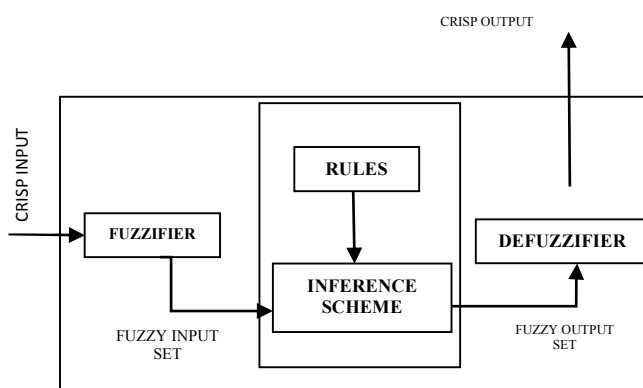


Fig. 3. Fuzzy logic system structure

The fuzzifier apply corresponding membership functions the crisp data (numbers) to convert it in fuzzy sets. These membership functions have different shapes. Triangular, trapezoidal, piecewise linear and Gaussian [8] are the most common shapes for membership functions, which depend on domain knowledge or using different learning techniques such as neural network etc. Experts describes rules derived from domain knowledge in the form of all possible combinations of values. Using conditional statements (if-then) based on these rules, fuzzified data are processed. Then the input fuzzy sets are mapped to output fuzzy sets in inference.

Finally, the crisp data is derived from fuzzy sets in the defuzzifier. Different defuzzifier are discussed in the literature [8], such as Maximum defuzzifier, mean of maxima defuzzifier, centroid defuzzifier, etc.

Liang, Q. et al. [8] have proposed two approaches for event detection using fuzzy logic system as (i) double sliding window, and (ii) hybrid event detection. The authors used Berkeley MICA2 motes [8] to evaluate the event-detection from the acoustic data collected by conducting different experiments on the test bed.

Amplitude of sound is measured using an acoustic amplitude sensor and data was collected using MOTE-Kit.

(i) Double sliding window: in this approach, two successive sliding windows of the sensed signal energy are calculated. Event is detected based on the ratio of the total

energy contained within the two successive windows as illustrated in Fig. 4.

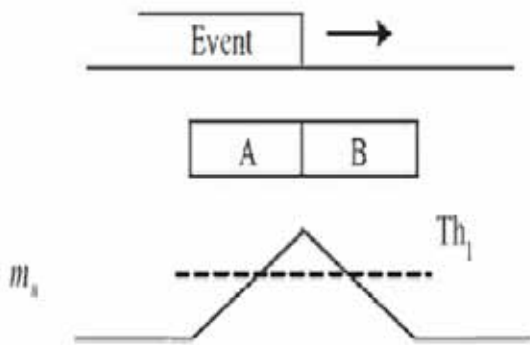


Fig. 4. Double slide window approach for event detection [8]

The windows referred in Fig. 4 are estimated by following set of equations.

$$E_A = \sum_{m=0}^{M-1} |z_{n-m}|^2 \quad \text{..... (I)}$$

$$E_B = \sum_{m=1}^M |z_{n+m}|^2 \quad \text{..... (II)}$$

Where, M is the number of rules.

The decision variable  $m_n$  is evaluated by

$$m_n = \frac{E_A}{E_B} \quad \text{..... (III)}$$

The better aspect of this was that the decision of event depends on the ratio of the energy of two successive windows instead of current sensed data. However, if an event occurs continuously in sensing area of a node, ratio will still be flat. Thus, the probability of event detection will decrease accordingly.

To overcome this problem, the authors presented a hybrid event-detection approach based on fuzzy logic system. They proposed a fuzzy logic system:  $R^l$ : IF  $E_s$  is  $F_1^l$  and  $m_n$  is  $F_2^l$ , THEN the possibility that there is event ( $y$ ) is  $G^l$ . ( $l = 1, 2, \dots, 9$ ).

They have used two inputs for the FLS: the accumulated signal energy  $E_s$  in a fixed time interval and ratio of the accumulated signal energy in two consecutive sliding windows  $m_n$ . The possibility that an event occurs was considered in five levels: (i) very strong, (ii) strong, (iii) medium, (iv) weak and (v) very weak. They have utilized trapezoidal membership functions. Table 1 describes different rules to detect an event

Rule	$E_s$	$m_n$	Consequent
1	low	low	very weak
2	low	mod	weak
3	low	high	mod
4	mod	low	weak
5	mod	mod	mod
6	mod	high	strong
7	high	low	mod
8	high	mod	strong
9	high	high	very strong

TABLE I  
RULES FOR EVENT DETECTION

At the last, height defuzzification method was used to determine crisp values.

Although the base work was based on fuzzy logic, the accuracy of the results was determined by simulations.

Marin-Perianu, M. et al. [9] have presented a general-purpose reasoning engine for wireless sensor networks (D-FLER). Residential fire detection application was studied by applying two types of sensor nodes i.e., smoke and temperature sensors.

D-FLER used fuzzy logic for combining current sensed data of a node and its neighbors sensed data, and determine if an event has occurred or not. Fig. 5. represents the D-FLER system.

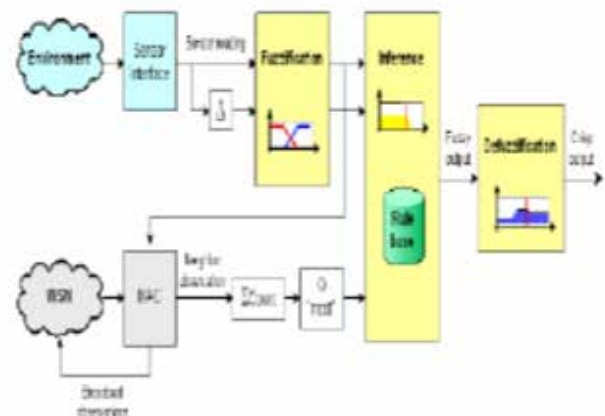


Fig. 5. D-FLER structure [9]

Following procedure was adopted by the authors.

**1. Fuzzification:** The raw values obtained from sensor interface, and differential variations of sensed values are fuzzified using triangular membership functions. Then, nodes send their fuzzified results to their neighbors by the MAC layer.

**2. Quantification:** The sigma-count factor and  $\mu_{most}$  operator are used to process neighbor's data.

$$\sum Count(F) = \sum_i \mu_F(x_i) \quad \text{..... (IV)}$$

Where, F is a feature of interest associated to the observations (such as "smoke level is high") and  $X = x_1, \dots, x_n$  is the set of neighbors.

A fuzzy majority quantifier was used to classify the neighbors's sensed data, e.g., most

$$\mu_{most} \left( \frac{\sum Count(F)}{|X|} \right) = \mu_{most} \left( \frac{\sum_i \mu_F(x_i)}{n} \right) \dots \dots \dots (V)$$

$$\text{Where, } \mu_{most}(x) = \begin{cases} 0 & \text{if } x \leq 0.3 \\ 2x - 0 & \text{if } 0.3 < x < 0.8 \\ 1 & \text{if } x \geq 0.8. \end{cases}$$

**3. Inference:** Input fuzzy sets are mapped to output fuzzy sets in inference using following rules:

IF  $s_1$  is  $F_{11}$  AND  $s_2$  is  $F_{12}$  AND ...  $s_p$  is  $F_{1p}$  AND  
 $Q_{n1}$  is  $F_{11}$  AND  $Q_{n2}$  is  $F_{12}$  AND ...  $Q_{np}$  is  $F_{1p}$   
 THEN  $o$  is  $G$

Where:

$s$  : Sensor's fuzzified data

$n$  : Neighbor's observations

$Q$ : majority quantifier

$F_{ik}$  : Input fuzzy sets

$G$ : Output fuzzy sets

**4. Defuzzification:** In the presented scheme, centroid was used for defuzzification of the fuzzified data.

Through simulations the authors showed that presented system provides better event detection accuracy. However, transmission of sensed data to all neighbors in 1-hop causes high communication cost. The proposed approach does not use any temporal semantic.

Manjunatha, P. et al. [10] proposed a scheme with fusion of sensed data from multiple sensors. Fig. 6. illustrates the diagram of fuzzy logic system with multi-sensors.

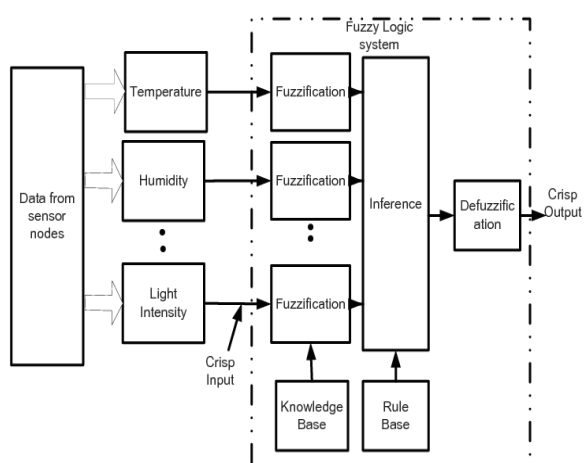


Fig. 6. Diagram of a fuzzy logic system with multi-sensors [10]

In the proposed approach, the authors considered a clustered wireless sensor network, in which each cluster has one cluster head. All cluster members (ordinary sensor

nodes) collect the environmental information i.e., temperature, humidity, CO, and light intensity and transmit to the cluster head. In the proposed method, Monte Carlo simulation was used to generate sensor data. The cluster head estimates mean of each variable and send to fuzzification. In the fuzzification, crisp inputs are fuzzified into fuzzy sets through the predefined membership functions.

The fuzzy sets LOW, MEDIUM and HIGH are defined on each input variable as trapezoidal, triangular, and trapezoidal membership functions respectively. After fuzzification, the inference engine refers to the fuzzy rule base containing fuzzy rules (IF antecedent THEN consequent) to derive the fuzzy outputs. The most commonly used fuzzy inference technique is Mamdani method. The Fuzzy if-then rules in expert system are usually in the following form of:

IF  $x_1$  is  $A_{11}$  and  $x_2$  is  $A_{21}$  ... THEN  $y$  is  $B_1$   
 else IF  $x_1$  is  $A_{12}$  and  $x_2$  is  $A_{22}$  ... THEN  $y$  is  $B_2$   
 ...  
 IF  $x_1$  is  $A_{1k}$  and  $x_2$  is  $A_{2k}$  ... THEN  $y$  is  $B_k$

where  $x_1, x_2 \dots$  are the fuzzy input(antecedent) variables  $y$  is a single output(consequent) variable and  $A_{11} \dots A_{1k}$  are the fuzzy sets.

In the proposed scheme, Four input variables are used (i.e., temperature, humidity, CO, and light intensity). Three fuzzy linguistic variables (i.e., LOW, MED, HIGH) are assigned for each input variables. Hence, there are total eighty one rules (i.e., all possible combinations for input variable). In the paper, authors describe few example rules, such as:

IF **Temperature** is **low** and **humidity** is **high** and **light intensity** is **low** and **CO** is **low**  
 THEN **Fire** probability is **low**.  
 IF **Temperature** is **medium** and **humidity** is **low** and **light intensity** is **high** and **CO** is **high**  
 THEN **Fire** probability is **high**.

Fuzzy Inference System (FIS) editor with Matlab Fuzzy Toolbox is used to generate these rules.

At last, the defuzzification of fuzzy outputs into crisp values is done using centroid.

The proposed approach improves the event detection accuracy via processing data from multiple sensors. However, the authors do not address false event detection issue. Moreover, the proposed approach does not consider time and location for composite event detection.

Kapitanova, K. et al. [11] proposed an approach for robust event detection in WSNs using fuzzy logic. In the proposed approach, the authors considered a fire detecting scenario in a building. It will activate an alarm on the fire



occurrence. A number of sensors were deployed in each room, as well as in the hallways. Sensor nodes and a master node are deployed in the building to monitor the floors of the building. There are four variables used for fire detection: temperature (T), temperature change (DT), smoke obscuration (S), and smoke obscuration change (DS). These inputs are fuzzified into fuzzy sets through predefined membership functions. On each input variable, fuzzy sets Low (L), Medium (M), and High (H) are given as trapezoidal, triangular, and trapezoidal membership functions respectively. To further decrease the number of false alarms, they also consider temporal features of the monitored events.

In proposed approach, every node detects fire individually. Each node sends event report on the occurrence of event to the master node. The proposed approach enhances event detection's accuracy considerably.

Xingming, S. et al. [12] proposed an improved neighbor based fuzzy logic event detection approach. In the proposed work, the authors consider fire detection. To increase fire detection accuracy, the node first selects the appropriate neighbor using specific rules. Then, the final decision of fire is determined by the average readings of its selected neighbors in fuzzy logic system.

### III. COMPARATIVE STUDY OF EVENT DETECTION PROTOCOLS

The main features of an event detection protocol are illustrated in fig. 7. These features are considered while developing event detection protocol for any application.

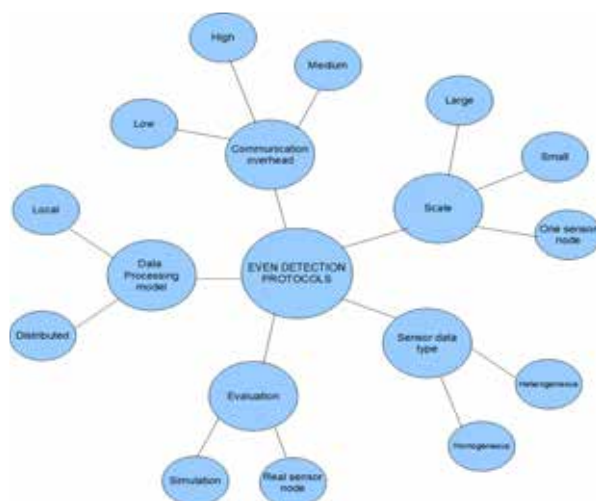


Fig.7. Feature of event detection

In this section a comparative study of the various event detection protocols reported in the literature is presented in the light of the features as identified in the above figure. Initially an explanation of the features is presented and then the comparison is given in tabular form (Table II).

#### A. Data processing model for event detection

Data processing model for event detection defines how and at what location, data is processed for event detection. Broadly speaking, for event detection data is processed either locally (i.e., data is processed at the sensors individually to detect events without getting information from other nodes) or distributed (i.e., events are detected at base station or intermediate node through information exchange between sensor nodes).

#### B. Scale

Sensor network scale define the number of sensors involved for event detection, i.e., large scale (thousands sensors are used for event detection), small (hundreds sensors are used for event detection), and single sensor (the entire event detection is done on one sensor).

#### C. Sensor data

Events may be detected based on homogeneous or heterogeneous sensor data. More sensor data type increases data dimensionality and require more sophisticated event detection approaches.

#### D. Communication overhead

The number of messages generated in the WSN. To evaluate communication overhead, we use High and Low. The communication overhead of protocols is High, if data is periodically transmitted by each node to the base station or intermediate nodes for event detection. The communication overhead of protocols is low, if the occurrence of event is transmitted by node.

#### E. Evaluations of event detection protocols

Two methods are used to evaluate: (i) through simulation where computer simulators are used to predict the occurrence of event, and (ii) through real implementation where real sensors are used.

TABLE II  
COMPARISON OF EVENT DETECTION PROTOCOL

Protocols	Technique for event detection	Data Process model	Scale	Sensor data	Communication overhead	Evaluation
[1]	T	D	L	Hetero	H	Sim
[2]	T	D	L	Hetero	H	Sim
[3]	T	D	L	Hetero	H	Sim
[4]	T	D	L	Hetero	H	Sim
[5]	P	D	L	Hetero	H	Sim
[6]	P	D	L	Hetero	H	Sim
[7]	P	D	L	Hetero	H	Sim
[8]	F	D	L	Hetero	H	Sim
[9]	F	D	L	Hetero	H	Sim
[10]	F	D	L	Hetero	H	Sim
[11]	F	D	L	Hetero	Low	Sim
[12]	F	D	L	Hetero	Low	Sim

T: Threshold based protocol

F: Fuzzy-based protocols

L: Large scale

H: High communication overhead

P: Pattern matching based protocols

D: Distributed manner nodes deployed

Hetero: Heterogeneous network

Low: Low communication overhead

Sim: Simulation

A lot of remarkable information about the current status of the event detection protocols can be inferred from the Table II. We can generally summarize the limitations of the existing techniques as under:

- It is challenging to select threshold value for event in threshold-base protocols.
- The major drawback of pattern-based schemes is the requirement of predefined patterns for event which can be used for exact pattern matching at the time of event detection.
- Fuzzy-base event detection protocols require a significant memory to store rule-base. The number of rules raise exponentially with number of input variables. In rule-base,  $m^n$  rules will be generated for  $n$  input variables.
- Most of the existing event detection protocols suffer from high communication overhead.
- The existing event detection protocols are evaluated through simulation (e.g., MATLAB, NS2).
- Some metrics are required to prove feasibility and applicability of event detection protocols for real sensor nodes. These metrics can be big-O calculation for simulated results, and time and energy consumption for real implementations. However, majority of existing protocols do not present such metrics.
- There is a need for studies which are mainly focused on event detection technique itself to address explicit requirements of event detection in WSNs such as energy efficiency, accuracy, and adaptability.
- Event detection protocols for WSNs should be robust to cope with inherent failure of sensor nodes.
- Event detection protocols for WSNs should be computationally cheap.

#### IV. CONCLUSIONS

A comparative study of the present event detection protocols was conducted. Limitations of each protocol was identified and presented in the section II. From the review of these protocols, it is concluded that:

1. Taxonomy of the present protocols have been proposed highlighting to main contributes of each protocol which could be considered as basis for classification.
2. There are many essential features such as data processing model, scale, sensor data, communication overhead, and evaluations of event detection protocols have not been referred in almost all the protocols and needed to be incorporated in the protocol.

#### REFERENCES

- [1] A. Phani Kumar, A. Reddy V, and D. Janakiram, "Distributed collaboration for event detection in wireless sensor networks," in *Proceedings of the 3rd international workshop on Middleware for pervasive and ad-hoc computing*. ACM, 2005, pp. 1–8.
- [2] C. Vu, R. Beyah, and Y. Li, "Composite event detection in wireless sensor networks," in *Performance, Computing, and Communications Conference, 2007. IPCCC 2007. IEEE International*. IEEE, 2007, pp. 264–271.
- [3] M. Marta, Y. Yang, and M. Cardei, "Energy-efficient composite event detection in wireless sensor networks," *Wireless Algorithms, Systems, and Applications*, pp. 10–103, 2009.
- [4] Y. Yang, A. Ambrose, and M. Cardei, "Coverage for composite event detection in wireless sensor networks," *Wireless Communications and Mobile Computing*, vol. 11, no. 8, pp. 111–1181, 2010.
- [5] W. Xue, Q. Luo, L. Chen, and Y. Liu, "Contour map matching for event detection in sensor networks," in *Proceedings of the 2006 ACM SIGMOD international conference on Management of data*. ACM, 2006, pp. 145–156.
- [6] I. Solis and K. Obraczka, "Efficient continuous mapping in sensor networks using isolines," in *Mobile and Ubiquitous Systems: Networking and Services, 2005. MobiQuitous 2005. The Second Annual International Conference on*. IEEE, 2005, pp. 325–332.
- [7] M. Li, Y. Liu, and L. Chen, "Nonthreshold-based event detection for 3d environment monitoring in sensor networks," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 20, no. 12, pp. 129–1711, 2008.
- [8] Q. Liang and L. Wang, "Event detection in wireless sensor networks using fuzzy logic system," in *Computational Intelligence for Homeland Security and Personal Safety, 2005. CIHSPS 2005. Proceedings of the 2005 IEEE International Conference on*. IEEE, 2005, pp. 52–55.
- [9] M. Marin-Perianu and P. Havinga, "D-fler—a distributed fuzzy logic engine for rule-based wireless sensor networks," *Ubiquitous Computing Systems*, pp. 5–101, 2007.
- [10] P. Manjunatha, A. Verma, and A. Srividya, "Multi-sensor data fusion in cluster based wireless sensor networks using fuzzy logic method," in *Industrial and Information Systems, 2008. ICIIS 2008. IEEE Region 10 and the Third international Conference on*. IEEE, 2008, pp. 1–6.
- [11] K. Kapitanova, S. Son, and K. Kang, "Using fuzzy logic for robust event detection in wireless sensor networks," *Ad Hoc Networks*, vol. 10, no. 4, pp. 709–722, 2012.
- [12] Xingming, S. et al, "INFLE: An improved Neighbor-Based Fuzzy Logic Event Detection Algorithm for Wireless Sensor Networks" *IJFGCN*, vol. 7, no. 3, pp. 13–24, 2014.