

# Survey on Sink Repositioning Techniques in Wireless Sensor Networks

Prerana Shrivastava  
Research Scholar,  
Department of Electronics,  
G.H.Raisoni College of Engineering,  
Nagpur University, India

S. B. Pokle  
Phd, Professor and Head,  
Department of ECE,  
Shri Ramdeobaba College of Engineering and  
Management, Nagpur

## ABSTRACT

Sink relocation is essential in wireless sensor network since relocating sink can balance the traffic load in the midst of multiple nodes and thereby decrease the miss rate of real-time packets. The main constraint in wireless sensor networks is limited energy supply at the sensor nodes so it is important to deploy the sink at a position with respect to the area of interest; a specific area that is of more interest to the end-user, such that total energy consumption is minimized. But little survey work has been done in the sink repositioning topic till date. Existing survey on sink repositioning techniques in wireless sensor networks consider only static and mobile sink. No much importance is given to the hybrid sink deployment techniques. Hence in this survey, we present an up-to date survey of the sink deployment technique in the wireless sensor networks. This survey provides a thorough analysis of the metrics supported by each technique during sink relocation and presents the limitations of these techniques. Finally, a comparative study of sink deployment techniques based on various metrics is also performed.

## Keywords

Wireless Sensor Networks (WSN), Sink relocation, Sensor Nodes, Sink Repositioning Techniques

## 1. INTRODUCTION

### 1.1 Wireless Sensor Network

The architecture of Wireless sensor network (WSN) comprises of large number of sensor nodes that reports its measurements to sink or other gateway nodes that are distributed locally. [1] Generally, sensor nodes are distributed in a geographical region for sensing some physical phenomena like vibrations, temperature and earthquake. Sort of the sensor used decides the consumption of energy in the sensing subsystem [2].

The communication between base station or the gateway and several wireless sensors is achieved in WSN through radio links. Sensed data at node level is gathered, compressed and transmitted to the gateway, either directly or by intermediate nodes based on necessity. Finally, the transmitted data is presented to the system by gateway.

The most important issue to be considered in managing wireless sensors is energy consumption. Replacing or recharging the batteries of sensors is not such an easy task, in order to achieve efficiency in energy consumption for a longer period; the energy has to be shielded. Energy is consumed in sensor nodes by various means, among that the communication module devours energy for a larger extent. The sensed data can be transmitted directly to the sink for smaller networks, conversely for larger networks data must be transmitted by multihop communication, i.e data is transmitted through intermediate

sensor nodes. In above discussed two cases, communication distance is an important factor for energy consumption. [3] The topologies namely star, mesh, bus, tree and fully connected are the types of topologies in sensor networks. [4]

### 1.2 Sink Relocation

In WSN, sinks are bounded with abundant resources and sensors that generate data are termed as sources. The sources can transmit data to one or multiple sinks for the purpose of analysis and processing. [5]

In wireless sensor networks, sink relocation is preferred by all applications that involve real time traffic for even in the middle of multiple nodes it can balance the traffic load and thereby lessen the miss rate of real time packets. To carry out sink repositioning, multiple sink deployment and sink mobility can be considered. Precise information of the area being monitored is needed to offer an ideal solution by the sink deployment method, but this method is not a realistic often. To reallocate the sink, its odd pattern of energy must be considered. [6]

### 1.3 Sensor Node Positioning

In query based WSN, the major challenge is that the sensor nodes not being able to know the real time information on the sink position. In multihop communication strategy, certain estimation strategy is required at each hop. [7] The effectiveness of WSN and efficiency of its operation are determined by the node position. Deployment of nodes in the network relies independent of metrics that base the state of the network or it assumes a pattern of network operations, which remains unchanged throughout the network lifetime. [8] The nodes can be distributed adaptively based on the requirement as either controlled or random.

### 1.4 Issues of Sink Repositioning

1. When the gateway relocation is sensible, where the gateway has to be placed and how the data traffic has to be handled during the gateway's movement is the most basic issue. Given the traffic distribution and network state at that time, gateway relocation must be based on the motivation by the inefficient pattern of energy depletion or an intolerable increase in the missed deadlines whenever real time packets are used. If such condition is detected, then to enhance the network performance the gateway should identify the most suitable location.

2. Finding an efficient strategy for optimal gateway location is complex and it is NP hard problem. Two characteristics of gateway that are responsible for complexity are as follows,

- The gateway can be moved to immeasurable possible positions, which is the first responsible characteristic for complexity.

- Every temporary discovery solution of gateway location requires the construction of new multi-hop network topology to confirm that the current temporary solution is qualifyable than previous temporary solution. The mathematical expression for this problem necessitates more parameters such as positions of all deployed sensors and state parameters like energy level and transmission range. For a network with large number of nodes, the pursuance of exhaustive search will be impractical.
- Further, when the gateway is moved multiple times, the optimization process has to be repeated for the dynamic nature of the network makes the sensor state and sources of data variant.

3. All the above to the optimization problem, the gateway have need of performing trade-off analysis between the old and new location for gain achieved and overhead on sensors. For instance, the affect on the lifetime of an individual sensor and the system level metrics such as average energy per packet has to be considered when energy metrics are of utmost concern. The gateway moves towards the location in case the relocation is justified. For this scenario, the trade off analysis has to be done so that the data are not lost. However, data loss can occur by packet losses, when the gateway moves out of the transmission range of the sensors that have direct communication path with the gateway. [9]

## 2. RELATED WORK

Mohamed Younis et al [8] in this paper, reported on the current state of the research on optimized node placement in WSNs. The issues like identifying the various objectives and enumerating the different models and formulations are highlighted. The placement strategies is categorized into static and dynamic depending on whether the optimization is performed at the time of deployment or while the network is operational, respectively. Further the published techniques are classified based on the role that the node plays in the network and the primary performance objective considered. The paper has also highlighted on open problems in this area of research [8].

Zoran Bojkovic et al [10] in this work, dealing with challenges for WSNs deployment, started with mobility-based communication in WSNs. Then, service-oriented SANETs (SOSANETs) was introduced as an approach to build customizable SANETs. In the second part, localization systems are described and then self configurability, situation awareness and intrusion detection system are analyzed. In the third part, wireless distributed detection as well as a model for WSN simulation is presented. Finally, conclusions and proposals for future research are given [10].

## 3. SINK DEPLOYMENT

### 3.1 Static Sink Deployment: Single Sink [12], [9]

#### 3.1.1 Base Station Repositioning based on Quality of Route set up [12]

In order to enhance the system lifetime, we propose a limited motion to the stationary gateway towards the sources of largest traffic. Such approach has multiple advantages. First, the gateway will be near to nodes collecting maximum data, thus reducing communication related energy consumption. It is thus expected that the average energy per packet to be reduced. In addition, the overall latency time for data collection will be

lower. Moreover, the packet throughput will be higher since it is expected that messages pass through fewer hops and travel shorter distances making them less likely to be dropped.

To achieve such performance gains, one needs to address the basic questions of when to relocate the gateway, where to place it and how to handle data traffic during the transition.

#### 1] When to Move the Gateway

The decision for a gateway movement has to be based on an odd pattern in route setup. Once the gateway detects that pattern it will consider relocation. Such consideration does not necessarily lead to an actual relocation. The gateway qualifies the impact of the potential new location on the quality of the route setup, measured in some static metric. If the impact is positive, the gateway starts to move. Therefore the “When” and “Where” issues of the gateway movement are very closely inter-related. The following notations are used for considering gateway relocation:

$G$ : Gateway in a cluster.

$G_I$ : Set of sensors less than distance  $D$  from  $G$

$G_R$ : Set of sensors that are one hop away on the active route.

$G_{R1}$ : Set of sensors in  $G_R$  which are also in  $G_I$ , i.e.  
 $G_{R1} = G_I \cap G_R$

$G_1^{new}$ : Set of sensors less than distance  $D$  away from the gateway at the new location.

$G_R^{new}$ : Set of sensors those are one hop away on new route at the new location.

$G_{R2}$ : Set of sensors in  $G_R^{new}$  which are also in  $G_1^{new}$ , i.e.,  
 $G_{R2} = G_1^{new} \cap G_R^{new}$

$P_i$ : Packet traffic measured as the number of packets per frame, going through a node.

$PT$ : Set consisting of packet traffic of each sensor in  $G_{R1}$  in an ascending order

$PT^{new}$ : Set consisting of packet traffic of each sensor in  $G_{R2}$  in an ascending order

$E_{(Tri)}$ : Energy consumed by a node  $i$  in transmission of a packet to the next hop.

Note that  $G_1^{new}$ ,  $G_R^{new}$ ,  $G_{R2}$  and  $PT^{new}$  are calculated by placing the gateway at new location. The basic idea is for the gateway to check the changes in data paths in consecutive routing cycles. Typically re-routing is performed in response to a high packet loss caused by the energy depletion or failure of a relay node or is triggered by a change in data sources that requires setting a new topology. When comparing routes in two consecutive cycles, if the data sources are the same and the nodes in previous  $G_{R1}$  differs from that the current  $G_{R1}$ , the gateway perform further analysis. The gateway checks the nodes that used to be in the previous  $G_{R1}$  and are excluded from the new  $G_{R1}$ . If these nodes were among the bottom 70%

of the set PT, relocation would not be necessary. On the other hand if these nodes were forwarding high traffic (among top 30% in PT), the gateway perform a heuristic search for a better position.

To qualify the impact of placing the gateway at a new position, some static network performance metric has to be used. Most of the popular performance metrics for sensor networks, such as average delay per packet and throughput, are based on the network operation over a period of time and are typically network-wide in scope. Therefore, we use as a metric the total transmission energy of the relay node that are one-hop away from the gateway, basically those in  $G_R$  and  $G_R^{new}$ . A positive impact on the total transmission energy for these nodes would be a good indication for the gateway move. However, the reduction in the total transmission energy has to exceed an application threshold to justify the overhead, as discussed later in this section. A constant  $\delta$  can be derived based on the overhead for handling the motion both at the gateway and at the network level. The condition for relocation can be mathematically defined as:

$$\sum_{\forall i \in G_R} E(TR_i) \times P_i - \sum_{\forall j \in G_R^{new}} E(TR_j) \times P_j > \delta$$

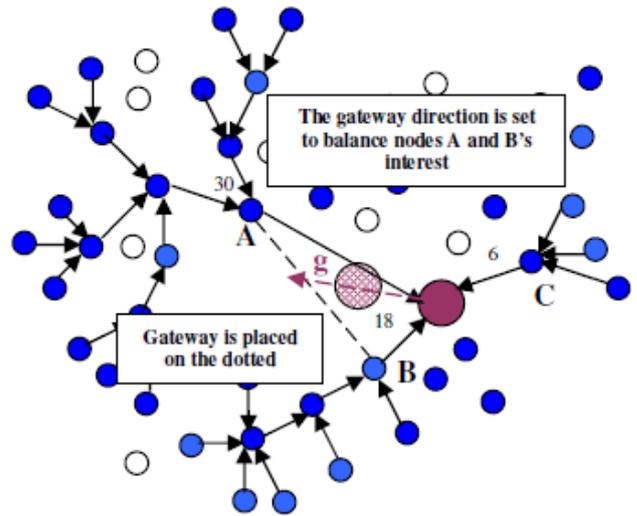
As mentioned earlier, finding an optimal new location for the gateway is an implicit issue for the relocation decision. Optimal placement of the gateway is an NP-hard problem in the general term. Therefore we pursue a heuristic search and settle for a quasi-optimal location to overcome such complexity. In the next subsection we explain such heuristic approach.

## 2] Where to Move the Gateway

Once we decide the gateway movement, the complex problem of determination of direction of motion for G could be based on the on the network traffic. The idea is to move G towards the sensors that generate the most number of packets. However, it may be infeasible for G to go that far from its current position, e.g. due to limited mobility resources. The network topology changes dynamically and thus there is a risk for wasting substantial resources for reaching a far position, which turns shortly after to be not optimal due to changes in the environment. To achieve similar effect, we try placing G close to the relay nodes in  $G_R$ , which routes the largest number of packets. To identify such relay nodes, the list PT is consulted and the top nodes on the list are picked as candidates.

In our approach, G is to move towards the most dominating node in PT, whichever gives a greater value of  $\delta$ . In case multiple relay sensors in the top part of PT share that dominance, a weighted average based on the distance between the gateway and these relay sensors and their traffic density is taken into account. The idea is to balance the gateways interest while picking the direction of the move.

A position “g” equidistant, in terms of distance  $\times$  traffic density, from the relay sensors on the different high traffic paths is calculated. While such repositioning will be ideal for high traffic paths, it can worsen the total power consumed on other paths with lower traffic density. Therefore, the gateway will be placed on some point on the straight line between its current position and the newly calculated position “g” that is equidistant from picked relay nodes.



**Figure 1. G balances its interest in serving nodes A and B while deciding how far should it travel in that direction.**

Figure.1 illustrates how to pick the direction of the gateway movement to optimize total transmission energy for the top two relay nodes in PT. Balancing the interest of the relay node A and B changes the slope of the gateway travel path. However, placing the gateway at position g on the line between A and B increases the efficiency of packet transmissions from A and B to the gateway at the expense of node C. Therefore a position on the path to g that minimizes the total transmission energy for the entire  $G_R$  set will be looked for using a bisection search.

## 3] Handling Gateway Motion

Once the new location of the gateway has been picked and confirmed to enhance performance of the sensor network, a plan for moving the gateway without losing the packets destined to it has to be generated. In order to handle this problem, they move the gateway during the routing phase of the utilized algorithm. The underlying routing approach has different phases, namely data transfer phase and routing phase. In the data phase, each sensor node sends data packets to the gateway using its routing table. On the other hand, in the routing phase, the sensors send their information to the gateway and based on such information, multi-hop routes to the gateway are created and routing table information is sent back to the sensor nodes. Since no data packet is being transmitted during this phase, gateway relocation can be easily done by changing the position of the gateway followed by the modification of network topology at the new position without any effect on data transfer.

### 3.1.2 Routing for Real Time data [9]

In this system, when the gateway is moved, is decided according to the process discussed in the previous section.

#### 1] New Gateway's position

Determining the new position to which the gateway would relocate is the central problem. To limit the complexity of such optimization they determine few directions for the gateway motion based on the current network traffic. The idea is to move the gateway towards the sensors that generate the most number of packets. However, it may be infeasible for the gateway to go very far from its current position due to limited mobility resources or harsh terrain. Moreover, a big move in one direction may cause the gateway to lose connectivity to other data sources; an issue that is not easy to sufficiently handle by extending the path from these sources due to a

shortage in the available live nodes in G1 that can act as forwarders. In addition, the topology of the sensor network changes dynamically and thus there is a risk for wasting substantial resources for reaching a far position, which turns shortly after to be not optimal due to changes in the environment. To achieve similar effect, they try placing the gateway close to the relay nodes in  $G_R$ , which routes the largest number of packets. To identify such relay nodes, the list PT is consulted and the top nodes on the list are picked as candidates.

If the top node in PT is dominating in terms of traffic density, the gateway is to move towards that node. For simplicity they use a linear motion pattern, i.e. the gateway move in a straight line. In case multiple relay sensors share that dominance in packet traffic, a weighted average based on the distance between the gateway and these relay sensors and their traffic density is pursued. The idea is to balance the gateway's interest while picking the direction of the move. A position that is equidistant, in terms of distance  $\times$  traffic density, from the relay sensors on the different high traffic paths is calculated. Finding such a position is similar to identifying the centroid of a polygon consisting of multiple points.

While positioning the gateway at the centroid will be ideal for high traffic paths, it can worsen the total power consumed on other paths with lower traffic density. Therefore, the gateway will validate the overall impact on transmission energy before confirming the move. The expected network topology at the new location is checked to ensure that the total transmission

energy of nodes in  $G_R^{new}$  is significantly less than those of  $G_R$ .

## 2] Handling Packet Delivery During the Move

Once the new location of the gateway has been picked and confirmed to enhance the performance of the sensor network, a plan for moving the gateway without losing the packets destined to it has to be generated. Our gateway motion handling mechanism tries to maintain continual packet delivery to the gateway while restricting changes to the network topology. Avoiding radical changes to the data routes limits the disruption of on-going data traffic and also curtails the overhead that the gateway's relocation introduces to the network.

The gateway is assumed to move towards its new position in strides. The strides are to be in straight line between intermediate points. At each intermediate point, the gateway checks whether it can still be reachable by the last hops while traveling on the next stride. If the gateway at the new intermediate point stays within the transmission range of the last hop nodes, the gateway will simply indicate to these nodes the need for adjusting the output power of their radio to cover the gateway's next move. The gateway can then start approaching the new intermediate point without worrying about the potential of broken links. If the gateway detects that it can lose connectivity to the last hop nodes while on the move, it explores the option of employing sensor nodes to forward the packets. Once, it assesses the need for employing forwarder nodes, the routes are extended by those forwarders and that information is sent to relevant sensor nodes. This procedure is repeated till reaching the new position set by the relocation algorithm.

## 3.2 Mobile Sink Deployment

[3],[6],[8],[9],[10],[11],[12]

### 3.2.1 Single Mobile Sink [11], [17]

#### 3.2.1.1 Dynamic base Station Repositioning using Genetic Algorithm [11]

In DBSR algorithm the primal population consists of  $n$  chromosomes which show the position of BS. Each chromosome includes two parts; X (length of network environment) & Y (width of network environment). They have encoded by binary encoding scheme. Each chromosome is evaluated by fitness function. They have applied modified 2-point crossover and random point flip for mutation operation. In addition, for new population replacement, the selected population is replaced with next population. The condition of genetic algorithm expiry is based on the number of generations they've supposed.

#### 1] Population

They have applied binary encoding in the proposed algorithm that is each chromosome is related to BS position. The length and width are supposed for the environment which sensors are distributed. It has supposed that all of the sensors are placed in a point with a specific width and length. Chromosomes are consisted of two parts: First binary part is related to X (length of sensor point) and the second binary part is related to Y (width of sensor point). The number of X & Y bits depends on the length and the width of network environment.

#### 2] Fitness

Fitness function is calculated based on distance and residual energy parameters in sensors. Each chromosome which enjoys random X & Y that it shows the position of BS. Summation of distance between this random point and all of the sensors is achieved by multiply ratio for each sensor (this ratio introduces inverse of residual energy in sensor) that shown in equation (1). Residual energy is supposed as a number between 1 and 10.

$$abf = \sum_{i=1}^n \frac{1}{W_i} \sqrt{(x-x_i)^2 + (y-y_i)^2}$$

Where  $n$  = number of sensor nodes

The fitness function is given as follows:

$$\text{Fitness} = m - abf$$

#### 3] Selection

The selection process selects chromosomes from the mating pool according to the survival of the fittest concept of natural genetic system. In each successive generation, a proportion of the existing population is selected to breed a new generation. This approach uses 80% as crossover probability, which means that 80% of the population will take part in crossover. The probabilities for each chromosome are calculated according to their fitness values, and selection is in proportion to these probabilities where the chromosome with lower probability has more chance of being selected. The proportions are calculated as given below.

$$prob(ch_i) = \frac{fitnes(ch_i)}{\sum_{i=1}^n fitnes(ch_i)}$$

Once the probabilities are calculated, Roulette Wheel selection is used to select parents for crossover.

### 3.2.1.2 Adaptive Sink Mobility in Event Driven Clustered Network [17]

Nodes with little remaining battery power could be spared if the sink would move closer to them, while moving away from nodes with more remaining energy. One such possible strategy is when the maximum relative energy that a node has to spend on transmission is minimized, i.e.,

$$\max \frac{E_i}{E_{rem,i}} \rightarrow \min \quad i \in A$$

Where  $E_{rem,i}$  denotes the average remaining energy per node within the  $i^{\text{th}}$  cluster. Note that they do not take into account the remaining energy of the current cluster head, but the total energy within the cluster. By doing so they ensure a more balanced energy consumption in the network. The sink will not get close to a cluster full of energy, just because its CH is near of depletion; the CH should be changed instead. On the other hand, a cluster that is running out of energy will be spared even if its current CH is still in good shape. The CH will continuously monitor the energy level of the sensors in the cluster. An aggregated report on the cluster's energy level can then be passed to the sink, e.g., by piggybacking it on the normal reports sent by the active cluster heads; the sink can then use the information and adapt its movement accordingly. When the CH role is passed to a new sensor, the information about the remaining energy in the cluster is transmitted as well. The minimization problem is equivalent with the following:

$$(x_0, y_0) = \arg \min_{(x,y)} \left( \max_{i \in A} \frac{\sqrt{(x-x_i)^2 + (y-y_i)^2}}{E_{rem,i}} \right)$$

## 3.2.2. Multiple Mobile Sink

### 3.2.2.1 Sink Repositioning Algorithm Based on Particle Swarm Optimization [6]

The proposed Multi Objective Sink Repositioning (MOSR) algorithm is designed to solve the Mobile Element Scheduling problem, which aims to schedule the visits of the mobile element to each sensor to avoid data loss due to sensor buffer overflow. With the MOSR algorithm, first all nodes are grouped into clusters, such that nodes in the same cluster have similar deadlines and are geographically close to each other. Then, to solve the scheduling problem of the mobile element within a single cluster, the PSO is used to reduce the buffer overflow or data loss of the sensor nodes. Finally, the schedules for individual groups are concatenated to form the entire schedule.

Let  $C_j = 1, \dots, N$ , denote cluster  $j$ , where  $N$  is the total number of clusters. In MOSR, nodes are first partitioned into clusters in such a way that overflow times of the nodes in cluster  $C_i$  is smaller than those in  $C_j$ , for  $j > i > 0$ . Moreover, the range of overflow times for nodes in  $C_{i+1}$  is twice that of  $C_i$ . This allows the nodes in  $C_i$  to be visited twice more frequently than the nodes in  $C_{i+1}$  during generation of the visit schedules.

Then, each cluster again is partitioned into sub-clusters so that nodes in the same sub-cluster are geographically close to each other. This two level partitioning results in groups of nodes with similar deadlines and locations. Therefore, in each sub-cluster, node visits can be scheduled using a PSO and vehicle routing problem.

## 1] Clustering According to Overflow Times

Let  $\min_{otime}$  and  $\max_{otime}$  denote the minimum and maximum overflow time of all nodes. Nodes are assigned to clusters according to the following equation:

$$n_i \in c_j, \{if 2^{j-1} \min_{otime} \leq i_{otime} \leq 2^j \min_{otime}, j=1,2,K K N-1$$

$$\{if 2^{j-1} \min_{otime} \leq i_{otime} \leq \max_{otime}, j = N$$

A cycle is defined as a closed path among a set of nodes, such that no node is included more than once in the same cycle. They have also defined a supercycle as a closed path composed of concatenated cycles such that every node is included at least once in a supercycle. In this algorithm, a supercycle is equivalent to the period of the sink schedule.

### 1.1] Sub-Cluster Partitioning According to Locations

Each cluster obtained above is then partitioned into sub-clusters according to the node locations such that the nodes in the same sub-cluster are geographically close to each other. The number of subclusters of a cluster  $C_j$  is calculated based on the index  $j$ . The KD-tree algorithm is utilized to realize this partitioning. KD-tree is a k-dimensional binary search tree for information retrieval by associative searches. In this case, they used the 2D-tree where the two dimensions are the length and width of the sensor deployed field.

## 2] Particle Swarm Optimization

PSO is a kind of evolutionary computation proposed by Eberhart and Kennedy. The algorithm is inspired by the research of the social behavior of bird flocking. PSO simulates the behaviors of bird flocking. Suppose the following scenario: a group of birds are randomly searching food in an area. There is only one piece of food in the area being searched. All the birds do not know where the food is. But they know how far the food is during each of search iterations. So what's the best strategy to find the food? The effective one is to follow the bird which is nearest to the food. PSO learned from the scenario and used it to solve the optimization problems. In PSO, each single solution is a "bird" in the search space. They call it "particle". All of the particles have fitness values which are evaluated by the fitness function to be optimized, and have velocities which direct the flying of the particles. The particles are "flown" through the problem space by following the current optimum particles.

Fitness is used to evaluate the performance of particles in the swarm. Generally, choosing a proper objective function as fitness function to represent the corresponding superiority of each particle is one of the key factors for successful resolution of the relevant problem using PSO algorithm. The vehicle routing problem is a difficult combinatorial optimization problem, and generally can be described as follows: Data are to be delivered to a set of sensor nodes in each cluster by the mobile sink. The locations of the sensors and the clusters are given. The objective is to determine a viable route schedule which minimizes the distance or the total cost with the following constraints:

1) Each sensor nodes is served exactly once by exactly one sink;

Each sink starts and ends its route within a cluster;

The total length of each route must not exceed the constraint;

The total demand of any route must not exceed the capacity of the sink.

Therefore, they choose the following equation as fitness function:

$$f(x_i) = \sum_{i=1}^N \sum_{j=1}^N c_{ij} B_i B_j$$

Where  $N$  represents the number of nodes in a cluster,  $C_{ij}$  is the cost of traveling from sensor  $i$  to sensor  $j$  by sink  $k$ .  $B_i$  and  $B_j$  are the buffer overflow times of sensors  $i$  and  $j$  respectively.

Then by substituting the fitness function in to Algorithm for PSO, the particles are updated and the best visiting path for the sink within a cluster is determined.

### 3.2.2.2 Coordinated Relocation of Sink Nodes [1]

It is assumed that during the network bootstrapping phase the gateways will perform node discovery to establish contact with the largest number of deployed sensors. The gateways then divide the discovered sensor nodes among them, forming clusters. But this approach is independent from the clustering criteria which may be load balancing, communication range, etc. A gateway is responsible for managing the sensors in its clusters. A sensor node is capable of communicating with the gateway of its cluster either directly or over a multi-hop path. Nodes that are incapable of reaching any gateway are considered orphaned.

A gateway node is assumed to have motion capabilities, e.g a rover. It is assumed that gateways understand the layout of their own clusters in relative terms at least. They have also assumed that a gateway node has initial contact with all other gateways in the network. This algorithm will maintain this contact, which will ensure that gateway nodes (through inter gateway communication) will be able to build a global view of the entire network by sharing local views of clusters.

Gateway nodes do not need to be in direct transmission range of all other gateways in the network, but a path must be available that links each gateway to another.

#### 1] Problem Description

Gateway relocation is an effective scheme for optimizing the network operation within the individual cluster. The main issues are when it makes sense to move the gateway, where to place it and how to handle the packet traffic while the gateway is on the move.

However, the relocation problem is substantially compounded when a network consists of multiple clusters. A gateway node cannot choose to arbitrarily wander around its cluster to enhance the intra-cluster network operation (local gain) without considering the potential impact this could inflict on its ability to maintain communication with the gateway nodes of other clusters (global effect). In this context, local gain is defined as improvement to the number of sensor nodes in the cluster

(coverage), the average hops required to communicate with the sink node from a sensor node (cost of communication), and throughput of data to the sink node – all other relevant data can be determined from these metrics.

In terms of gateway-to-gateway communications, at the coarsest grained level, when a gateway node moves one of three things can happen. Links between gateway nodes can be broken, links between gateway nodes can be formed, or no change to the network occurs. When a link is broken between two gateway nodes, the resulting severed communication could affect the remaining gateways.

A synchronization protocol needs to be applied in order to ensure that a gateway bases its relocation decision on a consistent global state; particularly the location of all gateways.

#### 2] The Core Algorithm

CORE strives to counter the multi-gateway relocation issues. The goal is to enable optimization of the intra-cluster operation while ensuring the inter-gateway connectivity. Thus, the first phase of CORE involves locating a suitable place for a gateway node to relocate to in order to better serve its cluster. A number of schemes are proposed in the literature for finding such new location. A scenario is depicted in Figure. 2. Upon identifying the new location, CORE requires G to base its analysis on a consistent network state. That is being implementing by using a mutex. Basically, a global variable is defined and shared among the gateways (they are in the process of removing the need for this shared data resource by using distributed synchronization techniques). Access to this mutex is exclusive and granted to only one gateway at a time. Such exclusive access to this mutex will allow only one gateway to restructure the inter-gateway topology and prevent the potential of race conditions discussed earlier. While having the exclusive access to the mutex, G is sure that there is no change in the network state and can thus go forward to evaluate the inter-cluster ramifications of its move. If the move is deemed appropriate, i.e. it does not cause partitioning of the inter-gateway network, G can go ahead and relocate. G then releases the mutex and informs the other gateways about its new location. If, however, G finds that its move will sever the network, it will need to initiate a recursive process of requests for movement to attempt to keep the inter-gateway connectivity. To do this, G will have to examine exactly which communication links it would break by moving. For each of these links, G will need to make a request to the gateway node G' on the other end of the link. G' must then find a location to move to, in order to reestablish the link while maintaining good performance within its cluster. G' will strive to minimize the distance it travels in order to maintain other links it has. In case, G' loses connection with some of its neighbor gateways, it asks them to adjust their location and so on. To terminate the procedure, a gateway will be allowed to adjust its location only once.

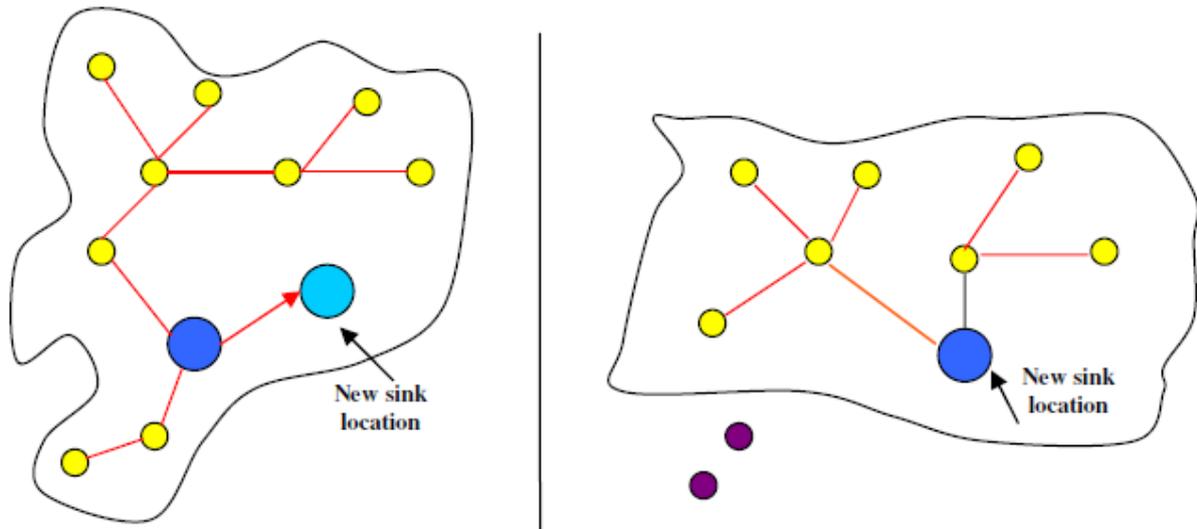


Figure.2 The movement of the gateway node restructures the Intracluster topology, leaving a few source nodes orphaned

At the functional level, the CORE algorithm consists of five parts. The outer CORE function, *EvaluateAndMove()* is responsible for initializing the recursion, maintaining the critical section, i.e. the *mutex*, and deciding to send the movement message. Inside *EvaluateAndMove()*, CORE needs access to the *FindNewLocation()* function. This function is responsible for identifying a desired location *inside* the gateway's cluster to move to. Following this, the *MoveToLocation()* recursive function is called. This function trickles down through the network of connected gateways, recalling itself for each subsequent link break and updating CORE's view of the virtual network via the *UpdateVirtualNetwork()*. The recursion also needs one other function, *FindHelpLocation()*, which calculates the closest location for a gateway to move to establish communication with a given location subject to intra-cluster constraints. When the recursion returns, *G* will compare the view of the network with the view presented in the virtual network. Only if the virtual network performs better in simulation based on recent network activity, *G* will actually move.

### 3.2.2.3 Deployment in event driven WSN [15]

#### 1] Network Model

The following simplifying assumptions are made in building the system model (Figure.1 illustrates the system model):

- Sensors remain stationary at the nodes of a bidimensional square grid composed of same-size cells.
- Sensor nodes are homogeneous and wireless channels are bi-directional, symmetric and error-free.
- Sensor nodes communicate with the sink by sending data via multiple hops along the shortest path; a hop is of one cell side length, i.e., the distance between two adjacent nodes in the grid equals the transmission range of nodes.
- Data transmission and reception are the major energy consuming activities.
- The sinks can move freely on the grid from one node to another in eight directions. After the sink arrives at a node, sensors can communicate with the sink. For analytical simplicity, the traveling time of the sink between two nodes is considered negligible, and the sojourn time of the sink visit at sensors is equal.

- The event occurs at any grid cross point independently with stationary distribution.

#### 2] Path Selection

If a sensor node *i* is neither co-located with sink *k* nor directly connected with it (i.e., if *k* is not co-located with any of the nodes in *S<sub>i</sub>*), then data packets generated at node *i* have to be relayed through multiple hops to reach the sink. The sink can only be located at one node position in the grid (the sensor locations and the possible sink locations are the same). The sink keeps moving among grid positions until the maximum network lifetime is reached, which occurs when one sensor node's residual energy drops below a predefined threshold required for it to operate (when this occurs the sensor "dies").

When a sensor node lies on the same horizontal or vertical line of the current position of the sink, a unique shortest path exists between the two nodes. Otherwise, multiple shortest paths exist.

### 3.2.3 Single and Multiple Mobile Sink Deployment

#### 3.2.3.1 Sink Repositioning using 1Hop Algorithm [3]

##### 1] Assumed Network Model

They assume a densely deployed, time-driven sensor network where every sensor sends a data message in every time period. The sensors use multi-hop communication, the messages travel to the closest sink using a geographical routing. They considered a distributed routing solution, where there is no central authority to select the end-to-end route and inform the participating nodes about it. It is up to the nodes to decide locally to whom the packet should be handed over. To choose the next hop among the neighbors within radio range, they applied the GOAFR routing algorithm. The GOAFR routing combines the *greedy* and the *face* algorithms. The *greedy* algorithm always picks the neighbor closest to the sink to be next node for routing. However, in certain situations it can occur that no neighbor is closer to the sink than the current node, for example if there is a "hole" in the sensor field. In this case the routing switches to the *face* algorithm, and passes around the hole on its border. When it is possible, the routing switches back to the *greedy* algorithm.

It is important, that all the data packets that are sent by a sensor has to arrive to the sink before it moves away. It can be ensured by that the sink stays at its broadcasted position for a certain

(guaranteed) amount of time, also included in the update message. They assume that data delivery fits safely into this guaranteed time period.

## 2] 1Hop Algorithm

In practical situations the *global* algorithm is hardly applicable (if applicable at all). In a wide-area network with huge number of nodes it is not feasible for a sink to collect and store the location information of every sensor. This problem made us to propose local information based multi-sink deploying algorithm called *1hop*. It uses only locally available information; the sinks know only the locations of the sensors that communicate directly with them and the locations of the other sinks. It also knows how many routes are passing through these sensors. Given an initial sink deployment, in the next step the sinks determine which sensors are communicating directly with them. Let  $H_j$  denote the set of the indices of these sensors, i.e.,

$H^j = \{i : d_i^{(j)} < r_c, i \in N\}$ , where  $r_c$  stands for the communication range of a sensor.

After that the sinks wait for a certain time period ensuring that within this time interval every sensor sends at least one message to the sinks. The header of every message contains the ID of the original sender node. When sink  $j$  receives a message from a sensor in  $H_j$  then it notes the ID of the original sender node, thus at the end of the period it can determine how many sensors are sending the messages through sensor  $t$ ;  $t \in H_j$ . Except the neighboring nodes the sinks do not know the positions of all the sensors in the network. Therefore, in order to minimize the average distance between the sensors and the nearest sinks they have to approximate somehow the location of the distant sensors. The approximation is done in the following way: let sensor  $t$  be the neighbor of sink  $j$  and denote  $nr_t$  the number of sensors that send their messages through sensor  $t$  toward sink  $j$ . Let  $route_{ij}$  denote the set of the indices of the sensors on the route from sensor  $i$  to sink  $j$ .

Then  $nr_t = \#\{i : t = \min_l d_i^{(l)}, t \in route_{jt}\}$ . Sink  $j$  assumes that there are  $nr_t$  sensors in the direction of sensor  $t$  and the resultant vector for sink  $j$  is approximated as:

$$r_j = \frac{\sum_{t \in H_j} e_t^{(j)} nr_t}{\sum nr_t}, j = 1 \dots k$$

If  $|r_j|$  is below a threshold then sink  $j$  remains at its present position  $S^j$ , otherwise moves to  $S^j + r_j \cdot \max\_step$ , where  $\max\_step$  stands for the maximal possible step for a sink. The method returns to the second step and the iteration continues in case there is a sink that moves. If every sink has stopped to move, the 1hop method ends.

In this paper they propose a novel method to relocate multiple sinks in a coordinated way during the operation of WSNs.

## 3] 1Hop Relocation Algorithm

The proposed method is called *1hop relocation* and it is derived from the *1hop* method. The sinks know their own locations, the locations of the other sinks and they can determine the locations of the neighboring sensors.

Initially the sinks are deployed in the network randomly. Next the sink relocation procedure begins. The sinks determine the sensors that are in one hop distance and get their locations, after that they collect the incoming messages over a time period  $t$ . Using the sender ID of the messages they count the number of routes passing through the neighboring sensors.

At the end of the time period they calculate the resultant vectors for every sink using. If the size of the vector for sink  $j$  is above the *threshold* parameter then the sink  $j$  moves to a new location. At the end of the procedure the method starts again and the iteration continues.

The difference between the *1hop* and *1hop relocation* methods is that the second one does not stop if none of the sinks were relocated. The reason for this is that *1hop* is used during the initializing phase of the network while the *1hop relocation* algorithm is used in the operational phase of the network.

### 3.2.3.2 Optimization Algorithm for WSN [16]

The dual problem is given by:

$$\text{Minimize} \sum_i E_i w(i) \quad (1)$$

$$\sum_i \lambda_i W(i, k) \geq 1 \quad \forall k \quad (2)$$

$$\sum_{j \in P, P \in P_{ik}} w(j)(e_j^{tx} + e^{rx}) - W(i, k) \geq 0 \quad \forall ik \quad (3)$$

$$w(j) \geq 0 \quad \forall i, j, k \quad (4)$$

Where  $W(i, k)$  is the weight assigned to a “commodity” (data flow injected at a node) from node  $i$  to the sink location during the  $k$ th epoch and  $w(j)$  is the weight assigned to a node  $j$ . The weight of a node  $w(j)$  represents the marginal “cost” of using an additional unit of energy of the node, and the weight of a commodity  $W(i, k)$  represents the marginal “cost” of rejecting a unit of demand of the commodity.

Let us denote the objective of the dual problem by  $G(w) = \sum_i E_i w(i)$ . In order to minimize  $G(w)$ ,  $w(i)$  should be as small as possible, but it is bounded from below by  $W(i, k)$  through (2) and (3). Taking an arbitrary assignment  $w$  and

$$W(i, k) = \sum_{j \in \min\{p | p \in P_{ik}\}} W(j)(e_j^{tx} + e^{rx})$$

(i.e., the length of the shortest path from  $i$  to  $k$ ), they meet (3). Then (2) becomes the following constraints:

$$\sum_i \lambda_i \left[ \sum_{j \in P, P \in P_{ik}} W(j)(e_j^{tx} + e^{rx}) \right] \geq 1 \quad \forall k$$

This assignment is not necessarily feasible because it might violate the above constraints. However, it can be made feasible by finding the most violated constraint and scale the assignment accordingly. In other words, if there is an oracle that identifies

$$\min_k \rho_k : \rho_k(w) = \sum_i \lambda_i W(i, k) < 1,$$

They can scale all assignments  $w(j), w(i, k), \forall i, j, k$  by  $[\min_k \rho_k(w)]^{-1}$  and make a feasible assignment. Therefore, the dual problem is equivalent to finding a weight assignment

$w: V \rightarrow R_0^+$  such that  $G(w)/\rho(w): \rho(w) \equiv \min_k \rho_k(w)$  is minimized. They denote  $\min_w [G(w)/\rho(w)]$  by  $\beta$ . Note that this interpretation of the dual problem already suggests a duality theorem analogous to the max-flow min-distance ratio theorem (which is in turn analogous to the max-flow min-cut theorem of Ford and Fulkerson for single s-t flow).

The algorithm proceeds in iterations. Let  $w_{i-1}, W_{i-1}$  be the weight assignment at the beginning of the  $i$ th iteration and let  $\{t_{k,i-1}\}$  be the time schedule after iterations  $1, \dots, i-1$ . In the  $i$ th iteration, they route  $\sum_l \lambda_l$  units of commodity along the paths (and thus to the corresponding sink location) given by an oracle (they will specify it later) that computes  $\min_k \rho_k(w)$  and let  $t_{k,i} = t_{k,i-1} + 1$ . Let  $f_i(l)$  be the flow through node  $l$  and  $p_{l,k}, \forall l$  be the paths suggested by  $\min_k \rho_k(w)$  in this iteration. The new weight assignment to a node  $l$  is given by  $w_i(l) = w_{i-1}(l)(1 + \epsilon f_i(l)(e_l^{tx} + e^{rx})/E_l)$ , and the new weight assignments to a commodity are computed as  $w_i(l, k) = \sum_{j \in p_{l,k}} w_i(j)(e_j^{tx} + e^{rx})$ .

Note that  $p_{l,k}$  is indeed the shortest path from  $l$  to  $k$  because it is suggested by the oracle that computes  $\min_k \rho_k(w)$ . Now the dual objective is updated as:

$$\begin{aligned} G(w_i) &= \sum_l E_l w_i(l) \\ &= G(w_{i-1}) + \epsilon \sum_l W_{i-1}(l) f_i(l) (e_l^{tx} + e^{rx}) \\ &= G(w_{i-1}) + \epsilon \cdot \rho(w_{i-1}) \end{aligned}$$

Initially, the weight assignment to a node  $l$  is  $w_0(l) = \delta / E_l$ . The iteration stops when  $G(w_i) \geq 1$  for the first time.

The oracle that computes  $\min_k \rho_k(w)$  is simply an extension of the Floyd-Warshall algorithm that computes all-pairs shortest path with a time complexity of  $\theta(n^3)$ . They organize the results of the Floyd-Warshall algorithm into “clusters”; each cluster includes paths that have a common end. Then they run a search algorithm in order to find the best “median”  $k$  that achieves  $\min_k \rho_k(w)$ . This oracle has a time complexity of  $\theta(n^3)$  (because the later clustering and searching both have a time complexity at least one order lower than that of the Floyd-Warshall algorithm).

### 3.3 Hybrid Sink Deployment

#### 3.3.1 Hybrid Sink Deployment for Data Collection [13]

In this paper, we have developed a hybrid sink positioning technique to achieve the above goals. The following are features of the proposed technique: It uses both static as well as

mobile sinks to gather the data from the sensor nodes. Initially the sensor nodes are categorized into two groups based on their residual energy and data generation rate. The nodes which are having low residual energy and high data generation rate are categorized as “urgent” and the nodes which are having high residual energy and low data generation rate are categorized as “non-urgent”. In order to collect the data from the “urgent” sensor nodes, they deploy a static sink within the centre of the network. Then they determine a set of “relay” nodes such that each urgent node has one relay node which is nearest to the static sink. The relay nodes are selected based on their residual energy and can be changed in each interval. The urgent nodes send their data to the static sink through these relay nodes. In order to collect the data from the “non-urgent” sensor nodes, they deploy a mobile sink which periodically collects the data from the nodes.

#### 1] Sensor Node Classification

Let  $RE_i$  be the Residual Energy and the  $GR_i$  be the Data Generation Rate of sensor  $S_i, i=1,2,\dots$ . Then nodes are classified as urgent and non-urgent based on the following two conditions.

Urgent, if  $RE_i < RE_{th}$  and  $GR_i > GR_{th}$   
Non-Urgent, if  $RE_i > RE_{th}$  and  $GR_i < GR_{th}$

Where  $RE_{th}$  is the minimum threshold value of the Residual Energy and  $GR_{th}$  is the maximum threshold value of the Data Generation Rate.

#### 2] Positioning Relay Nodes near Sink

They consider a two-tier network in a sensing field  $V$  with a

Static sink  $N$ ,  
Set of urgent sensors  $S = \{s_1, s_2, \dots, s_n\}$ , and  
Set of relay nodes  $R = \{r_1, r_2, \dots, r_e\}$ .

Every node has a short-range antenna and each sink and relay nodes has a long-range antenna with a transmission distance  $TD$ . Short-range antennas can only communicate with short range antennas, and so are long-range antennas. Our objective is to utilize the resource efficient relay nodes with high residual energy, to relay the sensory data from urgent sensors to  $N$ .

Initially, the set of urgent sensors  $S$  and the set of relay nodes  $R$  are randomly deployed in  $V$  such that  $S$  is densely deployed and  $R$  is sparsely deployed. A relay node that is connected to  $N$  in the high-tier network is called connected and it is un-connected otherwise. The location of any node in  $S$  and  $R$  is not known. Data collection is conducted by the cooperation of  $S$  and  $R$ . Let  $MR(s_i)$  is the set of primary relay nodes which are nearest connected relay nodes to  $s_i \in S$ .  $s_i$  initially send its sensory data along the low-tier network to  $MR(s_i)$ . Then  $MR(s_i)$  can relay the data to  $N$  via the high-tier network. Let  $H(m_i) = \{s_j | MR(s_j) = m_i\}$ , the set of sensors served by  $m_i$ , called  $m_i$ 's virtual Voronoi cell ('cell' for short).

#### 3] Mobile Sink Deployment Algorithm

They propose a local information based multiple mobile sink deployment algorithm in order to collect the data from the “non-urgent” sensor nodes. The mobile sinks periodically collect the data from the nodes. Let  $\{v_1, v_2, \dots, v_n\}$ , be set of non-urgent sensors.

The sinks identify only the locations of the  $v_i, i=1,2,\dots,n$ , which communicate directly with them and the locations of the other sinks by using the locally available information.

Except the neighboring nodes, the sinks do not know the positions of all the sensors in the network. It also knows how many routes are passing through these sensors.

Given an initial sink deployment, the sinks determine which non-urgent sensors are communicating directly with them. Let  $Q_j$  denote the set of ids of these sensors, such that  $Q_j = \{i : d_i^{(j)} < TD; i \in N\}$ . After that, the sinks wait for a certain time period during which every non-urgent sensor sends at least one message to the sinks. The header of every message contains the ID of the original sender node. When sink  $j$  receives a message from a sensor from  $Q_j$ , then it notes the ID of the original sender node. Thus at the end of the period, it can determine how many sensors are sending the messages through sensor  $k$ ,  $k \in Q_j$ . Therefore, in order to minimize the average distance between the sensors and the nearest sinks, they have to approximate the location of the distant sensors which is done in the following way:

Let sensor  $k_j$  be the neighbor of sink  $j$  and  $Z_k$  denotes the number of sensors that send their messages through sensor  $k$

toward sink  $j$ . Let  $route_{ij}$  denote the set of the ids of the sensors on the route from sensor  $i$  to sink  $j$ . Then  $Z_k = \#\{i : k = \min d_i^{(j)}, k \in route_{ik}\}$ . Sink  $j$  assumes that there are  $Z_k$  sensors in the direction of sensor  $k$  and the resultant vector for sink  $j$  is approximated as:

$$RV_j = \frac{\sum_{k \in Q_j} U_k^{(j)} \cdot Z_k}{\sum Z_k}, j = 1K \quad (5)$$

The unit vector pointing from the sink to sensor  $i$  is denoted by  $U_i$ . If  $|RV_j|$  is below a threshold then sink  $j$  remains at its present position  $s^j$ , otherwise moves to  $s^j + RV_j \cdot St_{max}$ , where  $St_{max}$  stands for the maximal possible step for a sink. The method returns to the second step and the iteration continues in case there is a sink that moves. If every sink has stopped its movement, the mobile sink deployment algorithm ends.

#### 4. COMPARATIVE STUDY OF SINK REPOSITIONING TECHNIQUES

The following table summarizes the comparative study of various sink repositioning techniques.

Table 1. Comparison of Sink Repositioning Techniques

Paper No	Algorithm	Deployment: Static/mobile/hybrid	Sink	Energy Consumption	Area Coverage	Connectivity	Techniques	Performance Metrics	Traffic Type
11	Dynamic optimized BS positioning method	Mobile	single	yes	no	No	Genetic algorithm	lifetime	
12	BS repositioning based on quality of route setup	Static	single	yes	yes	no	Route Setup	Delay, throughput, overhead, lifetime	Traffic density is considered
9	Routing for real time data	Static	single	yes	yes	yes	Route Setup and Traffic pattern	Throughput, delay, overhead, lifetime	Delay constrained, unconstrained traffic
6	MOSR algorithm	Mobile	Multiple	yes	no	no	PSO	Delay, cost, overhead, packet loss	
13	Mobile sink deployment algorithm	Hybrid	Multiple	yes	yes	no	Hybrid sink repositioning	delay	
1	CORE	Mobile	Multiple	yes	yes	yes	Distributed synchronisation	Throughput, cost	
14	ILP solver	Mobile	multiple	yes	no	no	Maximum Overlapping Region	lifetime	
3	1 hop algorithm	Mobile	Single and multiple	yes	no	no	Distributed routing technique	lifetime	
15	Event driven sink deployment	Mobile	multiple	yes	no	no	Optimal mobility trajectory	Lifetime, delay	
16	Optimization algorithm for increasing network lifetime	Mobile	Single and multiple	yes	no	no	Approximation algorithm and duality theorem	Lifetime	
17	Adaptive sink mobility in event driven clustered network	Mobile	single	yes	no	no	Clustering technique	distance	

## 4.1. Routing Metrics

- i. **Delay:** Defined as the average time a packet takes from a sensor node to the gateway [12], [9], [6], [13], [15].
- ii. **Area Coverage:** A geographic area within which a carrier provides service [12], [9], [13], [1].
- iii. **Connectivity:** The notion of area coverage can be considered as a measure of the quality of service (QoS) in a sensor network, for it means how well each point in the sensing field is covered by the sensing ranges [9],[1].
- iv. **Energy:** This metric represents the average energy consumed in transmitting, and receiving a data packet [11], [12], [9], [6], [13], [1], [14], [3], [15], [16], [17].
- v. **Throughput:** Defined as the total number of data packets received at the gateway divided by the simulation time [12], [9],[1].
- vi. **Overhead:** is generally considered any combination of excess or indirect computation time, memory, bandwidth, or other resources that are required to attain a particular goal [12],[9],[6].
- vii. **Cost:** The cost calculated is a function of communication energy dissipated in transferring certain bits of data over a particular distance [6], [1].
- viii. **Packet Delivery Ratio:** It is the ratio of the number of packets received successfully and the total number of packets sent [6].
- ix. **Lifetime:** This gives a good measure of the network lifetime [12], [9], [14], [3], [15], [16].

## 4.2 Limitations

### 4.2.1 Sink Deployment Limitations

- 1] In [12], the technique does not allow transmission of packets during movement of the gateway when the gateway goes out of the transmission range of the nodes.
- 2] In [12], the work does not plan on the repositioning of the gateway under QoS traffic in order to provide better QoS.
- 3] In [1], the overhead analysis during sink relocation is not considered.
- 4] In [3], this work considers only the time driven network operations whereas the event driven network operations are not considered.
- 5] In [3], the actual power level of the sensor during the relocation process is not taken into account.

### 4.2.2 Limitation of Existing Survey

- 1] Little survey work has been done in the sink repositioning topic till date.
- 2] In the existing surveys, the hybrid deployment (static and mobile) of sink is not considered.
- 3] The impact of performance metrics like delay, throughput, cost, etc is not discussed or analyzed in any of the existing survey work.

### 4.2.3 General limitations of Existing sink deployment techniques

- 1] In most of the works, the performance metrics such as average packet delivery ratio and packet loss are not considered.
- 2] Only two of the existing work considers all of the objectives like power consumption, connectivity and area coverage.
- 3] None of the existing works consider the buffer overflow problem in the case of the mobile sinks.
- 4] Different type of the traffic is not considered in most of the existing works.
- 5] Very less work is done in the hybrid sink deployment technique.

## 5. CONCLUSION

In this paper, we have analyzed and compared various sink deployment techniques in wireless sensor networks. From this survey, we come to the conclusion that existing works not concentrate on the hybrid sink deployment technique and only very few works consider all the major objectives like power consumption, area coverage and connectivity. In the case of the mobile sinks, none of the work considers the buffer overflow problem which is due to the dropping of data at the sensors due to the incorrect visiting schedules of the sinks. Also the type of the traffic (real-time or non real-time) and its impact on relocation are not considered. Hence a hybrid sink deployment technique which considers different types of traffic and covers all the objectives like power, area coverage and connectivity has to be developed.

## 6. REFERENCES

- [1] Jesse English., Michael Wiacek., and Mohamed Younis., “CORE: Coordinated Relocation of Sink Nodes in Wireless Sensor Networks”, June 2008.
- [2] Elham Hajian., Kamal Jamshidi., and Ali Bohlooli., “Improve Energy Efficiency Routing in WSN by using Automata”, International Journal of Ad hoc, Sensor & Ubiquitous Computing (IJASUC) Vol.1, No.2, June 2010.
- [3] Zoltan Vincze., Rolland Vida., and Attila Vidacs., “Deploying Multiple Sinks in Multi-hop Wireless Sensor Networks”, Pervasive Services, IEEE International Conference, July 2007.
- [4] F. L. Lewis., “Wireless Sensor Networks”, To appear in Smart Environments: Technologies, Protocols, and Applications, 2004.
- [5] Xu Li., Amiya Nayak., and Ivan Stojmenovic., “Sink Mobility in Wireless Sensor Networks”, 16 June 2010.
- [6] Rajeev Paulus., and Gurmit Singh., “A Multi Objective Sink Repositioning Algorithm Based on Particle Swarm Optimization in Wireless Sensor Networks”, American Journal of Scientific Research, ISSN 1450-223X Issue 11(2010), pp.159-170, 2010.
- [7] Yimin Chen., Long Cheng., Canfeng Chen., and Jian Ma., “Wireless Sensor Network for Data Sensing in Intelligent Transportation System”, 2009.
- [8] Mohamed Younis., and Kemal Akkaya., “Strategies and Techniques for Node Placement in Wireless Sensor Networks: A Survey”, Elsevier, 2007.

- [9] Kemal Akkaya., Mohamed Younis., and Meenakshi Bangad., “Sink repositioning for enhanced performance in wireless sensor networks”, Science Direct, 2005.
- [10] Zoran Bojkovic., and Bojan Bakmaz, “A Survey on Wireless Sensor Networks Deployment”, WSEAS TRANSACTIONS on Communications, ISSN: 1109-2742, Issue 12, Volume 7, December 2008.
- [11] Amir Mollanejad., Leili Mohammad Khanli., and Mohammad Zeynali., “DBSR: Dynamic base station Repositioning using Genetic algorithm in wireless sensor network”, IJCSI International Journal of Computer Science Issues, Vol. 7, Issue 2, No 2, March 2010.
- [12] Mohamed Younis., Meenakshi Bangad., and Kemal Akkaya., “Base-Station Repositioning For Optimized Performance of Sensor Networks”, Vehicular Technology Conference, 2003. VTC 2003-Fall. 2003 IEEE 58th, 2003.
- [13] Prerana Shrivastava, Dr. S. B. Pokle,” A Hybrid Sink Positioning Technique for Data Gathering in Wireless Sensor Networks”, International Journal of Engineering and Innovative Technology (IJEIT) Volume 1, Issue 3, March 2012
- [14] Waleed Alsalih., Selim Akl., and Hossam Hassanein., “Placement of multiple mobile base stations in wireless sensor networks”, Signal Processing and Information Technology, 2007 IEEE International Symposium, 2007.
- [15] Bin WANG., Dongliang Xie., Canfeng Chen., Jian Ma., and Shiduan Cheng., “Deploying Multiple Mobile Sinks in Event-Driven WSNs”, 2008.
- [16] Jun Luo., and Jean-Pierre Hubaux., “A Unified Framework for Joint Sink Mobility and Routing to Increase the Lifetime of Wireless Sensor Networks”, June 2008.
- [17] Z. Vincze., D. Vass., R. Vida., and A. Vidács1., “Adaptive Sink Mobility in Event-driven Clustered Single-hop Wireless Sensor Networks”, July 2006.

## **AUTHOR’S PROFILE**

**Mrs. Prerana Shrivastava** obtained her Bachelor’s degree in Electronics Engineering from University of Nagpur, India. Then she obtained her Master’s degree in Electronics Engineering and currently pursuing her PhD in Wireless Sensor Networks from G.H.Raisoni College of Engineering, Nagpur, India. She is working as an Assistant Professor in Electronics Department at Lokmanya Tilak College of Engineering, University of Mumbai, India. Her specializations include Electromagnetic Wave Theory, Image Processing and Computer Networks. Her current research interests are Wireless Adhoc Networks, Wireless Sensor Networks and Network Security.

**Dr. Sanjay. B. Pokle** obtained his Bachelor’s degree in Electronics and Telecommunication Engineering from Govt. College of Engineering, Pune University, India in 1993. He then obtained his Masters degree in Electronics Engineering and also Ph. D. in Electronics from Visvesvaraya National Institute of Technology Nagpur, India. His research area includes designing aspects of MIMO-OFDM Wireless Communication Systems and Wireless channel Estimation Algorithms. He has published research papers in the reputed national and international Journals and also has presented papers in the reputed national and international conferences. He has guided several projects in the area of signal processing, Digital image processing, Artificial intelligence etc. at post graduation level. He is member of technical societies like ISTE and IEEE. He has total 18 years of experience which includes 3 years industry and 15 years of teaching experience. Presently he is working as Professor and Head of Electronics & Communication Engineering Department, Shri Ramdeobaba College of Engineering and Management, Nagpur.