International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 4 Issue 05, May-2015

EECEMP: Energy Efficient Composite Event Monitoring Protocol in Wireless Sensor Networks

D. Lissy., M.C.A., M.Phil.,
Research Scholar, Bharathiar University,
Coimbatore.

A. Yogameena, M.C. A. M. Phil.,NET.,
Asst. Professor,
Thiruthangal Nadar College,
Chennai – 51.

***Abstract*—Current trend in Wireless Communications technology needs the development of multifunctional Wireless Sensors which minimize cost, energy and maximize reliability, scalability, flexibility. The purpose of WSNs is collecting relevant and accurate data for processing and reporting. The data reporting in WSN is based on either time-driven or event-driven. The event-driven scheme must be able to more accurately analyze the environment to detect critical events and it should disseminate the data with reduced energy consumption. This is possible by dynamically switching between time-driven and event-driven data reporting schemes with the help of neighbor's information. EECEMP concentrates on composite event monitoring by getting the local decision of each node in case of critical event by discussing with its neighbor's, route the decision finally to the sink and they switches between time–driven reporting scheme and event driven to make the system an energy efficient one.**

Index Terms – Composite Event, EWSN, Energy efficient, Event-Driven, Time-Driven, Routing.

I. INTRODUCTION

Wireless Sensor Networks is a collection of wireless nodes with limited energy capabilities that may be mobile or stationary and are located randomly on an environment. A special class of wireless sensor networks [1], event-driven wireless sensor networks (EWSNs) are composed of large numbers of sensor nodes that are deployed in the terrain to sense physical changes in the environment. Thus, EWSNs will have the capability to transmit the sensor data, including critical data (e.g., the location of an event), to one or more centralized sinks who are expected to perform real time processing and to make accurate decision quickly[2]. The main purpose of deploying a EWSN is to collect relevant data for processing and reporting to a base station or sink. The drawbacks of most of the existing methods are that less energy level (recovery of a dead node) is not considered, and the amount of the exchange data may be huge. In this paper, we propose an energy efficient composite event monitoring protocol to monitor infrequent events and save a node by itself in a critical situation. We compress the sensed event raw data by analyzing the characteristics of event(time-driven or event-driven), thus the event is detected in a distributed and efficient way. Also, the final decision in the form of alarm messages will be delivered to the sink in an energy efficient and fast manner.

Consider an abnormal nuclear fission event in a Nuclear Atomic Plant. As soon as there is change in fission rate followed by an increase in temperature, it is immediately reported to an observer to rapidly introduce moderator rods. By continuously receiving abnormal change information like radiation and/or temperature from the sensor nodes and its neighbors where no abnormalities has yet been detected but may soon occur, an observer is able to accurately identify where the radiation is leaked out, forecast where it may be heading, and determine where the radiation leak out should be avoided. Also the node within the event should protect by itself by moving from that place to another place.

The contributions of this paper are as follows. In Section I, we have proposed a composite event monitoring protocol that allows a WSN to detect a critical event based on more than one event attribute. In Section II the system dynamically switches between the time-driven and event-driven data reporting scheme based on node context to reduce energy consumption. Thus, it is able to more accurately analyze environments than in the previous data-reporting schemes. To minimize energy the nodes remain more in time-driven until an event is detected. In Section III, the proposed protocol differs from those of previous works in that the node checks by itself whether it is under a critical condition or failure state. If so, it adapts a self-healing data-gathering process and changes it location to the nearest neighbor from where the decision about critical events is taken place.

II. RELATED WORKS

Conventional Event Monitoring protocols in WSNs have less focus on accurate event detection and also lead to wastage of sensor resources. ESRT (Event-Sink Reliable Transport)[5] provides a solution for reliable event detection by proper congestion control protocol based on rate adjustment. But it treats both sensitive and insensitive areas equally because all sensor nodes are controlled in the same time [6], [7], [8] using the conventional distributed detection theory. Lige Yu investigated [6] three detection schemes: the centralized, distributed and quantized scheme. A centralized scheme requires each sensor node to forward all its observations to the fusion center, which results in large energy in communication. In distributed scheme, each sensor node makes its own decision and then sends out only its 1- bit decision. This reduces communication energy at the cost of increased processing energy and reduced detection accuracy. In a quantized scheme, each node processes observations data and sends a quantized M-bit quantity to the fusion center, and the control center makes the final decision based on the K quantized quantities.

Considering a sequential detection strategy, a sequential probability ratio test (SPRT) at the fusion center is adopted. Based on the work of Lige Yu suggested an energy-driven scheme where each sensor node sends out its 1-bit decision if that decision exceeds a predetermined detection accuracy threshold in, and sends out all its observations otherwise. The scheme sets a limit for the maximum number of observations collected by each sensor, which avoids the potential delay at sensors, but the scheme caused large number of observations in the case of sequential detection. In a simple event detection protocol was proposed for a line-deployed wireless sensor network configuration to detect fire. Depending on environment the system was categorized into global phenomenon (GP) and local phenomenon (LP). To minimize the power consumptions, each node makes a local decision considering its own observation and also the decision made by the previous node. To minimize the error probability, a simple fusion rules for both the global and local phenomenon scenario are proposed. However, the chain network, which is not scalable for more complex network topology. In WSNs, an event should be decided by combining the properties of events together.

Composite event is first proposed to make accurate event detection. To ensure the quality of surveillance, some applications require that if an event occurs, it needs to be detected by at least *k* sensors, where *k* is a user-defined parameter. Authors examine the Timely Energy-efficient k-Watching Event Monitoring (TEKWEM) problem and propose a detection model and a warning delivery model to monitor the composite adjustment. However, detection set and a *(α, β)*-Light Approximate Shortest Path Tree should be established which is not suitable in the surroundings with low bandwidth and limited computation ability. Greedy Perimeter Stateless Routing (GPSR)is a very well-known routing protocol utilizing the positions of nodes and the destination to make packet forwarding decisions. However, GPSR is designed for the ad-hoc networks and does consider the characteristics in EWSNs, where the event decision should be made in the forwarding procedure to shorten the event information delivery delay. SPEED protocol is designed to provide end-to-end deadline guarantees for real-time packets in sensor networks, which is very suitable for emergency information dissemination in EWSNs [2]. However, the SPEED protocol provides only one network-wide speed, which is not suitable for differentiating various traffics with different deadlines. In addition, it is limited to provide any guarantee in the reliability domain.

A typical WSN consists of hundreds to thousands of sensor nodes deployed over an area; the dense deployment of sensor nodes leads to high correlation of the data sensed by the neighboring nodes. Hence, the main idea of data aggregation and in-network processing is to combine the data from different nodes en route by eliminating redundancy, thereby minimizing the number of transmissions. This process ultimately saves energy and prolongs the network lifetime.

With regard to the dynamic switching of data-reporting schemes, APTEEN and SINA are previous studies that are closely related to the current research. Cluster-based routing divides the entire system into distinct clusters, compresses data arriving from nodes that belong to the respective clusters, and sends an aggregated message to the base station. Challenges faced by such a clustering-based approach include how to select the cluster heads and how to organize the clusters. Many studies propose interesting algorithms and mechanisms to handle the different node role assignments.

III. PRELIMINARIES

EVENT

An event is defined to be an instantaneous, atomic (happens completely or not at all) occurrence of interest at a point in time. Events can be classified as Primitive Events, Compound Events and Composite Events. Users want to be notified immediately as soon as these complex events are detected. Sensor node devices generate massive sensor data streams. These streams generate a variety of primitive events continuously. The continuous events form a sequence of primitive events, and recognition of the sequence supplies us a high level event, which the users are interested in[9].The detection of a composite event may require the detection of one or more constituent events as well as one or more occurrences of a constituent event. Events requiring multiple event occurrences (either of the same type or of different types) for the detection of a composite event, give rise to alternate ways of computing the history as well as parameters, as the events are likely to occur several times over an interval.

EVENT OPERATORS

An event type is denoted by an event expression. A primitive event (type-name) itself is an event expression. If E_1, E_2, E_3. . . , E_n,, are event expressions, an application of any event operator, described below, over the event expressions is an event expression. For an event E, begin-of E, end-of E, and (E) are all event expressions. If event modifier is omitted, end-of is assumed by default. The operator semantics described below assumes the end-of modifier.

An event E (either primitive or composite) is a function from the time domain onto the Boolean values, (True and False)

E: T → {True, False}

Given by

E (t) = True if an event of type E occurs at time point t False

Otherwise,

~E denotes the negation of E which computes the non-occurrence of an event at that point.

Some of the event operators applied for composite events is specified here.

i. OR (▼): Disjunction of two events E_1 and E_2, denoted by E1▼E2, occur3 when E_1occurs or E_2 occurs.
ii. AND (Δ): Conjunction of two events $E_1$and E_2, denoted by E1 Δ E2, occurs when both E_1 and E_2 occurs irrespective of their order.
iii. ANY: The conjunction event, denoted by ANY (m, E_1, E_2, En) where $m \leq n$, occurs when m events out of the n distinct events specified occur, ignoring the relative order of their occurrence.
iv. SEQ (;): Sequence of two events E_1 and E_2, denoted E_1; E_2, occurs when E_2 occurs provided E_1 has already

occurred. This implies that the time of occurrence of E_1 is guaranteed to be less than the time of occurrence of E2.

v. Periodic Event Operators (P, P*): A periodic event is a temporal event that occurs periodically. A periodic event is denoted as P (E_1, TI [: parameters], E_2) where E_1 and E_2 are events and TI [: parameters] is a time interval specification with optional parameter list. P occurs for every TI interval, starting after E_1 and ceasing after E_2. Parameters specified are collected each time P occurs. If not specified, the occurrence time of P is collected by default.

COMPOSITE EVENTS

We can formally express the occurrence of a composite event E with respect to its constituent events that form part of the occurrence of E. At some level, the constituent events are primitive events. We denote an occurrence of an event type E_j by $\boldsymbol{e_j^i}$ where i indicate the relative time of occurrence with respect to other occurrences of the same event. Composite events are represented as a set of constituent event occurrences within which the order of event occurrencesis preserved. Note that it is possible for the same constituent event occurrence to be used for more than one occurrence of a composite event. The last event in the set is one whose occurrence made the composite event occur. The time of occurrence of a composite event is the time of occurrence of the last constituent event.

Global Event History/Event Log is a set of all primitive event occurrences and is denoted by H. Each primitive event occurrence is represented as a singleton set in the log.

H = {{$\boldsymbol{e_j^i}$} | for all j, primitive event e^i has occurred at instance i relative to events Ej }.

Primitive Event History/Event Log of the primitive event type E is a set of the occurrences of E present in the Global History H and is denoted by E [H].

E_j[H] = { {$\boldsymbol{e_j^i}$} | for all i, {$\boldsymbol{e_j^i}$} ∈ H }.

Composite Event History/Event Log of a composite event E that has n constituent events E_1, E2,……., En is a mapping from the global event history H to a subset of E_1[H] ∪ … ∪E_n[H] where ∪ is an operator that computes the cross product of two sets (whose elements are sets) and merges the elements of the cross product using the union operator. For example, given-event histories E_l[H] = {{$\boldsymbol{e_1^1}$,$\boldsymbol{e_3^4}$}, {$\boldsymbol{e_1^2}$} } and E_2[H] = {{$\boldsymbol{e_2^1}$}, {$\boldsymbol{e_2^2}$} }

Where, E_1[H] ∪ E_2[H]= {{$\boldsymbol{e_1^1}$,$\boldsymbol{e_3^4}$,$\boldsymbol{e_2^1}$},{$\boldsymbol{e_1^1}$,$\boldsymbol{e_3^4}$,$\boldsymbol{e_2^2}$},{$\boldsymbol{e_1^2}$, $\boldsymbol{e_2^1}$},{$\boldsymbol{e_1^2}$, $\boldsymbol{e_2^2}$}}

Event Collection is a collection of all event occurrences of a particular type within a specified time interval. It is denoted by function ρ as follows.

ρ(E, start-time, end.time) = {e | {e} ∈E [H]and start-time ≤ t-occ (e) ≤ end-time}.

IV. PROTOCOL DESIGN

The protocol is designed as 3 Sections.

SECTION 1: Detection of Composite Event.

SECTION 2: Dynamic Switching between Time-Driven data reporting scheme and Event-Driven Scheme to minimize energy.

SECTION 3: Self-Healing Scheme in case of node failure or if the node is under critical position.

SECTION 1:DETECTION OF COMPOSITE EVENT

A number of nodes with different sensors are used to detect a composite event (Fig.1). Each node maintains event histories data with its neighbors and makes its own decision. An alarm message is sent to the sink detecting the type of event. Self-healing technique is used to ensure the reliable and timely transmission of the event alarm by choosing the nearest neighbor in case of node failure to send the alarm message. The detection of a composite event may require the detection of one or more constituent events as well as one or more occurrences of a constituent event. Events requiring multiple event occurrences (either of the same type or of different types) for the detection of a composite event, give rise to alternate ways of computing the history as well as parameters, as the events are likely to occur several times over an interval.

The occurrence of a composite event is marked by the occurrence of a constituent event that makes the composite event occur (using the end-of event modifier semantics). In Fig. 1 this constituent event is termed the **terminator** of the composite event. Several constituent events can act as terminators, but there is at least one terminator event for a given composite event. Analogously, there is always a constituent event that initiates the occurrence of a composite event. This constituent event is termed the **initiator** of the composite event. There may be more than one initiators for a composite event. For a primitive event, the primitive event itself is the terminator and initiator. The composite event detector needs to record the occurrences of each constituent event and save its parameters so that they can be used to compute the parameter set of the composite event.

It is preferable to perform In-Network Processing inside sensor network to reduce the transmission cost between neighboring nodes. For normal sensor data processing, the centralized approach proceeds in two steps; the sensor data captured by all the sensor nodes is sent to the sink node and then routed to the central server (Base Station/Global Sink) where it is stored in centralized database. High volume data are arriving at the server. Subsequently, query processing takes place on this database by running queries against stored data. Each query executes one time and returns a set of results [9].

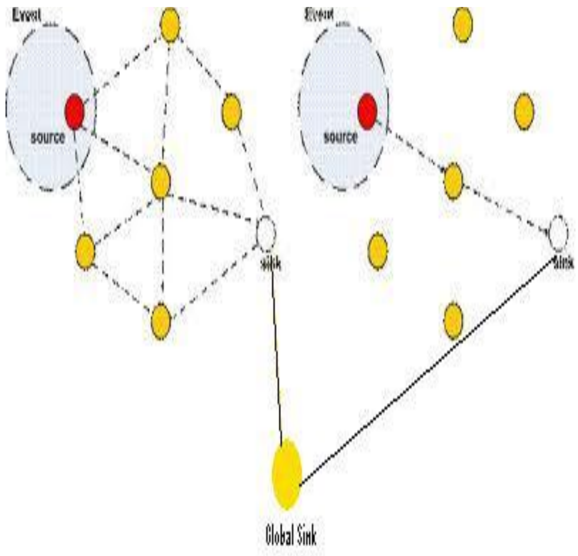

Fig. 1. Event Detection in WSNs

SECTION 2: DYNAMIC SWITCHING BETWEEN TIME-DRIVEN DATA REPORTING SCHEME AND EVENT-DRIVEN SCHEME

The scheme maintains 2 threshold values (**m**: for making time-driven, **n**: for making event-driven) for each event attribute. The sensors of each node monitor the value of its event parameters. If the value crosses the threshold value m, it signals its primitive event occurrence to its neighbors and takes the decision based on the event history and finally reports it to the sink. The neighborhood information can be determined by the number of hops the node should take to disseminate the data. A variable TTL (Time To Live) can be used to set the number of hops. For each hopping it decreases the variable TTL. If the value of Event parameters falls below n or if TTL = 0 it start working in Event-driven mode.

SECTION 3: SELF HEALING SCHEME

In this section we provide a way to self-heal the node and to communicate with process of sending alarm message in case of failure or under critical environment. Each node copies and saves the message sent to it and immediately it finds a neighbor and send all the information to its neighbor node and keep moving from that place to another place .Even though it is impossible as soon as the node receives an event-driven event it has to check whether the node is in a critical state. Let us consider the Node under critical condition as N_C and it neighboring Node as N_i. If node N_i receives an alarm from node N_C, and does not finds a data transmission from the node N_C, it finds a link failure. It starts to extract a copy of alarm and forwards to next neighbor.

V. ALGORITHM FOR EECEMP:

Step 1: Choose a Global Sink node S_G to store all the sensor data captured by all sensor nodes and with the help of Event Graph database datas are stored in a centralized database.

Step 2: Set a Global Event History, Primitive Event History and Composite Event History for all nodes.

Step 3: Set the application rule for determining Composite Event in the Global Sink

Step 4: Create an Event Tree for each Composite Event by inserting primitive event occurrences as the leaves.

Step 5: Merge Event Trees into an Event Graph.

Step 6: During Event Monitoring, linked list of Event Attributes is computed and passed to Application Rule.

Step 7: Choose a local sink node S_l for each area to detect local event to which all primitive events are signaled.

Step 8: Set the Application Rule for each sensor nodes (like establishing Threshold Value[m and n] for Event Attributes) to switch between time driven and event driven mode.

Step 9: If the Rule is satisfied, the node switches itself to Time-Driven Data Reporting Scheme and passes it computed value to Local Sink. If the rule is not satisfied it switches to Event-Driven thus saves energy and resources.

Step 10: Global Sink uses Event Graph to find the occurrence of Composite Event.

Step 11: Local sensor nodes informs about the occurrence of Local primitive Event to the local sink which then reports it to the global sink in the form of an alarm message.

Step 12: The alarm message is copied to the nearest neighbors as per the specified number of hop counts (TTL value)

Step 13: If the nearest neighbor finds any node failure it starts extracting its copy of message and sends the message to its neighbor nodes.

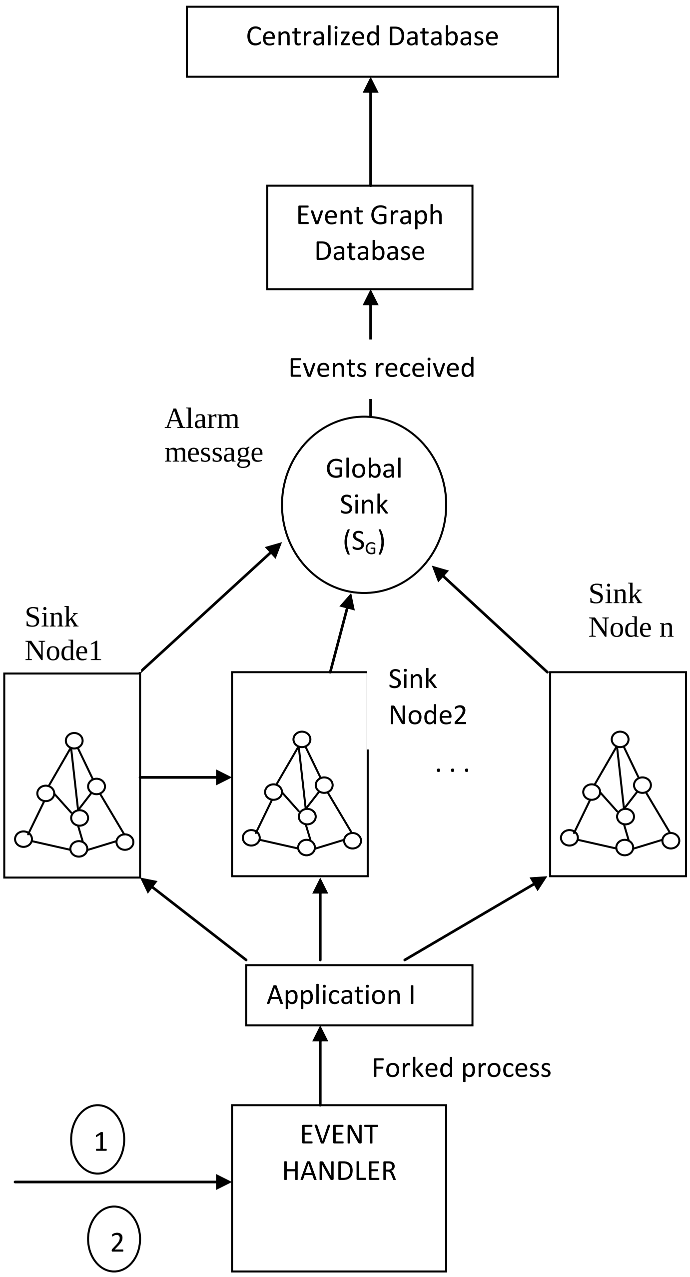

Fig.2. EECEMP.

Consider the Sensor nodes as shown in the above fig.2 it is clearly understood that the node switch between events. Here 1 indicates a time-driven event and 2 indicates a Event driven data. Whenever the nodes doesn't receive a Event-driven data it remain in time-driven event to save energy and the nodes protect by itself when there is a critical condition. A Global Sink receives the alarm message.

VI. CONCLUSION

In this paper, we proposed an Energy Efficient Composite Event Monitoring Protocol which dynamically switches between the time-driven data reporting scheme and the event-driven data-reporting scheme.

The essential elements of the protocol were that:

1) Detection of Composite Events which reduces overhead and saves storage
2) It switches between two data-reporting schemes to acquire as much information as possible while consuming only less amount of energy.
3) It uses self-healing scheme to safe guard the alarm message.
4) It allows sensor nodes continuous data dissemination, thus enabling accurate analysis of future behaviors of the environment being monitored. To make it possible it follows a timely manner (when to switch between the data reporting schemes, and which sensor nodes to involve in the time-driven data-reporting process).
5) We proposed an event tree for each composite event and these trees are merged to form an event graph for detecting a set of composite events. This will avoid the detection of common sub-events multiple times there by reducing storage requirements.
6) The Event Tree allows us to optimize the solution as all the event detection conditions are used in one place. The combination of event-condition trees will allow conditions to be evaluated on a demand basis avoiding unnecessary computations.

VII. FUTURE WORK

Energy consumption is one of critical issues in the sensor Network. If a node consumes less energy then there is a chance of getting long life for a sensor Node. In the related work we learned about switching a node between events and to save the energy of nodes by staying most of the times in Time-driven event until and Event-driven event occurs. Our objective is to reduce energy consumption and to save a node that is under critical condition. In future this work can be enhanced by considering nodes mobility and simulation works.

REFERENCES

(1) I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Commun. Mag.*, vol. 40, no. 8, pp. 102–114, Aug. 2002.
(2) Efficient Event Detecting Protocol in Event-Driven Wireless Sensor Networks Lulu Liang, DeyunGao, Hongke Zhang, and Oliver W. W. Yang, *Senior Member, IEEE*
(3) An Energy-Efficient Hybrid Data-Gathering Protocol Based on the Dynamic Switching of Reporting Schemes in Wireless Sensor Networks Byoung-Dai Lee and Kwang-Ho Lim
(4) A. V. U. P. Kumar, A. M. V. Reddy, and D. Janakiram, "Distributed collaboration for event detection in wireless sensor networks," in *Proc.MPAC*, Grenoble, France, Nov. 2005, pp. 1–8
(5) O. B. Akan and I. F. Akyildiz, "Event-to-sink reliable transport in wireless sensor networks," IEEE/ACM Trans. Netw., vol. 13, no. 5, pp. 1003–1016, Oct. 2005.
(6) L. Yu and A. Ephremides, "Detection performance and energy efficiency trade-off in a sensor network," in Proc. Allerton Conf., Allerton, IL, Oct. 2003, pp. 390–399.
(7) L. Yu, L. Yuan, G. Qu, and A. Ephremides, "Energy-driven detection scheme with guaranteed accuracy," in Proc. IPSN, Nashville, TN, Apr. 2006, pp. 284–291.
(8) L. Yu and A. Ephremides, "Detection performance and energy efficiency of sequential detection in a sensor network," in Proc. HICSS, Jan. 2006, p. 236.
(9) Complex Event Processing in Wireless Sensor Networks OmranSaleh Faculty of Computer Science and Automation Ilmenau University of Technology Ilmenau, Germany.