7th International Conference on Recent Innovations in Science, Engineering and Management
The Institutions of Engineers, Delhi State Center (India) RISEM - 16
16th September 2016, www.conferenceworld.in ISBN : 978-93-86171-07-8

PARADIGM OF CRITICAL EVENT SENSING IN WIRELESS SENSOR NETWORK COMMUNICATION

Kawser Mohiuddin[1], Dr. K. P. Yadav[2]

[1]Research Scholar - Computer Science, OPJS University, Churu, Rajasthan, India)

[2]Director, KCC Institute of Technology and Management, Greater Noida, India)

ABSTRACT

The wireless Sensor nodes having multiple capabilities can be deployed in varying and difficult conditions to sense the data and events of various kinds. They can sense and report the sounds, humidity, pressure, gases, temperature, light, fog, speed, fire and the moments caused by vehicles, human being, animals, birds, clouds, jets, balloons etc. Its applications are enormous ranging from a playfield to battlefield, marine navigation to mars navigation, floods, mountains, oceans and everywhere.This paper provides the concept of *Embedded Critical Event Sensing paradigm* (ECSEP) for sensor network communication. Everything occurring in the environment is an event. . The sensor CPU looks into the ECSEP database for the said event. If the event is found in ECSED, it is immediately reported to the base station. It receives the varying values acquired by sensors deployed in the network as input. The acquired data can be much or less precise and are processed using CESAalgorithm. The captured event values are mostly floating point having decimal values and are converted into integer values. The corresponding variables used for storing of floating point values is declared as *float (x)*, whereas *int (y)* is the variable for storing converted integer values. The process of conversion of floating point real number values into whole number integer values is governed by certain rules and conditions This paper classifies events in to two categories, i.e. simple event and critical event. Every event occurring in the environment is not a critical event. Depending upon the requirement, certain nodes keep on monitoring the general environment, while on other hand certain nodes are programmed to sense only critical events.

I. INTRODUCTION

As the sensor nodes are deployed in the network as per requirements and necessary configurations, the sensor nodes start functioning for sensing of the data and events. However, all the events sensed are not communicated to the base station, because they are first processed for their critical nature by comparing with a database. This database contains a list of user nominated critical events. If the event sensed matches with the entry in database, the event is termed as a critical event and is sent for further communication to legitimate nodes in the network. However, if the event does not match with the event database, it is simply ignored and simultaneously removed from the memory. This paper shows the embedding of critical events to the destination. These critical events are sensed by the nodes deployed in environment and needs to be transmitted with respect to their appearance and occurrences in the environment from time to time securely to the base station.Sensing of critical events is most important and fundamental part of sensor network technology. Sensing in TinyOS (an operating system for

sensor nodes) is usually attached with ADC converters i.e. analog to digital converters. An oscilloscope collects the acquired and sensed data by sensors through ADC and its various control interfaces. Sensing is dine via two things i.e. to configure the node for sensing purpose and to acquire the sensed data.Wireless sensor networks are composed of a large number of wirelessly communicating and computing devices, nodes. A typical sensing node is usually composed of following components:

- Central Processing Unit
- Memory Unit
- Sensor Array
- Radio Transceiver
- Actuators
- Analog to Digital Converter (ADC Unit)
- GPS
- Power Supply
- Control Unit

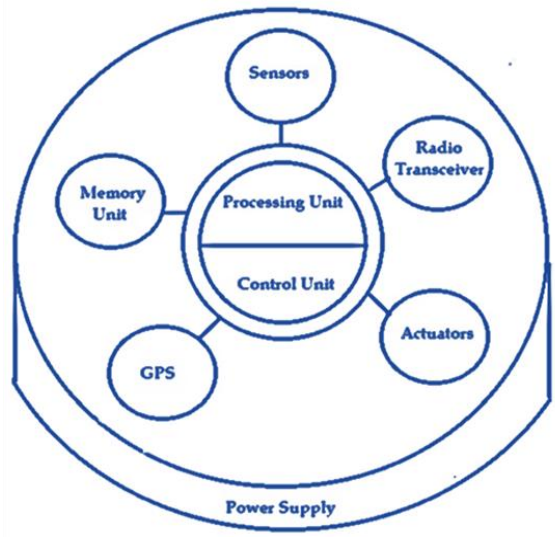

Figure 1: Components of a Sensing Node.

II. EMBEDDED CRITICAL-EVENT SENSING PARADIGM (ECSEP)

Every event occurring in the environment is not a critical event. Our work acquires all the events but communicates only critical events to the base station and simultaneously ignores the simple events. This system acquires the event, analyzes the event and then either ignores it or transmits it to the base station. The underlying algorithm maintains and drives the event sensing system as per the pre-programmed instructions. Below is given the algorithm in its simplest form that governs the transmission of critical events to the base station.As soon as

an event is triggered in the monitoring field, it is gets instantly captured by the sensor nodes deployed there in the form of a single large network. Whether the said event is a simple or critical event is decided by CESA algorithm after processing of the said event by looking up the ECSED database. The entire system works in a sophisticated manner with high precision. As the number of events is increased, the system does not experience any heavy load because of the fact that this system filters out the unnecessary and simple events and considers only critical events for communication and further processing.

```
// Algorithm for capturing and transmission of
// critical events and ignoring of simple events
1:    Event Captured
2:    Store/ Process the Event
3:    ECSED database Lookup (ECSED)
4:    Decision Making:
      If Event (E) present in ECSED,
      Report E to BS
      <Event Communicated>
      Else  Ignore the E
5:    ECSED Closed
6:    Session Closed.
```

Sensor nodes may give varying readings regarding a particular critical event and is considered for validation only after receiving the reading from multiple sensor nodes with respect to a particular event. A single witness is not enough and hence a decision is more precise when taken after taking multiple witnesses in the consideration regarding a particular event. This makes our system smart and reliable. Events captured are stored in the memory. From memory they are processed for CESED database lookup.

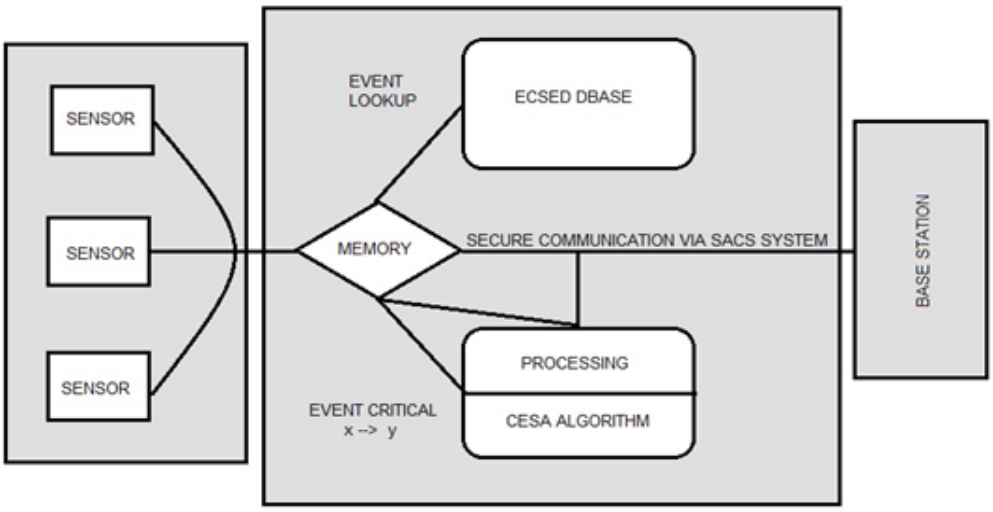

Figure 2: System Structure for Critical Event Sensing

If the event captured matches with the events available in the CESED dbase, they are termed as critical and are considered for further processing and secure communication. However if the captured event is not found in the CESED dbase, it is simply rejected and ignored by the system and freed deleted from memory locations. With the help of CESA algorithm developed for the purpose, the simple events are destroyed and only critical events are taken into consideration. The events captured are floating point values containing decimal points. Before secure transmission over SACS system, these are converted into integer values for faster and precise processing of the events. However, for the purpose of monitoring the temperature in a building and to keep the temperature range intact between certain given limits, a simple system has been demonstrated here. Without relying on a single node, four sensors have been deployed for a particular task. The building is equipped with four temperature sensors at four different locations where we want to have an average temperature range of 22 ^{o}C to 26 ^{o}C. The sensor-A relays the temperature of 19 ^{o}C, sensor-B relays 21 ^{o}C , sensor-C relays 18 ^{o}C and the sensor-D relays 22^{o}C. In order to know and maintain the average temperature in the said building, the values from all the four deployed sensors will be taken into consideration before taking any step to turn on cooling or heating systems.

$$t = \frac{(a+b+c+d)}{n}$$

Whereas T denotes the average temperature and *n* denotes the number of sensors in the building.

$$t = \frac{(19+21+18+22)}{4}$$

Therefore, the average temperature in the building is 20 ^{o}C which is below the average requirement. The data is reported to base station where from instructions are issued to keep the average temperature intact and the heating systems are switched on to raise the level of temperature up to the average requirements. Similar is the case when temperature, humidity or any other required factor is in abnormal state. Our work uses the integer values that makes it easy to evaluate the issues of critical event sensing in sensor networks. It provides best properties for describing and evaluating the sensor events. It has ability to read the inexact event readings from the sensors and is easy to use. The sensing module for critical events is shown as:

```
Configuration SenseAppC
{
}
implementation {
Components SenseC, MainC, LedsC, new TimerMilliC();
components new DemoSensorC() as Sensor;
SenseC.Boot ->MainC;
SenseC.Leds ->LedsC;
SenseC.Timer->TimerMilliC;
SenseC.Read -> Sensor;
}
```

III. DATABASE FOR EMBEDDED CRITICAL EVENT SENSING (ECSED)

This database contains a list of all the critical events required for reporting by the monitoring team. This database is a customized and can be designed as per the user requirements. The sensor CPU looks into the ECSEP database for the said event. If the event is found in ECSED, it is immediately reported to the base station and sometimes to the nearest neighboring nodes also. However the said event is simply ignored and erased from memory if it does not match with the list of particular critical events in the ECSEP database. As our design is much flexible and can be applied and used by any person or organization working in any kind of domain. Before using the critical event sensing technology, one needs to enlist and feed all the required critical events in the ECSED database, so that unnecessary events and actions in the monitoring field are ignored and filtered out by the sensor network. The possible list of critical events in ECSEP database varies from people to people and organization to organization. Following is the list possible critical events for monitoring in different kinds of domains and fields:

- **Agriculture Monitoring:**

  Abnormal temperature, values
  Abnormal humidity, values
  Abnormal rainfall, values
  Abnormal winds, values
  Intrusion of animals or pests in the field, values
  Fire in the field, values
  Abnormal water levels in the field, values
  Excessive falling of leaves, crops or fruits, values

- **Monitoring in Educational Institutions:**

  Entry of outdoors in the college premises, values
  Late entry by staff or students, values
  Appearance of fire or smoke, values
  Jumping over boundary walls, values
  Jumping through windows, values
  Stealing lab equipment or other listed material, values
  Sleeping in classrooms, values
  Fight or vandalizing things in the college, values

- **City Traffic Monitoring:**

  Wrong parking, values
  Abnormal vehicle speed, values
  Vehicle accident, values
  Wrong side driving, values
  Driving over public paths, values
  Disobeying traffic signals, values

- **Military Monitoring at Country Border:**

  Invasion by enemy, values
  Suspicious or unauthorized movements by various living or non-living bodies on any side of the border, values
  Attacks from enemy side, values
  Absence of soldiers from a particular guarding post, values
  Touching or jumping over border walls, values
  Appearance of huge fire or smoke, values
  Abnormal weather, values

However, there are thousands of various other domains that have their own particular demands and requirements according to their monitoring needs. The events are sensed and acquired by sensors as a chain of values that specify the parameters to characterize a particular event. For instance if we want to know the temperature of a particular place in the monitoring area. The sensors deployed close to each other will have different values than those of located at far in the same network.

IV. CRITICAL EVENT SENSING ALGORITHM (CESA)

Our work uses CESA algorithm developed for recognizing of critical events. It receives the varying values acquired by sensors deployed in the network as input. The acquired can be much or less precise and are processed using *if-then* statements in the algorithm. The captured event values are mostly floating point having decimal values and are converted into integer values. The corresponding variables used for storing of floating point values is declared as *float (x)*, whereas *int (y)* is the variable for storing converted integer values. The process of conversion of floating point real number values into whole number integer values is governed by certain rules and conditions including *if-then* statements. Consider *x* as an acquired value by the sensor node. This input is turned into *int* value *y* using *if-then* conditions. At the processing stage of algorithm, the *float* values are converted into *int* values using a simple converting algorithm. This algorithm retrieves the captured event values from the *float (x)* variable and after applying certain rules these values are converted into *integer*

values. For example x=15.85 is the captured value of an event in x variable. This values is converted into y=16 as an integer value for processing and communication. These values make the processing task easy and simple. However, these values are considered for conversion only after they are classified as critical events after looking into the ECSED dbase. The critical event monitoring in a WSN nodes is usually set with passive event detection capability and it allow a node to detect an event even when its wireless communication module is in sleep mode. Upon the detection of an event by the sensor, the radio part of the sensor node is immediately woken up and is ready to send an alarm message. The CESA algorithm and its various steps and procedures from capturing of an event to its transmission to base station is represented here respectively.

```
// CESA algorithm for capturing and transmission of
// critical events and ignoring of simple events
1:    Event captured
2:    Event stored
3:    ECSED lookup
// Events are looked first for their critical nature
4:    Decision making:
      IF event (E) present in ECSED,
      (E = x)
      THEN convert x into y
      ELSE IF event (E) absent in ECSED,
      THEN remove x from memory
// Where x represents the floating point event value
// and Y represents the critical event integer value.
5:    ECSED closed
6:    Event processed/ stored
      (E = y)
7:    Event (y) transmitted to BS
// using secure and authenticated communication system
// all the event messages are transmitted to base station
8:    Session closed
```

The algorithm provides an understanding about the overall functioning of the critical event detection system. However, the conversion of floating point values containing decimal points into integer values for secure communication over the network is the core of this algorithm. Without this module, the entire sensing paradigm is of no use. This module gives the system a uniqueness that makes authorized communication easier and secure. The process of sensing events is described in the following figure.

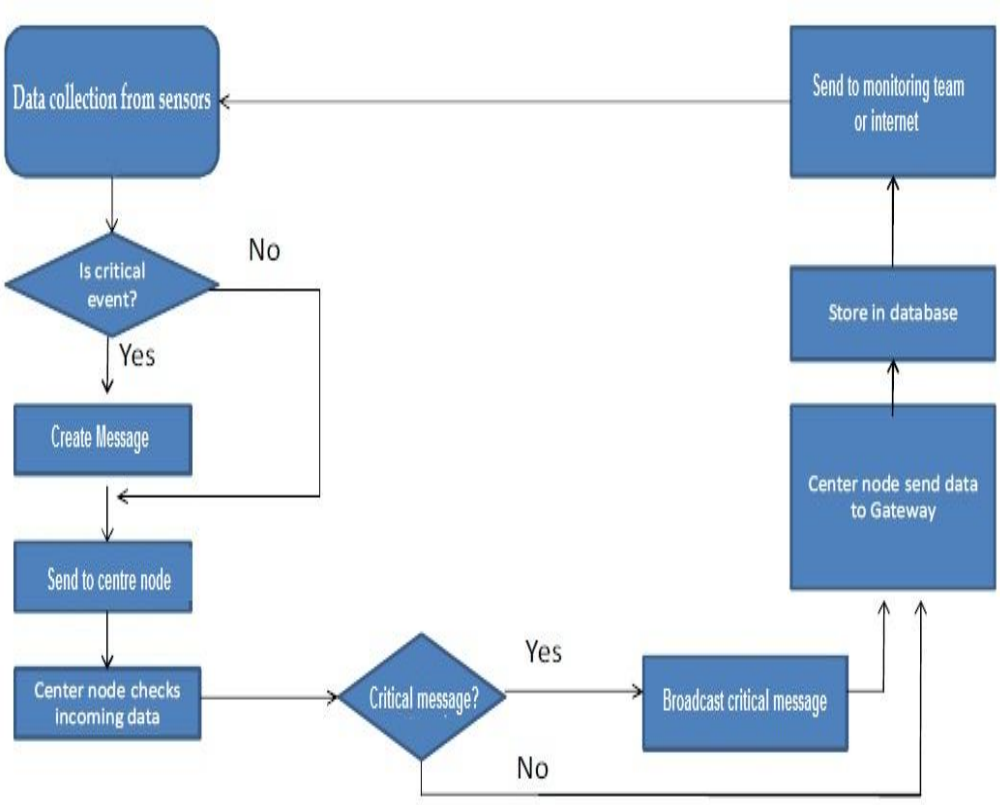

Figure 3: DFD for critical event sensing

V. CONCLUSION

This paper has presented a system of *Embedded Critical Event Sensing paradigm* for wireless sensor network communication. The sensor are deployed in the network to sense the events in the environment. As all the events in the environment are not critical events. The events captured for processing are taken to critical event database ECSED, which already contains a list of predefined critical events as per the user's requirements of different domains. These events are separate and vary from domain to domain as discussed above. . The sensor CPU looks into the ECSEP database for the said event. If the event is found in ECSED, it is immediately reported to the base station. It receives the varying values acquired by sensors deployed in the network as input. The acquired data can be much or less precise and are processed using CESAalgorithm. Once the event is declared as a critical event, it is simply transmitted to the base station. This system uses CESA algorithm and ECSED database to tackle the problem of critical event handling. This approach can be embedded to any existing sensor network and will have no effect on the existing settings and configuration of the network, beside this can be used in various application domains like defense systems, forest fire, wild life protection, mob control, surface and marine exploration, agricultural purpose and controlling the environment pollution.

REFERENCES

[1]. S. Maden, M. Franklen, J. Hellersten, & W. Hong, "The design of an acquisition query processor for sensor networks," in SIGMOD, 03, pp. 415.

[2]. K. Pits et al. (2011), Data Security on Wireless Sensor Network", IOP J. Phys., Conf. Series 52 12433-1-4.

[3]. L. Wyan et al. (2013). Secure and Efficient Data Transmission in Cloud based Wireless Sensor Networks. International J of Scientific & Research Pub. Vol-3 (5), 02-04.

[4]. T. Millano et al (2005). "Balancing computational and transmission power consumption in wireless image sensor networks,". International Conf. Virtual Security, Human-Computer Interfaces and Measurement Syst., vol. 16-18, pp. 5.

[5]. Michel Franklen, "Declarative interfaces for sensor network," NSF Sensor Workshop, 2003.

[6]. B. Jaio, S. Son, &J. Stankovic, "Generic event service middle-ware in sensor networks," INSS, 2005.

[7]. K. Kapitanova& S. H. Son, "MEDAL: A compact event description & analysis language for wireless sensor networks," INSS, 2009.

[8]. E. Tapia, S. Intille, & K. Larson, "Activity recognition in the home using simple & ubiquitous sensors," in Pervasive Computing, 2004, pp. 158.

[9]. C. Wren & E. Tapia, "Toward scalable activity recognition for sensor networks," in Location & Context-Awareness (LoCA), 2006, pp. 168.

[10]. P. Castro, P. Chiu, T. Kremenak, & R. R. Muntz, "A probabilistic room location service for wireless networked environments," Ubi Comp, pp. 18.

[11]. M. Duarte & Y.-H. Hu, "Distance based decision fusion in a distributed wireless sensor network," IPSN, pp. 392.

[12]. J. Miling et al. (2009), Data Security and Transmission in Wireless Sensor Networks. Ph.D. Thesis, Tokyo University, Tokyo.

[13]. M. Healy et al. (2007) "Efficiently Securing Data on a Wireless Sensor Network", IOP J. Phys., Conf. Series 76 12063-1-6.

[14]. Manel Abdel Kader (2009). Multi-target Tracking Using Wireless Sensor Networks Based on Higher-Order Voronoi Diagrams. Jour. OF NETWORKS. Vol-4 (7), 589.

[15]. M. Richard et al. (2008). Secure Transmission over WSN with a Surveillance. . Journal of Technical & Scientific Research Pub. Vol-4 (6), 01-04. 26-28.

[16]. Marinus Maris (2000). Security Aspects of a Surveillance System using a Sensor Network. TNO Defense, safety & security, Netherlands. 26-28.