Efficient Selection Scheme for Data Processing in Wireless Sensor Networks

Ahmed A.A. Gad-ElRab
Faculty of Science
Al-Azhar University - Cairo, Egypt

Doaa M. Alhilaly
Faculty of Science
Al-Azhar University - Cairo, Egypt

ABSTRACT

Recently, many data processing applications in wireless sensor networks (WSNs) works efficiently by using a coverage percentage of a target sensing area and a satisfaction percentage of collected data. Therefore, the whole coverage and complete satisfaction are not needed. As a result, finding new data processing techniques that can successfully minimize the data traffic and energy consumption for maximizing the network lifetime are required. In addition, using clustering with data processing techniques is an effective topology control approach in wireless sensor networks, which can increase network scalability and lifetime. In this paper, a (α -cov, β -sat) data processing problem is introduced and a new mobile agent clustering data processing methods are proposed. The proposed methods use a clustering with a mobile agent to cover α percentage of the target area such that the satisfaction percentage of collected data is β percentage. Simulation results show that the proposed methods achieve higher improvements in network lifetime, load balance and energy consumption than the existing methods.

Keywords
Mobile agent, cluster head , partial coverage, satisfaction

1. INTRODUCTION
Continued advances of MEMS and wireless communication technologies have enabled the deployment of large scale wireless sensor networks (WSNs).The applications of WSNs are highly varied, such as Military applications, Environmental applications, Health applications, Home applications and Commercial applications. Sensor nodes are battery powered and energy is the most crucial resource. The main task of a sensor node in a sensor field is to detect events, perform quick local data processing, and then transmit the data. One of the most important constraints on sensor nodes is the low power consumption requirement. Power consumption can hence be divided into three domains: sensing, communication, and data processing. Development of data processing in (WSNs) Comes from cooperation communication, including energy efficiency, scalability and reliability[14], these are the challenges that the focus of the researchers, most of researches focus on prolonging the network lifetime, allowing scalability for a large number of sensor nodes or supporting fault tolerance (e.g., sensors failure and battery depletion) [15], [16].

Clustering in WSN :is an effective topology control approach in wireless sensor networks, which can increase net-work scalability, stability and reduces power consumption by reducing flooding . In clustering schemes, large number of sensor nodes are densely deployed, neighbor nodes may be very close to each other. Hence, multihop communication in sensor networks between a sensor node and its designated cluster head CH is expected to consume less power than the traditional single hop communication. Furthermore, the transmission power levels can be kept low, which is highly desired in covert operations. Multihop communication can also effectively overcome some of the signal propagation effects experienced in long-distance wireless communication. The selected CHs collect data from member nodes in their respective clusters, aggregate the data, and send it to a base using multihop communication.

One the critical challenges in (WSNs) is overwhelming data traffic. To meet this challenge, one of the unique features of WSN applications is the necessity of cooperation. This cooperation come from using mobile agents reduce the network traffic, provide an effective means of overcoming network latency and helps you to construct more robust and fault-tolerant. Generally speaking an MA is a special kind of software that can execute autonomously, with identification, itinerary, data space and method as its attributes. For example, the sensory data of two closely located sensors are likely to have redundant or common parts when the data of two sensors are merged. Therefore, data aggregation is a necessary function in densely populated sensor networks in order to reduce the sensory data traffic.

This paper extends the previous work in [21] which was proposed by the authors. In [21] a mobile agent cluster-based algorithms by combining mobile agent (MA) and clustering scheme for data processing was proposed. As a result, data flooding can be reduced by clustering and a large amount of sensory data can be reduced by eliminating data redundancy by using MA which selects its visiting set of nodes by using satisfaction percentage of each node. However, the using of all clusters of the network in the data processing to maintain a certain application task leads to consume more energy consumption and thus it reduces the life of the network. So, in this paper, to use a required number of clusters in the data processing, a new concept which is called (α -cov, β -sat) will be introduced. Also, a new mobile agent clustering data processing methods will proposed. The proposed methods use a clustering with a mobile agent to cover α percentage of the target area such that the satisfaction percentage of collected data is β percentage. These required, α and β percentages are defined by a sink. The (α -cov, β -sat) means that the percentage of the total covered area of all clusters at any sensing round is larger than or equal to α percentage of the target area such that the total collected data at the sink satisfy the required satisfaction β percentage.

2. RELATED WORK

There are a lot of schemes have been proposed for data processing in WSNs [1-21]. In [2] author present a cooperative data processing algorithm based on mobile agent (MA-CDP) and considers MA in multihop environments, can autonomously clone and migrate themselves in response to environmental changes. In [3], the agent design in (WSNs) is decomposed into four components architecture, itinerary planning, middleware system design, and agent cooperation. Among the four components, itinerary planning determines the order of source nodes to be visited during agent migration, which has a significant impact on energy performance of the MA system. It has been shown that finding an optimal itinerary is a NP-hard problem. Therefore, heuristic algorithms are generally used to compute competitive itineraries with a suboptimal performance.

In [5], two simple heuristics are proposed: (i) a local closest first scheme that searches for the next node with the shortest distance to the current node and (ii)a global closest first scheme that searches for the next node closest to the dispatcher. These two schemes only consider the spatial distances between sensor nodes and thus may not be energy efficient in many cases. In [6] author propose a distributed randomized multihop clustering algorithm to organize the sensor nodes in a WSN into clusters. In [7] Sharma and Mazumdar have investigated the use of limited infrastructure, that is, networks with a number of wired connections between sensor nodes. Their approach establishes a small-world graph by utilizing wired links between a subset of nodes to reduce the overall energy demands as well as the different energy consumption rates of participating nodes. The additional efforts required for the wiring however make it suited for long-term deployments of sensor networks only. In [9] the authors have proposed energy-efficient hierarchical clustering algorithm (EEHCA) for WSN which improves the performance of LEACH and HEED [10]. In[11] authors have studied LEACH scheme and proposed two new schemes (i.e., energy-LEACH and multihop LEACH). Energy-LEACH improves the CH selection method and multi-hop LEACH (M-LEACH) improves the communication mode from single-hop to multi-hop between CH and BS. Both the schemes have better performance than LEACH scheme.

Most of the existing algorithms does not always guarantee the best sequence of nodes to be visited and consumed energy by the flooding. Most of existing works do not meet the required challenges where some of them can not maximize net-work life time and others generate high network traffic, so the previous assumptions in [21] are used to solve: (1) overwhelming data traffic(sending data to one place)by clustering (2) energy consumption (sending a query to all network as flooding) by an MA selectively migrates among clusters and sensor nodes inside each cluster by moving the processing function to the target clusters and nodes and performs local processing by using resources available at CHs and local nodes rather than bringing the data to a central processor (sink). The problem of [21] is the exploitation of all network clusters. To overcome this problem, a new concept which is called (α-cov, β-sat) is proposed. Then based on this concept, a new data processing methods are proposed to meet the challenges of data processing in WSNs.

3. (α-COV, β-SAT) DATA PROCESSING PROBLEM IN WSNs

In general, performing data operations inside the network, such as eliminating irrelevant records and aggregating raw data, can reduce energy consumption and improve sensor network lifetime significantly. This is referred to as in-network data processing, in which an intermediate proxy node is chosen to house the data transformation function to consolidate the sensor data streams from the data source nodes, before forwarding the processed stream to the sink. Data processing techniques that achieve this goal on sensor nodes are required to operate while meeting resource constraints such as memory and power to prolong a sensor networks lifetime. Data processing techniques improve the energy efficiency and it is a typical parameter measure of performance in sensor networks.

3.1 Definition , Assumptions, and Models

In this paper, the sensor network is divided into clusters, to create an independent clusters in data aggregation and processing processes. Each cluster has a cluster head which identifies all nodes in his cluster, contacts with another cluster. In each cluster sensors are uniformly dispersed within a rectangular field. All clusters ,sensors and the BS are stationary after deployment and the location of the BS is known by each cluster head and each cluster head is known for nodes that they belong to it. All sensors are homogeneous, i.e., they have the same capabilities as energy, memory, and communication. There is a mobile agent, MA, which is a special process that can autonomously migrate and aggregate data across nodes. MA systems provide new capabilities for energy-efficient data processing by flexibly planning its itinerary for facilitating agent-based data collection and aggregation. To determine MA path to mobility between clusters. MA with memory is provided for storing previous queries to make cluster more effective with nodes failure. Each application task request needs R rounds to collect a required sensing data. Each application task request has a required coverage and satisfaction are denoted by α-cov and β-sat, respectively. This means that this request needs to cover at α-cov percentage of the target area such that total collected data at the sink must satisfy at least β-sat percentage of the whole request needs. Here, the covered area of each sensor is defined as follows.

$$\text{cov}(i) = \pi r_i^2 \qquad (1)$$

where $\text{cov}(i)$ is the covered area for node i and r_i is the radius of the circle surrounding the node i. The total area which is covered by the cluster $C_{\text{cov}(ch)}$, is calculated as follows.

$$C_{\text{cov}(ch)} = \sum_{i \in ch} \text{cov}(i) \qquad (2)$$

Also, there is a satisfaction percentage for each node *i* called *sat(i)* which represents the ability percentage of node to satisfy the requested task based on its available memory and its remaining energy and is defined as follows.

$$sat(i) = \frac{e_i M + m_i E}{2ME} \qquad (3)$$

where M is the maximum available memory threshold and E is the maximum available energy threshold. By using sat(i), a satisfaction percentage of each cluster $C_{\text{cov}(ch)}$ which represents the ability of a cluster, ch, to satisfy a percentage of requested task needs based on available memory and energy of its member nodes and is defined as follows.

$$Csat(ch) = \sum_{i \in c_{ch}} sat(i) \tag{4}$$

Based on this model, there is a wireless sensor environment which contains a set of sensor nodes *V and* Set of clusters heads *CH* and there is C_{ch} Set of nodes in *ch's* cluster where $V = \sum_{n \in C_{ch}} C_{ch}$. An information generation rate at sensor node i is denoted as Q_i . A set of neighboring nodes of sensor i is denoted as S_i . e_{ij}^T and e_{ji}^R denote for the consumed energies at sensor i to transmit a data unit to sensor j and energy consumed at sensor j to receive the data unit transmitted by sensor i , respectively. q_{ij} and $q_{ij}^{\max}$ denote for a rate at which information is transmitted from sensor i to its neighboring node j and a maximum possible rate at which information can be transmitted from sensor i to its neighboring node j , respectively. A maximum power that sensor node i can spend at every round is denoted as p_i and p_{ch} are Maximum power that $ch \in CH$ can spend to every round . An initial amount of battery energy at sensor node is denoted as E_i . Also, E_{ch} Initial amount of battery energy at *ch* node. A number of aggregation rounds as R and a number of collection times by node i are denoted as t_i , t_{ch} . Number of collection times of each cluster head. q_{ch}^s and $q_{ch}^{\max}$ denote for the rate at which information is transmitted from *ch* to sink node S and Maximum possible rate at which information can be transmitted from ch to sink node s, respectively. Also, ge_{ch}^T denotes for consumed Energy at *ch* to transmit a data unit to sink node S and R is Number aggregation rounds.

Based on this model ,the consumed energy for a node i is defined as follows:

$$CE(V) = \sum_{n \in C_{ch}} \sum_{j \in S_i^{C_{ch}}} e_{ij}^T q_{ij}^{C_{ch}} t_i + \sum_{n \in C_{ch}} \sum_{j:i \in S_j^{C_{ch}}} e_{ji}^R q_{ji}^{C_{ch}} t_i \tag{5}$$

and for a cluster *ch* as follows:

$$CE(CH) = \sum_{c \in C} \sum_{ch \in CH} ge_{ch}^T q_{ch}^s t_{ch} \tag{6}$$

3.2 Problem Formulation
In this paper, the research motivation is maximizing overall sensor network lifetime by minimizing the total energy consumption of a WSN such that the required percentage covered α -cov of the target area and the required satisfaction

percentage β -sat of the application request needs are met. .To meet this objective, a number of collection times for each node $i \in V$ and each cluster head $ch \in CH$ are denoted as t_i and t_{ch} , respectively by minimizing t_i and t_{ch} for each application task request, the total energy consumption of a network will be minimized. As a result, the overall network lifetime will be maximize. Therefore, our problem is how to minimize the collected times t_i , of a node i and the collected times t_{ch} of each cluster head ch In other words, How to find the most suitable number of clusters that will operate in each round $r \in R$ such that this selected set of clusters and nodes must meet the required percentage covered α -cov of the target area and the required satisfaction percentage β -sat of the application request needs. This problem is the combinatory optimization problem. Linear programming is used to formulate this optimization problem which will capable of expressing network lifetime in terms of the number of collection times of a node and a cluster head. So, the problem of maximizing the overall network lifetime can be written as follows.

Minimize $\qquad CE(V) + CE(CH)$ (7)

subject to:

$$t_i, t_{ch} \geq 0 \tag{8}$$

$$0 \leq q_{ij}^{C_{ch}} \leq q_{ij}^{\max}, i \in V, j \in S_i^{C_{ch}} \tag{9}$$

$$CE(V) \leq E_i, i \in V, j \in S_i^{c_{ch}} \tag{10}$$

$$\sum_{j \in S_i^{C_{ch}}} e_{ij}^T q_{ij}^{C_{ch}} + \sum_{j:i \in S_j^{C_{ch}}} e_{ji}^R q_{ji}^{C_{ch}} \leq p_i, i \in C_{ch} \in V \tag{11}$$

$$\sum_{j:i \in S_j^{C_{ch}}} q_{ji}^{C_{ch}} + Q_i = \sum_{j \in S_i^{C_{ch}}} q_{ij}^{C_{ch}} i, j \in C_{ch} \tag{12}$$

$$0 \leq q_{ch}^s \leq q_{ch}^{\max} \tag{13}$$

$$\frac{\sum_{ch \in sc_x} C \text{cov}(ch)}{|sc_x|} \geq \alpha - \text{cov}, 1 \leq x \leq R \tag{14}$$

$$\frac{\sum_{ch \in sc_x} Csat(ch)}{|sc_x|} \geq \beta - sat, 1 \leq x \leq R \tag{15}$$

$$CE(CH) \leq E_{ch}, ch \in CH \tag{16}$$

$$\sum_{ch \in CH} ge_{ch}^T q_{ch}^s \leq p_{ch}, ch \in CH \tag{17}$$

Equation (8) describes that a number of collection times at any sensor node i and a number of collection times at any cluster head ch is larger than or equal 0. In other words, t_i and t_{ch} must be a positive value . Equation (9) represents the

rate at which information is transmitted from sensor i to its neighboring node j and to its cluster head inside cluster must be less than or equal to maximum possible rate at which information can be transmitted from sensor i to its neighboring node j inside cluster and must be greater than or equal to zero. Equation (10) represents the total energy consumed at sensor *i* during network operation is the sum of the quantities in (5) all times in collection must be less than or equal to initial amount of battery energy at sensor node $i \in V$. Equation (11) means that the power and energy expenditures are directly proportional to the rate at which information is transmitted/received. Given the number of times of collection of each node t_i and the information transfer rates q_{ij} the power (energy per time unit) consumed at sensor node i must be less than or equal maximum power that sensor node $i \in C_{ch}$ can spend inside cluster. Equation (12) represents the total incoming information transfer rate plus the information generation rate at a sensor node equals the total outgoing information transfer rate from the sensor. Equation (13) represents the rate at which information is transmitted from ch to sink must be less than or equal to maximum possible rate at which information can be transmitted from ch to sink and must be greater than or equal to zero. Equation(14)represents the total average coverage of selected clusters set sc_x at any round $x \in R$ and it must be larger than or equal to the required covered α-cov. Equation (15) represents the total average satisfaction of selected clusters set sc_x at any round $x \in R$ and it must be larger than or equal to the required satisfaction β-sat. Equation (16)represents the total energy consumed at sensor ch during network operation is the sum of the quantities in (6) all times in collection must be less than or equal to Initial amount of battery energy at ch node $ch \in CH$ .Equation (17) represents the power (energy per time unit) consumed for ch to transmitted data unit to sink must be less than or equal to maximum power that $ch \in CH$ .can spend to every round .Since that ch sends only to sink and don't receives from it, there is an energy-consuming to send only.

4. EFFICIENT SELECTION SCHEME FOR DATA PROCESSING (ESSDP)

4.1 Basic Idea

The basic idea is based on: (1) using a mobile agents (MAs) to move among nodes inside each cluster and among existing clusters heads, (2) dividing a set of member nodes inside each cluster *ch* into independent sets. These independent sets are activated successively, such that at any round $r \in R$ only one set is active. The nodes from the active set will be in their active state and all other sensors are in a low-energy sleep state, (3) using a required satisfaction percentage which is requested by application task through a sink. In addition to these assumptions, the required coverage percentage (α-cov) is assumed which is determined through the sink in order to avoid the consumption of the whole network. Sink determines the achieved satisfaction percentage (β-sat) and percentage of total area (α-cov). In case of the (α-cov) is not achieved but the (β-sat) is achieved, the number of selected nodes is accepted and the system will not select additional nodes.

4.2 The Proposed Selection Methods

Based on MA, clustering , α-cov and β-sat, a new scheme called *Efficient Selection Scheme for Data Processing Scheme* (ESSDP) is proposed. To minimize the total energy consumed for a cluster head and a node, ESSDP has three different selection methods. The first method is random selection which select the set of processing nodes randomly from the evaluated set of nodes by using the previous cost model with the required α-cov and β-sat. The second method is unsorting method which selects the set of processing nodes one by one from the evaluated set of nodes without any sorting. The third method is sorting method which selects the set of processing nodes one by one from the evaluated set of nodes after sorting this list in descending order based on the satisfaction of each node in the list. The three methods are described as follows.

Method 1:

(α-cov, β-sat) Random Selection

In this method, the proposed algorithm will determine a disjoint sets of clusters that will be visited by MA where each set will do online data aggregation in a round. This sets of clusters not exceeding area designated by sink(α-cov) and satisfy (β-sat). Each set will be activated for online data aggregation in round $r \in R$. Both sets of clusters and nodes are determined randomly and the coverage area of both sets of clusters and nodes is not larger than the required coverage area which is designated by sink(α-cov).

Algorithm 1:
$\alpha\beta$-*Random Selection[*$\alpha\beta$*-RC]*

Input: Required satisfaction β and required covered area α .

Steps:

1A. Sink broadcasts the query packet and required satisfaction β-sat and required covered area α to $ch \in CH$.

1B. $ch \in CH$ broadcasts the query packet and required satisfaction β to nodes inside cluster.

1C. $ch \in CH$ will construct a set of nodes by selecting these nodes randomly one by one from a set of nodes of *ch* and check if they can achieve the required satisfaction β .

1D. Nodes which are selected will be visited by MA.

1E. These nodes will calculate *cov(i)* and *sat(i)* and send it to $ch \in CH$.

1F. each ch calculates its covered area Ccov(ch) according to equations (2) and its satisfaction CSat(ch) according to equation (4).

1G. sink will construct a set of clusters for a round $r \in R$ by selecting these cluster randomly one by one and check if they can achieve the required satisfaction β and its coverage meets the required coverage α .

Output: independent sets of nodes for each cluster *ch*,

Method 2:

(α-cov, β-sat) Unsorting Satisfaction Selection

In this method, the proposed algorithm will determine a independent sets of clusters and sets of nodes according unsorting Satisfaction Selection method by selecting nodes one by without doing any sorting on the list of members of each cluster (nodes will be used for only once). This sets will be visited by MA inside each cluster where each set will be activated for online data aggregation in round $r \in R$. The coverage area of both sets of clusters and nodes is not larger than the required coverage area which is designated by sink(α -cov).

Algorithm 2:

$\alpha\beta$ *-Unsorting Satisfaction selection [$\alpha\beta$ -USC]*

Input: Required satisfaction β and required covered area α .

Steps:

2A. Sink broadcasts the query packet and required satisfaction β -sat and required covered area α to $ch \in CH$.

2B. $ch \in CH$ broadcasts the query packet and required satisfaction β to nodes inside cluster.

2C. $ch \in CH$ will construct a set of nodes and check if they can achieve the required satisfaction β this nodes use One-time and don't repeat in another round until the task is finish.

2D. Nodes which are selected will be visited by MA.

2E. These nodes will calculate *cov(i)* and *sat(i)* and send it to $ch \in CH$.

2F. each ch calculate its covered area *Ccov(ch)* according to equations(2) and its satisfaction *CSat(ch)* according to equation(4).

2G. sink will construct a set of clusters for a round $r \in R$ and check if they can achieve the required satisfaction β and its coverage meets the required coverage α .

Output: independent sets of nodes for each cluster *ch*,

Method 3:

(α -cov, β -sat) sorting Satisfaction Selection

In this method, the proposed algorithm will determine a independent sets of clusters and sets of nodes according to required satisfaction value β -sat by sorting the satisfaction values of all nodes in each cluster in descending order. This nodes will be visited by MA inside each cluster where each set will be activated for online data aggregation in round $r \in R$. The coverage area of both sets of clusters is not larger than the required coverage area which is designated by sink(α -cov).

Algorithm 3:

$\alpha\beta$ *-Sorting Satisfaction selection [$\alpha\beta$ -SC]*

Input: Required satisfaction β and required covered area α .

Steps:

3A. Sink broadcasts the query packet and required satisfaction β and required covered area α to $ch \in CH$.

3B. $ch \in CH$ broadcasts the query packet and required satisfaction β to nodes inside cluster.

3C. $ch \in CH$ will construct a set of nodes by selecting these nodes in descending order one by one from a set of nodes of ch and check if they can achieve the required satisfaction β .

3D. Nodes which are selected will be visited by MA.

3E. These nodes will calculate cov(i) and sat(i) and send it to $ch \in CH$.

3F. each ch calculates its covered area *Ccov(ch)* according to equations(2) and its satisfaction *CSat(ch)* according to equation (4).

3G. sink will construct a set of clusters for a round $r \in R$ by selecting these clusters in descending order one by one from a set of clusters of $ch \in CH$ and check if they can achieve the β and its coverage meets the required α .

Output: independent sets of nodes for each cluster *ch*,

4.3 Data Processing Steps
By using (ESSDP) algorithm, we will describe the steps of

data processing scheme as follows. 1) Generally, a sink receives a task request assigned by an application, sink broadcasts the query packet and α -cov and β -sat percentage to $ch \in CH$ directly or by getaway nodes, sink will deals with required α -cov therefore all clusters does not share this is reduces power consumption for nodes. 2) $ch \in CH$ creates its independent sets of nodes by using $\alpha\beta$ *-RC*, $\alpha\beta$ *-USC*, or $\alpha\beta$ *-SC* methods based maximum number of rounds R and required β -sat. 3) For all rounds R, each cluster among the target source nodes to be visited $ch \in CH$ will choose the first and last source nodes According: the size of an MA is the minimum in FirstNo while it becomes the maximum in LastSrc so the target source hich is the last (first) to send exploratory messages to the sink is chosen as FirstNo (LastSrc). When the MA arrives at the first source node, it is stored in it to construct another MA from its memory and dispatch it to initiate the new round. In the first round MA moves from source to source to collect and aggregate information and also copies processing code into the memory of each source node so MA does not carry the processing code any more in the following rounds When the whole task is finished, all the source nodes will discard the processing code When MA in firstNo,the sequence of visiting the other source nodes is dynamically decided by each target sensor based on its ToSourceEntry. ToSourceEntry is used for MA to roam among source nodes, target source nodes send exploratory messages enable sensor nodes to set up ToSourceEntry inside the cluster only. Finally, the MA will carry the data results, covered area and satisfaction for each node in the cluster to the $ch \in CH$ along the reinforced path. 4) In case of set of clusters that can satisfy both α -cov and β -sat, *ch* send data to sink. While in case of set of clusters that can satisfy the required β -sat but does not satisfy the α -cov , the number of selected nodes is accepted and the system will not add additional nodes. 5) Finally, After that MA moves among clusters that achieve both α -cov and β -sat to combine and aggregate data and finally to the sink.

5. EXPERIMENTAL RESULTS AND DISCUSSION

In order to demonstrate the performance of ESSDP, it is compared with the traditional random method, TR [2]. The OMNet++ (A. Varga, etal, 2001), simulator was used to evaluate our proposed algorithm (Shukui Zhang, et al, 2012), for discrete event. In the simulation model, the sink and sensor nodes were stationary. Also, a fixed number of rounds R is used, and it was 10 in all simulation experiments. The three different proposed methods, $\alpha\beta$ *-RC*, $\alpha\beta$ *-USC*, and $\alpha\beta$ *-SC* were applied and in each method sink determines the required covered area α -cov and the required satisfaction β -sat. Finally, the results was compared with TR. The simulation results indicate that the the three proposed methods are less in energy consumption, the death of the nodes, and average working times, taking into account that a particular area of the total area of the network is operating. In the rest of this section, the discussion of the simulation results will be introduced in details.
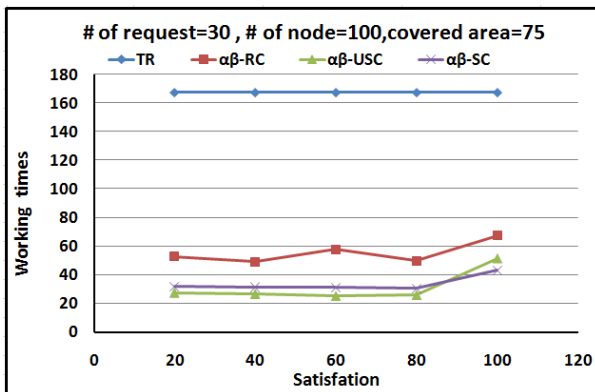

Fig. 1. the relation between required satisfaction and average working times

5.1 The average working times of a node

Figure 1 shows the average working times against the required satisfaction where the number of requests was 30 ,the number of nodes was 100 and required covered area was 75% of the total area. As shown in figure 1 the average working times of a node increases as satisfaction value increases. This is because to satisfy a higher satisfaction, select more nodes or each node will work for more times are needed. Also, this figure shows that the traditional random method TR achieves high average working times compared to $\alpha\beta$ *-RC*, $\alpha\beta$ *-USC* and $\alpha\beta$ *-SC* methods. This is because TR does not consider α -cov and β -sat as three methods do. This indicates that the proposed three algorithms are more efficient than TR.
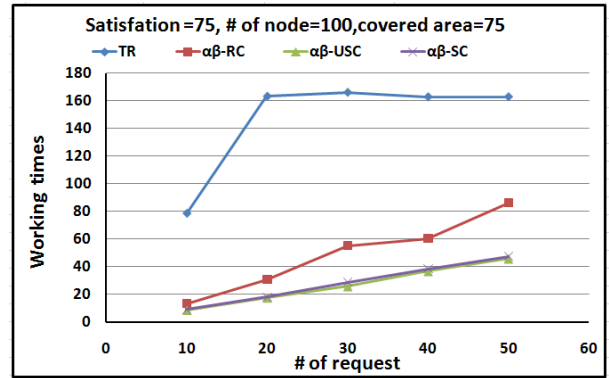

Fig. 2. the relation between number of requests and average working times

Figure 2 shows the average working time against the number of requests where the number of nodes was 100 and the required satisfaction was 75% and required covered area was 75% of the total area. As shown in figure 2 the average working times of a node increases as number of requests increases. This is because to satisfy a higher number of requests, select more nodes or each node will work for more times are needed. Also, the average working times of three methods are much lower than TR. Hence, the proposed three algorithms are better than TR.
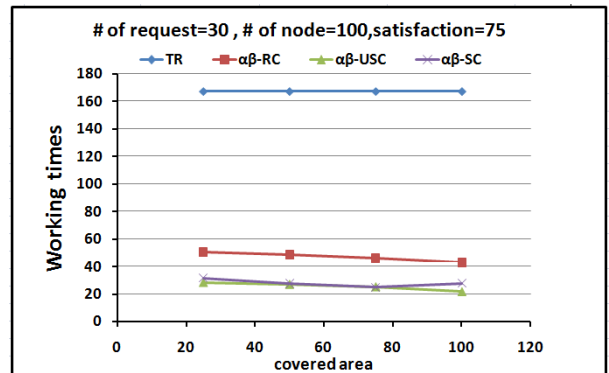

Fig. 3. the relation between required covered area and average working times

Figure 3 shows the time working against the required covered area where the number of requests was 30 and the required satisfaction was 75% and number of node was 100. As shown in figure 3 the average working times of a node decreases as required covered area increases. This is because when the covered area increases more nodes will be selected to cover the required area and this leads to decrease the working times of each node. Also, this figure shows that TR has higher working times compared to $\alpha\beta$ *-RC*, $\alpha\beta$ *-USC* and $\alpha\beta$ *-SC* methods. Hence, the proposed three algorithms are better than TR.
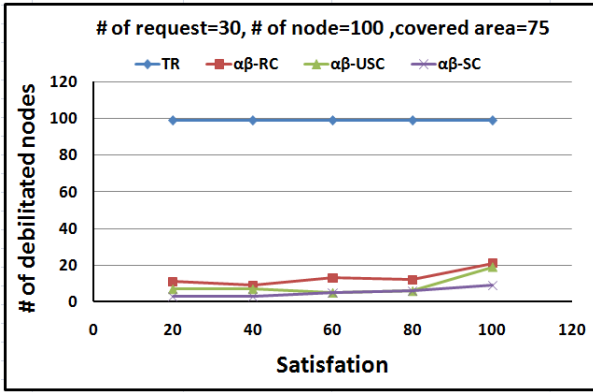
Fig. 4. the relation between required satisfaction and number of debilitated nodes

5.2 Total number of debilitated nodes

Figure 4 shows the number of debilitated nodes against the required satisfaction where the number of requests was 30 and the number of nodes was 100 and required covered area was 75% of the total area. As shown in figure 4 the number of debilitated nodes increases as satisfaction value increases. This is because to satisfy a higher satisfaction, select more nodes or each node will work for more times are needed which leads to the death of nodes. Also, this figure shows that TR has larger number of debilitated nodes compared to $\alpha\beta$ -RC, $\alpha\beta$ -USC and $\alpha\beta$ -SC methods. This is because TR does not consider α -cov and β -sat as to $\alpha\beta$ -R C, $\alpha\beta$ -USC and $\alpha\beta$ -SC do. This indicates that the Proposed three Algorithms are more efficient than TR.
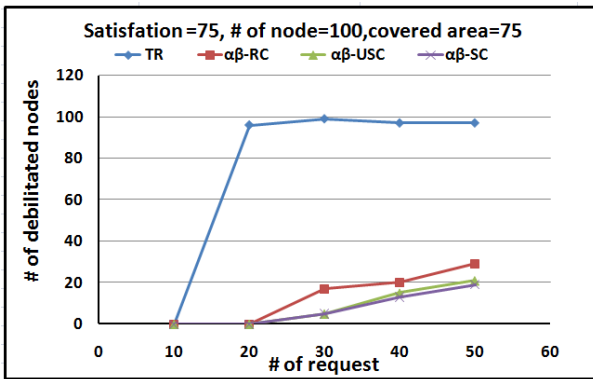

Fig. 5. the relation between number of requests and number of debilitated nodes

Figure 5 shows the number of debilitated nodes against the number of requests where that the number of nodes was 100 and the required satisfaction was 75% and required covered area was 75% of the total area. As shown in figure 5 the number of debilitated nodes increases as number of requests increases. This is because to satisfy a higher number of requests, select more nodes or each node will work for more times are needed which leads to the death of nodes. Also, this figure shows that TR has larger number of debilitated nodes compared to $\alpha\beta$ -RC, $\alpha\beta$ -USC and $\alpha\beta$ -SC methods. Hence, the proposed three algorithms are better than TR.
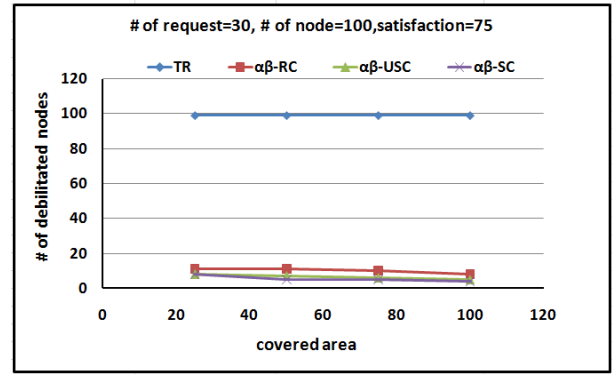

Fig. 6. the relation between required covered area and number of debilitated nodes

Figure 6 shows the number of debilitated nodes against required covered area where the number of requests was 30 and the required satisfaction was 75% and number of nodes was 100. As shown in figure 6 the number of debilitated nodes decreases as required covered area increases. This is because when the covered area increases more nodes will be selected to cover the required area and this leads to decrease the working times of each node which will decrease the number of debilitated nodes. Also, this figure shows that TR has higher number of debilitated nodes compared to $\alpha\beta$ -RC, $\alpha\beta$ -USC and $\alpha\beta$ -SC methods. Hence, the proposed three algorithms are better than TR.
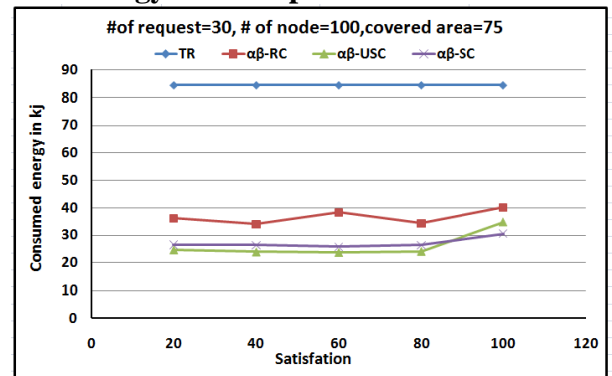
5.3 Energy Consumption

Fig. 7. the relation between required satisfaction and consumed energy

Figure 7 shows the energy consumption against the required satisfaction where the number of requests was 30 and the number of nodes was 100 and required covered area was 75% of the total area. As shown in figure 7, the energy consumption increases as satisfaction value increases. This is because to satisfy a higher satisfaction, select more nodes or each node will work for more times are needed which leads to consume more energy. Also, this figure shows that the traditional random method TR [2] consumes more energy compared to $\alpha\beta$ -RC, $\alpha\beta$ -USC and $\alpha\beta$ -SC methods. This is because TR does not consider α -cov and β -sat $\alpha\beta$ -RC, $\alpha\beta$ -USC and $\alpha\beta$ -SC do. This indicates that the proposed three algorithms are more efficient than TR.
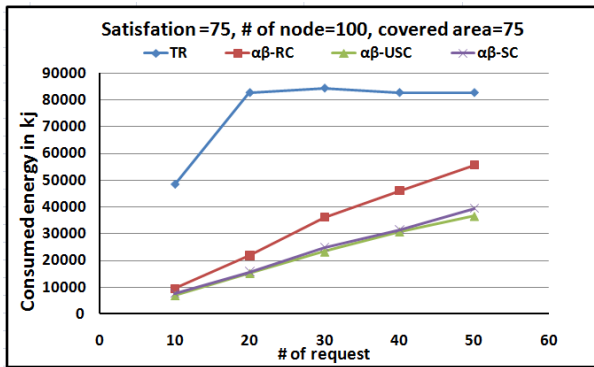
Fig. 8. the relation between number of requests and consumed energy

Figure 8 shows the energy consumption against the number of request where the number of nodes was 100 and the required satisfaction was 75% and required covered area was 75% of the total area. As shown in figure8, the energy consumption increases as number of requests increases. This is because to satisfy a higher number of requests, select more nodes or each node will work for more times are needed which leads to consume more energy .Also, this figure shows that TR consumes more energy compared to $\alpha\beta$ *-RC*, $\alpha\beta$ *-USC* and $\alpha\beta$ *-SC* methods. Hence, the proposed three algorithms are better than TR.
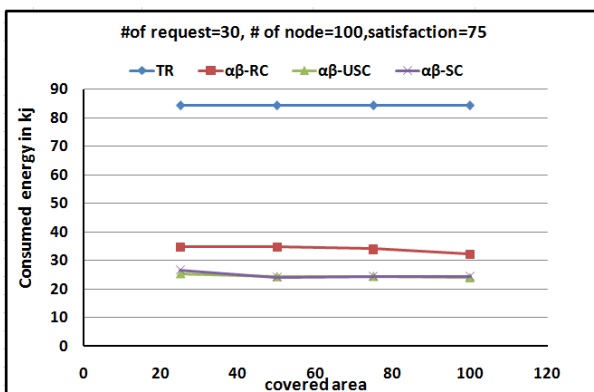

Fig. 9. the relation between required covered area and consumed energy

Figure 9 shows the energy consumption against required covered area where the number of requests was 30 and the required satisfaction was 75% and number of nodes was 100. As shown in figure 9, the energy consumption decreases as required covered area increases. This is because when the covered area increases more nodes will be selected to cover the required area and this leads to decrease the working times of each node which will minimize the consumed energy of each node. In other words, when the required covered area decreases, fewer nodes located inside this area will be selected. Also, this figure shows that TR has higher energy consumption compared to $\alpha\beta$ *-RC*, $\alpha\beta$ *-USC* and $\alpha\beta$ *-SC* methods. Hence, the proposed three algorithms are better than TR [2].

6. CONCLUSIONS

In this paper, the (α *-cov,* β *-*sat) data processing **in** wireless sensor networks was addressedto extend the life time and minimize the total energy consumption of WSNs. To solve

this problem, data processing based on cluster based algorithm and mobile agent called ESSDP was proposed. ESSDP selects processing nodes by using three different selection methods: random selection, unsorting satisfaction selection, and sorting satisfaction selection. All of these methods can reduce the consumed energy, the number of debilitated nodes and the average working times of a node in WSN compared to TR. Hence, ESSDP can extend the life time of WSNs in efficient way. In the future work, the proposed methods will be applied with different clustering algorithms. Also, the presence of the mobile sensor nodes in the network will be studied.

7. ACKNOWLEDGMENTS

8. REFERENCES

[1] Wooldridge, Michael, and Nicholas R. Jennings. "Intelligent agents: Theory and practice." Knowledge engineering review 10.2 (1995): 115-152.

[2] J Fan H Huang S Zhang, Y Sun. Cooperative data processing algorithm based on mobile agent in wireless sensor networks. International Journal of Distributed Sensor Networks, Vol.2012, 2012.

[3] M. Chen, S. Gonzalez, and V. C. M. Leung, Applications and design issues for mobile agents in wireless sensor networks, IEEEm Wireless Communications, vol. 14, no. 6, pp. 20–26, 2007.

[4] Y. C. Tseng, S. P. Kuo, H. W. Lee, and C. F. Huang, Location tracking in a wireless sensor network by mobile agents and its data fusion strategies,Computer Journal, vol. 47, no. 4, pp. 448–460, 2004

[5] H. Qi and F. Wang, Optimal itinerary analysis for mobile agents in ad hoc wireless sensor networks, in Proceedings of the IEEE International Conference on Communications (ICC '01), Helsinki, Finland, June 2001.

[6] Dilip Kumar and RB Patel. Multi-hop data communication algorithm for clustered wireless sensor networks. International Journal of Distributed Sensor Networks, Vol.2011, 2011.

[7] G. Sharma and R. Mazumdar, Hybrid sensor networks: a small world, in Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC '05), pp. 366–377, usa, May 2005.

[8] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, An application-specific protocol architecture for wireless microsensor networks, IEEE Transactions on Wireles Communications, vol. 1, no. 4, pp. 660–670, 2002.

[9] G. Xin, W. H. Yang, and B. DeGang, EEHCA: an energy efficient hierarchical clustering algorithm for wireless sensor networks, Information Technology Journal, vol. 7, no. 2, pp. 245–252, 2008.

[10] O. Younis and S. Fahmy, HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks, IEEE Transactions on Mobile Computing, vol. 3, no. 4, pp. 366– 379, 2004.

[11] F. Xianging and S. Yulin, Improvement on LEACH protocol of wireless sensor networks, in Proceeding of International Conference on Sensor Technologies and Applications, pp. 260– 264, 2007.

[12] A. A. Abbasi and M. Younis, A survey on clustering algorithms forwireless sensor networks, Computer Communications, vol. 30, no. 14-15, pp. 28262841, 2007.

[13] G. Anastasi, M. Conti, M. D. Francesco, and A. Passarella, Energy conservation in wireless sensor networks: a survey, Ad Hoc Networks, vol. 7, no. 3, pp. 537568, 2009.

[14] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, A survey on sensor networks, IEEE Communications Magazine, vol. 40, no. 8, pp. 102105, 2002.

[15] H. Qi, Y. Xu, and X.Wang, Mobile-agent-based collaborative signal and information processing in sensor networks, Proceedings of the IEEE, vol. 91, no. 8, pp. 11721183, 2003.

[16] J. J. Chang, P. C. Hsiu, and T. W. Kuo, Search-oriented deployment strategies for wireless sensor networks, in Proceedings of the 10th IEEE International Symposium on Object and Component-Oriented Real- Time Distributed Computing (ISORC 07), pp. 164171, Santorini Island, Greece,May 2007.

[17] I. Joe, A path selection algorithm with energy efficiency for wireless sensor networks, in Proceedings of the 5th ACIS International Conference on Software Engineering Research, Management, and Applications (SERA 07), pp. 419423, August 2007.

[18] M. Chen, K. Taekyoung, and C. Yanghee, Data dissemination based on mobile agent in wireless sensor networks, in Proceedings of the IEEE Conference on Local Computer Networks (LCN 05), pp. 527528, Sydney, Australia, November 2005.

[19] Karp, Richard M. Reducibility Among Combinatorial Problems. Complexity of Computer Computations, page 85-103. Plenum Press, New York, 1972.

[20] A. Varga and Andrs, The OMNeT++ Discrete Event Simulation System, in Proceedings of the European Simulation Multiconference (ESM'2001). Prague, Czech Republic, June 2001.

[21] Ahmed AA Gad-ElRab, Afaf AS Zaghrout, and Doaa M Alhilaly. Mobile agent cluster-based algorithm for data processing in wireless sensor networks. International Journal of Current Engineering and Technology, Vol.4(No.6):PP.65, Dec 2014.