

# Model of Real Time Architecture for Data Placement in Wireless Sensor Networks

Sanjeev GUPTA<sup>1</sup>, Mayank DAVE<sup>2</sup>

<sup>1</sup>*Doeacc Society, Autonomous Body of Department of I.T., Government of India, Chandigarh, India*

<sup>2</sup>*Department of Computer Engineer, National Institute of Technology, Kurukshetra, India*

*Email: sanju\_anita@yahoo.com, mdave@nitkkr.ac.in*

*Received October 6, 2009; revised November 19, 2009; accepted November 20, 2009*

## Abstract

Wireless sensor network (WSN) technology has promised fine grain monitoring in time and space as well as at a lower cost than is currently possible. These sensor networks are required to provide a robust service in hostile environments. Therefore the issue of real-time and reliable data delivery is extremely important for taking effective decisions in WSN. In this paper the architecture for reliable and real time approach by using sensor clusters has been proposed for storage management. Instead of storing information in an individual cluster head as suggested in some approaches, storing of information of all clusters, inside the cell is recommended within the corresponding base station. For data dissemination and action we have used Action and Relay Stations (ARS). We have developed programming model for formal specification and verification of our architecture.

**Keywords:** Modeling, SPIN, Promela, Centralized Clustering, Base Station, Sink, Cellular, Formal

## 1. Introduction

The issues of enhanced availability and reliability besides energy saving are extremely important for taking effective decision in a mission critical Wireless Sensor Network based applications. The behavior of sensor network based on infrastructure can be critical for human life, environment, decision-making and consequence of misbehavior can be catastrophic.

Given the increasing sophistication of WSN algorithms and the difficulty of modifying an algorithm once the network is deployed; system performance and functionality prior to implementing such algorithms must be validated using formal models. The formalism should be intuitive and should support specifying the algorithm at an appropriate level of abstraction, so that a formal specification can be well understood and can provide a useful starting point for an implementation of the algorithm.

The inherent complexity of concurrent real-time systems makes it necessary to employ mechanized, formally supported methods to analyze early life-cycle artifacts. In this context the main questions to be answered are whether the requirements are consistent and correct with the intended behavior of the system and whether the sys-

tem's design correctly implements the requirements. Analyzing a system amounts to exploring its behavior. A complete analysis makes it possible to predict the behavior under all circumstances. Intuitively a system is correct if it always behaves as intended. The protocol description under analysis should conform to its expected properties of reactivity and robustness.

The process of establishing or refuting a property for a given system is called verification. Verification process enables the designer to be confident that the formal description of the system does satisfy the system requirements of reactivity and robustness.

Our aim is to describe formally the architecture of reliable and real time data placement model. The rest of the paper is organized as follows. Section 2 introduces the modeling and analysis of WSN. Section 3 explains the System Architecture. Section 4 exposes the detailed Modeling and Specification. Section 5 describes Formal Verification using SPIN. Section 6 reveals results.

## 2. Modeling and Analysis of WSN

Testing is used to check whether a given system realization conforms to an abstract specification. But it can be applied only after a prototype implementation of the sys-

tem has been realized. Formal verification [1] as opposed to testing conducts an exhaustive exploration of all possible behaviors of the system. Formal verification works on models rather than implementations. Both techniques can be supported by tools.

Model checking is a fully automated technique for the verification of finite state systems. It is a method to verify the correctness of software designs. A model checker explores all states reachable from an initial state and validates a set of correctness properties on the model.

In model checking [2] algorithms executed by computer tools are used in order to verify the correctness of systems. Since sensor networks can be considered in terms of communicating finite state machines, they can be described as a set of concurrent communicating processes in Promela [3]. The Spin [4] is a model checking method and appropriate tool for verifying the sensor networks specifications. In this paper the model of reliable and real time data placement architecture has been formally described.

### 2.1. Protocol Specification with Promela (Process Meta Language)

PROMELA is mainly a “protocol validation model” language. At the validation level, the model does not have to describe the exact details of the implementation. The focus is on the structure of the model. The language is based on the theory of finite state machines; it is very easy to understand its concepts for everyone who knows the techniques of state machines.

The most important design goal of PROMELA was the

specification of distributed systems. Such systems are represented by sets of concurrent, parallel processes that are able to communicate with each other. The language allows describing the properties of process prototypes and of global resources such as channels or shared variables that can be used to model the communication between processes.

### 2.2. SPIN Model Checker

SPIN (Simple Model Interpreter) is a generic verification system that supports the design and verification of asynchronous process systems [4]. This model checker accepts design specifications written in Promela. It accepts correctness claims specified in the syntax of standard Linear Temporal Logic [5].

### 2.3. Visual Interface to Promela—VIP

The VIP [6] tool is a Java based graphical front end to the Promela specification language and the SPIN model checker. VIP supports a visual formalism called v-Promela which extends the Promela language with a graphical notation to describe structural and behavioral aspects of a system. v-Promela also introduces hierarchical modeling and object-oriented concepts.

## 3. System Architecture

The proposed architectural framework [7] as shown in Figure 1 for data placement strategy is real time, reliable

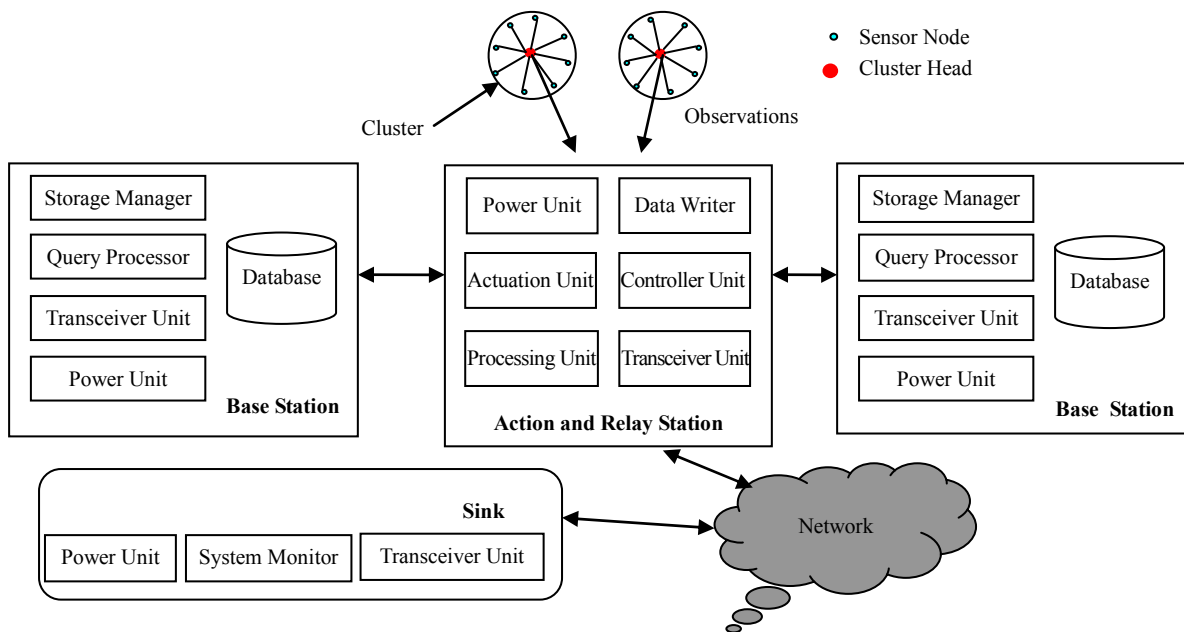


Figure 1. System structure.

and distributed. It supports basic kind of data integrity and disaster management approaches for wireless sensor networks [8]. Our architecture is portioned into modules each of which deals with various responsibilities of the overall system. The basic components and their functionalities used in our design are described briefly in following subsections.

### 3.1. Sensor Nodes

Sensors nodes are low-cost, low-power devices with limited sensing, computation and wireless communication capabilities. They can sense events in a circular coverage area with radius  $r_s$ . To save energy some sensors can be in sleeping state but they can be activated when it is necessary.

### 3.2. Cluster Structure

Since battery capacities of sensor nodes are severely limited and replacing the batteries is not practical we can use clustering of nodes for achieving efficient and scalable control. Clustering saves energy and reduces network contention by enabling locality of communication: nodes communicate their data over shorter distances to their respective cluster-heads. Only the cluster-heads (CH) need to communicate far distances to their respective action and relay station ARS (discussed in Subsection 3.5) saving energy of member nodes.

### 3.3. Cell Structure

Within every cell there will be number of clusters. An equal-sized cluster is a desirable property because it enables an even distribution of control (e.g., data processing, aggregation, storage load) over CHs; no CH is overburdened or under-utilized. Minimum overlap among clusters is desirable for energy efficiency because a node that participates in multiple clusters consumes more energy by having to transmit to multiple cluster-heads.

### 3.4. Base Station

Our primary concern is the persistence of data and to minimize the amount of data lost due to failures of nodes. The base stations are responsible for data storage in a distributed real time database framework.

### 3.5. Action and Relay Station (ARS)

The ARS are resource rich nodes equipped with better processing capabilities, higher transmission powers and longer battery life. The ARS nodes are placed on the

bordering areas of cells and are responsible for data dissemination in a time efficient manner.

During disaster any ARS may be collapsed. But probability of collapsing all ARS's of cell is very small. Only one ARS is enough to convey data from sensor network of a cell to a base station.

### 3.6. Sink

The sinks supervises and synchronizes the working of various components of the proposed model and depending upon feedback it sets the value of various parameters like retention period so that the network can work efficiently.

## 4. Modeling and Specification of WSN

Modeling and analysis of sensor networks require their formal specification [9]. In this paper models of various components of our architecture have been specified in Promela taking into account the following assumptions:

- Nodes are capable of measuring the signal strength of a received message [10].
- It is assumed that nodes have timers, but their time synchronization is not required.
- It is assumed that the network does not get partitioned. New nodes can join the network.
- All nodes belongs to a cluster and no node belongs to multiple clusters
- Communication links are bi-directional and unreliable
- The nodes are aware of the neighboring nodes within their transmission radius.
- Network of Sensors must always be a connected graph
- There is always a communication way between any two sensors in the network
- The sensor network does not contain unreachable sensors
- Any sensor will eventually be connected to the rest of the network

### 4.1. Model of Sensor Node

Different actions of sensor nodes executes at specific timestamp when the sensor nodes are in active state. Periodically it will send the reading to its cluster head. Upon receiving a message the sensor node will update its local variables and may take on further actions like go into the sleep state or may change the periodicity of sensor readings. Other actions of the sensor node can be advertising itself as a new cluster head and some proc-

essing actions. In Figure 2 working of normal sensor node is shown.

**Various possible states of sensor node (when working as normal node):**

- Probing (whenever sensor nodes wakes up it will start probing neighborhood)
- Sleeping (to conserve energy of sensor node it will periodically go for sleep)
- Active (during this state it will probe the environment for reading)

**Data objects of normal sensor node:**

- Sensor node location
- Energy the node has at the beginning
- Energy usage for packet transmission
- Energy usage for packet receipt
- Node status-active, sleeping
- Probing range of sensor node antenna's
- ARS location
- Cluster head alive timer
- BatchTimer:
  - It will be relevant to the sensor node if it is currently the leader of its cluster
  - The CH will read all the records from its buffer after the expiry of this timer
  - It will then aggregate the reading information location wise
  - It then will send the aggregated information to its respective ARS
- ProbeARSTimer:

- The CH will send the probe message for the ARS to its environment
- After expiry of this timer the CH will assume, there are no active ARS nodes
- EnergyLeftProbeTimer:
  - Periodically the cluster head will check amount of energy left
  - It will cease to be CH if energy left falls below a threshold limit

**Data objects of node working as CH:**

- Member nodes location
- Energy usage for packet transmission
- Energy usage for packet receipt
- Cluster head location

**Performance of sensor node:**

Two parameters decide the performance of the sensor node: The probing range and the wakeup rate:

- The desired redundancy can be achieved by setting the corresponding probing range. An application requiring high robustness may choose a small probing range to achieve high density of working nodes, thus high redundancy
- The number of wakeups decides the overhead, thus it should be kept low. However, if the wakeup rate is set to be too low, when a working node fails unexpectedly, there can be large “gaps” during which no working node is available
- Each probing node adjusts its wakeup rate according to the observation of its sleeping neighbors, so that transient node failures are tolerated by the application

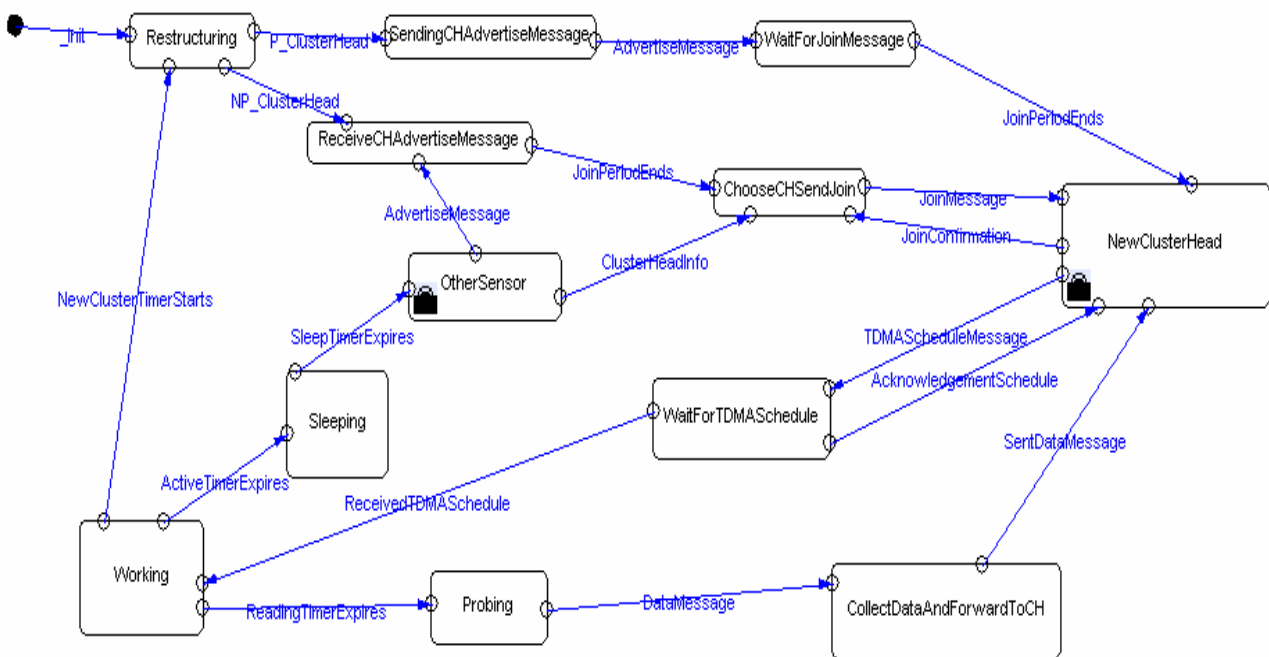


Figure 2. Working of normal sensor node.

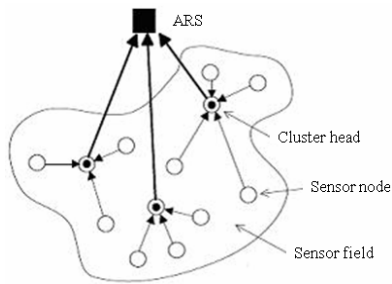


Figure 3. Cluster based sensor network.

4.2. Model of Cluster

As shown in Figure 3 centralized clustering method [11] has been used where the ARS organizes the network. The operation of cluster based sensor network is divided into rounds. Each round has clustering phase and the data transmission phase. Rounds are repeated to monitor events continuously. In the clustering phase, each sensor node first reports information about its current location and its battery level to the ARS. The ARS selects sensor nodes as CH depending upon their battery level. In the data transmission phase, each non-CH node sends monitored data to its CH. After receipt of data from the sensor nodes the CH performs data aggregation to remove redundant data and reduces the size of data to be sent to the ARS.

The CH will consume more energy than member nodes due to additional functions like receiving data from members, fusing data to reduce the size and sending the aggregated data. Therefore the CH role is rotated among nodes to evenly distribute the burden carried by a CH among all nodes, thus giving an opportunity for all nodes to have approximately the same lifetime. The sensor life

time may be improved further by selecting the proper points at which a CH role is relinquished to higher energy nodes via a CH rotation phase [12]. The working of Cluster class is shown in Figure 4.

4.3. Model of ARS

It performs two main tasks relaying received information to BS and in case of emergency automatically acting on stored data. It will store the information received in its local buffer and after some time send the combined information to further conserve power and communication bandwidth.

Working of ARS:

As illustrated in Figure 5 the working of ARS will be

- Maintaining a list of cluster (cluster heads) with whom it is communicating

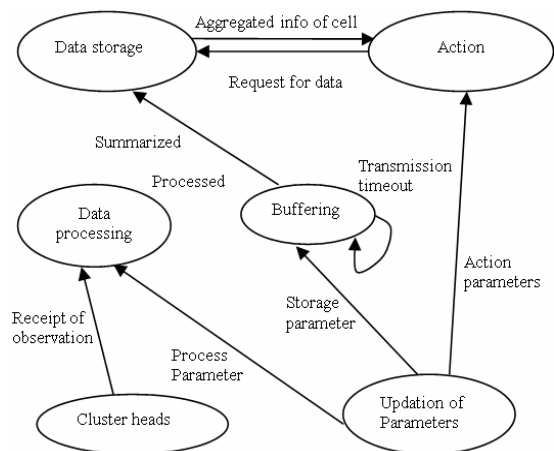


Figure 5. Finite state machine of ARS.

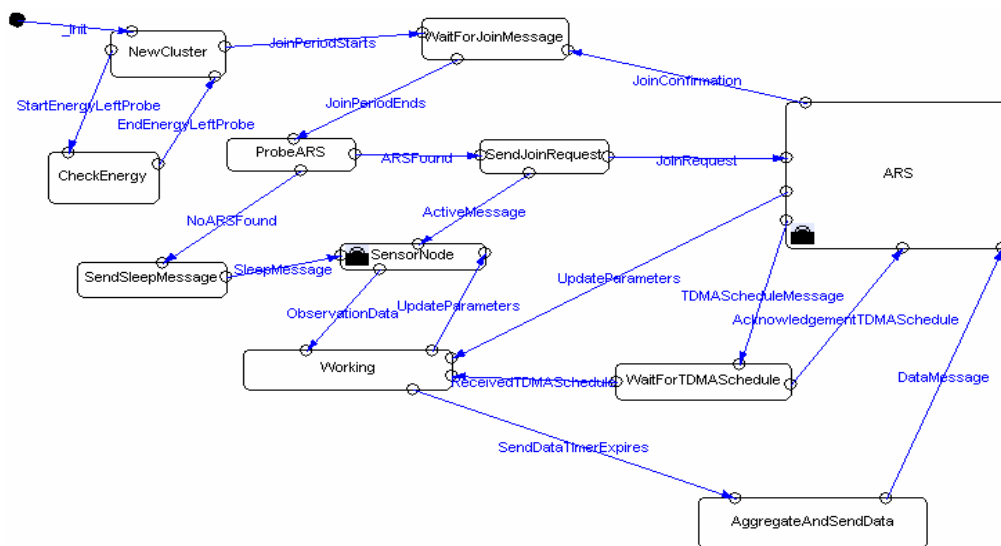


Figure 4. Working of cluster class.

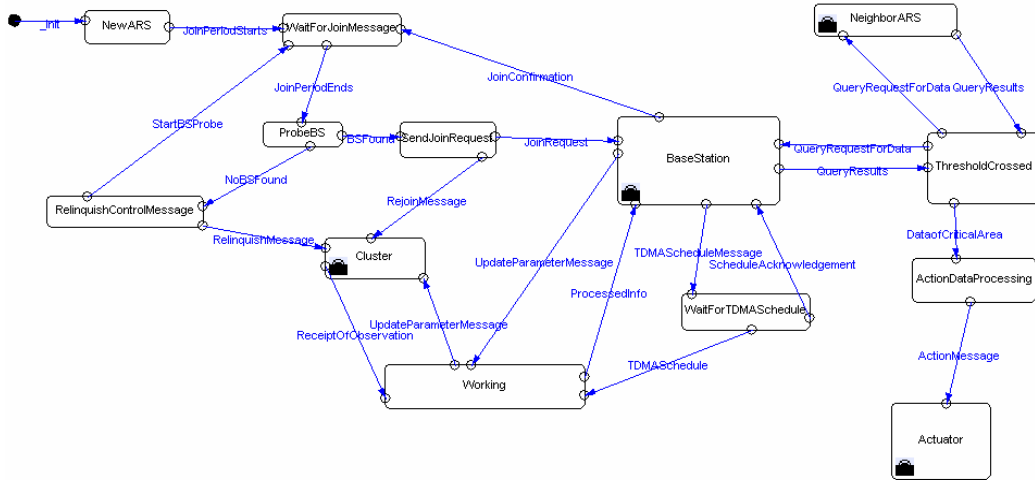


Figure 6. Working of action unit class.

- Relaying the information to adjacent base stations
- Periodically checking aliveness of the base stations
- Monitoring threshold value overshooting of which for a specified period is considered as emergency.

**Data objects of ARS class:**

- Status of ARS
- Action timer of the ARS node
- Radius of action cover area
- Status of base stations on each side of the edge

**4.4. Model of Action Unit Class:**

The action unit class consists of group of clusters and ARS. All the clusters belonging to action unit will send data directly to the ARS. The ARS will then forward the information received to the base station. In Figure 6 working of action unit class is depicted.

**Data objects of action unit class:**

- The list of cluster head locations
- Location Id of adjacent Base station
- List of neighboring ARS’s within the probe area

**4.5. Model of Base Station:**

The issue of energy saving, enhanced availability and reliability are extremely important for taking effective decision in a real world mission critical WSN based applications. Therefore the role of Base Station is very important in our model.

**Working of base station:**

- Periodically it will purge the old stored information
- Aggregating the information received over a period of time and storing the average readings only
- In case of emergency providing the information pertaining to the action area of the ARS to it

- Maintaining list of all ARS on the various edges of the cell of the base station

**Data objects of base station class:**

- Location of base station
- Database of reading records
- Purge timer on the firing of which it will delete old records from the database

**4.6. Model of Cell:**

A cell will have one base station and six Action Units. Figure 7 explains the working of cell.

**Data objects of cell class:**

- Location of base station
- List of locations of ARS nodes

**4.7. Model of Sink:**

The sink supervises the working of various components of WSN.

**Working of sink:**

- It will maintain a list of all base stations and their current status.
- It will maintain a list of all ARS and their current status.
- It will maintain a list of all clusters and their current status.
- It will store statistical information to control the operation of wireless sensor network.
- Periodically it will obtain the status of all the components of wireless sensor network
- To change the value of parameters to improve the behavior of WSN.
- From time to time providing the current status of WSN to the controlling authority

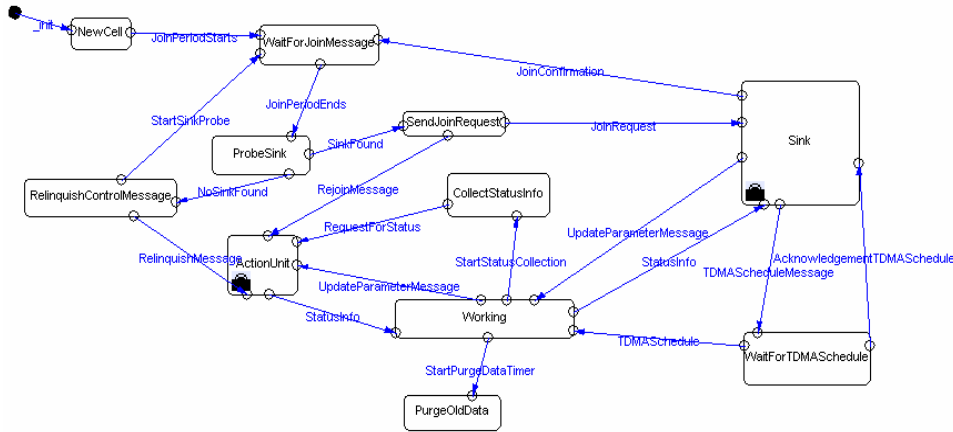


Figure 7. Working of cell.

Table 1. Simulation Parameters.

| Energy Assumptions  |                                      |                             |
|---|--------------------------------------|-----------------------------|
| $d_0$   | Threshold Transmission distance      | 80 meters                   |
| $E_{elec}$  | Electronics Energy                   | 50 nJ/bit                   |
| $E_{fs}$  | Amplifier Energy factor (free space) | 100 pJ/bit/m <sup>2</sup>   |
| $E_{tr}$  | Amplifier Energy factor (multipath)  | .0013 pJ/bit/m <sup>4</sup> |
| $E_{da}$  | Aggregation Energy                   | 5 nJ/bit/signal             |
| Initial Battery of all the nodes                          |                                      | 0.5J                        |
| Packet Size Assumptions                                   |                                      |                             |
| Cluster Head proposal Packet Size by Sensor Nodes         |                                      | 300 bits                    |
| Cluster Head Intimation Packet Size to Sensor Nodes       |                                      | 100 bits                    |
| TDMA Schedule Packet Size to Sensor Nodes by Cluster Head |                                      | 600 bits                    |
| Sensed Message Packet Size                                |                                      | 200 bits                    |

- It will request the controlling authority to replace faulty components of WSN.
- On the basis of statistical information it will change the value of performance parameters to improve the behavior of WSN.
- From time to time it will provide the current status of WSN to the controlling authority.
- It will request the controlling authority to replace faulty components of WSN.

#### Data objects of sink class:

- Storage of statistical information
- Initiates probe of wireless sensor network after expiry of enquiry status timer

#### 4.8. Model of RTWSN:

This class will represent entire WSN. This class will have one sink and many cells.

#### Data objects of RTWSN class:

- Location of sink
- List of locations of all cells

## 5. Formal Verification Using SPIN

We have used JSPIN [13] GUI interface for SPIN model checker for verifying our model because of its powerful model checking capabilities. The approach described in this work, has been to naively model the complete system and check it.

We have used state machine diagrams as a base for the PROMELA translation. A large number of properties have been specified to safeguard the system described above.

## 6. Results and Discussion

To show effectiveness of the proposed model the simulation software has been developed to conduct the said experiments. In the simulation 96 sensor nodes were randomly distributed in a cellular region of 120 m X 120 m. To simplify the analysis we have assumed maximum of 24 clusters and maximum of 4 clusters per ARS.

The data transmissions from sensor nodes were simulated until all sensor nodes died. For the experiments described here the parameters as described in Table 1

were used. For energy dissipation the model explained in[14] have been used, that depends on the distance between the transmitter and receiver. The performance of proposed model has been compared with popular Centralized-Clustering algorithm Leach-C taking exactly same environment and same assumptions. Although Leach-C appears to be promising centralized clustering algorithm, there are some areas for making protocol more energy efficient.

Instead of all the sensor nodes of the cell communicating to single BS, the proposed distributed model will be more energy efficient because average distance between cluster member nodes and respective ARSs will be much shorter than average distance between sensor nodes of clusters and the single base station of centralized clustering approaches.

Moreover instead of communicating and checking battery level of all the nodes of the cell to decide clustering, the ARS will only communicate with nodes reporting to it. Thus clustering process will be completed much faster and results in minimum use of invaluable energy of nodes and ARS.

The number of rounds that first node and last node dies as well as average energy dissipation and number of messages received is used as a key indicator to evaluate the proposed system. Experiment shows that our proposed model increases the lifetime of the network as illustrated in Figures 8, 9 and 10. The scalability is also very easy to achieve in our model.

### 7. Conclusions

It is fair to say that correct systems are as valuable as gold. To talk about correctness it is not sufficient to determine what wrong behavior is; but more importantly it has to be

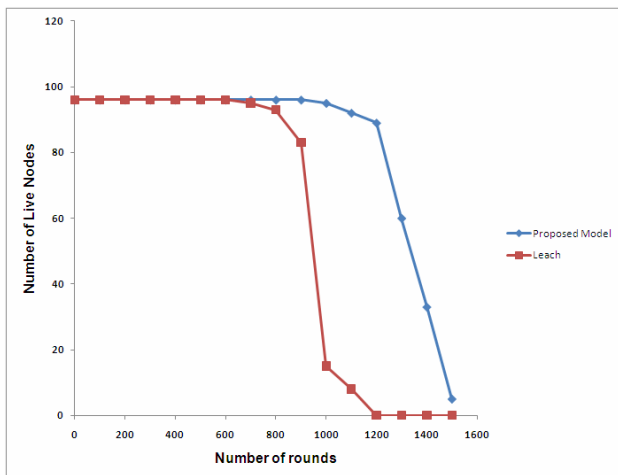


Figure 8. Number of live nodes over number of rounds.

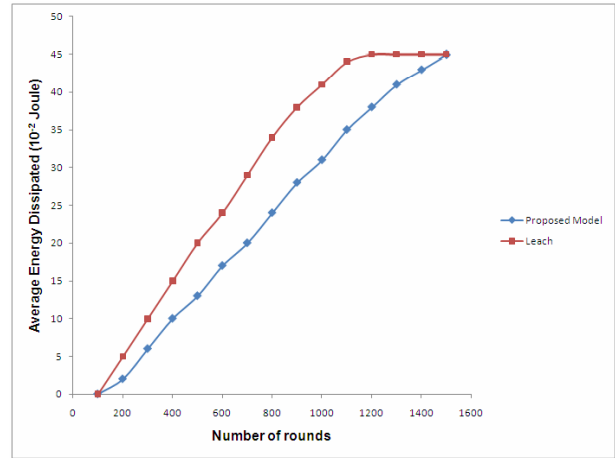


Figure 9. Average energy dissipation over number of rounds.

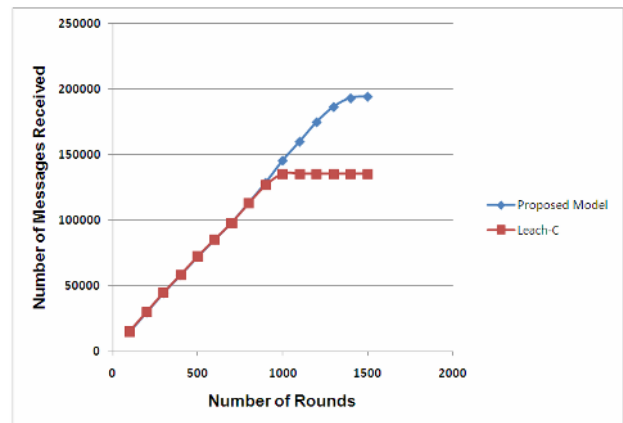


Figure 10. Number of messages received over number of rounds.

determined what is right. In this paper use of the PROMELA language and SPIN tool has been explained for verification of our model. It has been shown how a formal sensor behavior description can be transformed into Promela notation.

The sensor network has been described as a set of concurrent communicating processes in Promela for further verification using SPIN.

The SPIN has been applied for the analysis of our model to verify that it satisfies desired properties and/or is consistent with given global constrains.

We are also extending our simulation software to verify other aspects of our model like reliable, real time data retrieval and fault tolerance.

Looking forward, one can expect that a lot more power-efficient designs will be produced from this framework. Larger tests are needed to determine the optimal transition period to the change the number of clusters in the network. Similarly there should be synchronization



among different sensors reporting the sensed events to the ARS.

## 8. References

- [1] B. R. Haverkort, H. Hermans, and J. P. Katoen, "The use of model checking techniques for quantitative dependability evaluation," In IEEE Symposium on Reliable Distributed Systems, ISBN:0-7695-0543-0, pp. 228–238, 2000.
- [2] B. B'erard, M. Bidoit, A. Finkel, and F. Laroussinie, "Systems and software verification: Model checking techniques and tools," Springer-Verlag, New York, ISBN: 3-540-41523-8.L., pp. 190, 1999.
- [3] G. J. Holzmann, "Design and validation of computer protocols," ISBN: 0-135-39925-4, Prentice Hall, New Jersey, pp. 512, 1990.
- [4] G. J. Holzmann, "The SPIN model checker: Primer and reference manual," Addison Wesley, ISBN 978-0-321-22862-8, pp. 608, 2003.
- [5] Lamport, "The temporal logic of actions," ACM Transactions on Programming Languages Systems, Vol. 16, No. 3, pp. 872–923, 1994.
- [6] M. Kamel and S. Leue, "VIP: A visual editor and compiler for v-Promela. Tools and Algorithms for the Construction and Analysis of Systems," Springer Berlin, Heidelberg, doi: 10.1007/3-540-46419-0, pp. 471–486, January 2000.
- [7] S. Gupta and M. Dave, "Real time approach for data placement in wireless sensor networks," International Journal of Electronic Circuits System, Vol. 2, No. 3, pp. 132–139, 2008.
- [8] I. F. Akyildiz and I. H. Kasimoglu, "Wireless sensor and actor networks: Research challenges," Ad Hoc Networks, Vol. 2, No. 4, pp. 351–367, 2004.
- [9] M. G. Gouda and Y. R. Choi, "A state-based model of sensor protocols, Principles of distributed systems," doi: 10.1007/11795490, ISBN: 978-3-540-36321-7, Springer Berlin, Heidelberg, 2007.
- [10] Hill, J., R. Szweczyk, A. Woo, S. Hollar, D. Culler and K. Pister, "System architecture directions for networked sensors," ISSN: 0362-1340, ACM SIGPLAN, Vol. 35, No. 11, pp. 93–104.
- [11] W. B. Heinzelman, A. P. Chandrakasan, and H. Bala-krishnan, "An application-specific protocol architecture for wireless microsensor networks," IEEE Transaction on Wireless Communications, Vol. 1, pp. 660–670, 2002.
- [12] S. Gamwarige and E. Kulasekera, "Optimization of cluster head rotation in energy constrained wireless sensor networks," IFIP International Conference on Wireless & Optical Communications Networks, Singapore, doi: 10.1109/WOCN.2007.4284155, pp. 1–5, July 2007.
- [13] M. Ben-Ari, "Development environments for spin and erigone," <http://stwww.weizmann.ac.il/g-cs/benari/jspin/>.
- [14] T. Rappaport, "Wireless communications: Principles & practice," Prentice-Hall, 2nd Edition, ISBN: 0130422320, pp. 736, 2001.