Accepted Manuscript

WSN in cyber physical systems: Enhanced energy management routing approach using software agents

Elhadi M. Shakshuki, Haroon Malik, Tarek Sheltami

PII: S0167-739X(13)00050-2
DOI: http://dx.doi.org/10.1016/j.future.2013.03.001
Reference: FUTURE 2335

To appear in: *Future Generation Computer Systems*

Received date: 26 November 2012
Revised date: 17 February 2013
Accepted date: 17 March 2013

Please cite this article as: E.M. Shakshuki, H. Malik, T. Sheltami, WSN in cyber physical systems: Enhanced energy management routing approach using software agents, *Future Generation Computer Systems* (2013), http://dx.doi.org/10.1016/j.future.2013.03.001

WSN in Cyber Physical Systems: Enhanced Energy Management Routing Approach Using Software Agents

Elhadi M. Shakshuki[a], Haroon Malik[b], Tarek Sheltami[c]

[a]Jodrey *School of Computer Science, Acadia University, Wolfville, NS, Canada,* [a]*King Faisal University, Saudi Arabia,*

[b]*School of computing, Queens University, Kingston, ON, Canada,* [c] *Department of Computer Engineering, King Fahd University of Petroleum & Minerals, Dhahran, Saudi Arabia*

Received 15 October 2012; accepted 30 December 2013

Abstract

Recently, the cyber physical system has emerged as a promising direction to enrich the interactions between physical and virtual worlds. Meanwhile, a lot of research is dedicated to wireless sensor networks as an integral part of cyber physical systems. A wireless sensor networks (WSN) is a wireless network consisting of spatially distributed autonomous devices that use sensors to monitor physical or environmental conditions. These autonomous devices, or nodes, combine with routers and a gateway to create a typical WSN system. Shrinking size and increasing deployment density of wireless sensor nodes implies to smaller equipped battery size. This means emerging wireless sensor nodes must compete for efficient energy utilization to increase the WSN lifetime. The network lifetime is defined as the time duration until the first sensor node in a network fails due to battery depletion. One solution for enhancing the lifetime of WSN is to utilize mobile agents. In this paper, we propose an agent-based approach that performs data processing and data aggregation decisions locally i.e., at nodes rather than bringing data back to a central processor (sink). Our proposed approach increases the network lifetime by generating an optimal routing path for mobile agents to transverse the network. The proposed approach consists of two phases. In first phase, Dijkstra's algorithm is used to generate a complete graph to connect all source nodes in a WSN. In the second phase, a genetic algorithm is used to generate best-approximated route for mobile agents in a radio harsh environment to route the sensory data to the base-station. To demonstrate the feasibility of our approach, a formal analysis and experimental results are presented.

Keywords: Wireless Sensor Network, Software Agent, Cyber Physical Systems, Routing Algorithm

1. Introduction

Recent advances in wireless communications, networking, and embedded system technologies have led to a growing interest in developing Cyber Physical Systems (CPSs) for various purposes. In recent years, the CPS has emerged as a promising technology that can support the human-to-human, human-to-object, and object-to-object interactions in the physical and virtual worlds.

A CPS is the integration of abstract computations and physical processes [30][31][32], where sensors, actuators, and embedded devices are networked to sense, monitor, and control the physical world. A typical CPS application is to connect appliances embedded with sensor nodes (which are responsible for information collection from the physical world as the source of CPS inputs) to some real-time decision making system (which represents the virtual world). Upon receiving the inputs from sensor nodes, the CPS will make a corresponding decision based on the inputs and computational processing to the actuators in the physical world by a sequence of control processes.

These CPS applications can be roughly classified into smart space, healthcare, emergency real-time system, environmental monitoring and control as well as smart transportation. For all the aforementioned applications, Wireless Sensor Network (WSN) technology is an integral component of CPS designs. For example, a healthcare application could acquire vital signs by medical sensors worn by patients or elders. In the case of environmental monitoring and control applications, sensor nodes can be deployed in the outdoor environments to monitor soil moisture, air quality, and so forth. Similarly, Smart transportation is one of the most important CPS applications. Sensor nodes (such as accelerometer and GPS receiver) could be embedded in vehicles to improve the traffic safety and efficiency.

For all the aforementioned applications, Wireless Sensor Network (WSN) technology is an integral component of CPS designs. If WSN technology is not used in the development of CPSs, the real-time decision making system might have difficulty in acquiring available CPS inputs and making timely decisions.

1.1. Wireless Sensor Network (WSN)

Wireless sensor network is a collection of tiny disposable devices called motes [25] with micro sensors embedded in them, a CPU to perform computation, small amount of memory to store program code and data, and a radio transceiver for wireless communication. These motes collaborate and work together to sense, gather and process large volume of data [1] [27]. The base-station is a node in a WSN that has a dedicated power supply and more resources (CPU, memory, storage, etc.) than a mote. The mote that can identify the user requested i.e., attributes of interest is called source node. The source nodes report back the data to the sink either directly or by relaying it via other sensor motes in WSN.

Unlike traditional IP-based networks, WSN operates on batteries. The deployment of WSNs is usually hard to reach in remote areas such as glaciers, battlefields and forests. Therefore, recharging or changing their batteries is difficult and sometimes impossible. Thus, increasing the lifetime of WSN is one of the fundamental concerns to many researchers. The lifetime of a WSN correspond to the operational time of WSN, before it is unable to report the quality attributes of interest to a base-station due to communication holes created in network. The communication holes results from motes with drained batteries. The motes in WSN sense the environment and report the data either directly to a base-station or indirectly by relaying the data through other motes in path of base-station as in multi-hop communication.

1.2. Energy Waste in WSN

There are several causes of energy waste in WSNs. Firstly, in WSNs there is high density of mote deployment that needs to compete for same channel. Channel contention consumes some amount of battery power. Secondly, sensor motes are mostly deployed in ad hoc fashion rather than with careful pre-planning; thus, they self-organizing to form a communication network to draw quality data. There is sufficient amount of energy spent in maintaining the self-organizing network. Thirdly, the primary mode of communication in WSN is broadcast, which in some cases, results into fair amount of collision. Retransmitting a collided packet adds to the mote's energy consumption. Finally, due to the massive deployment of motes in WSN, a lot of redundant sensor traffic is produced. Routing of such redundant traffic to the base-station consumes a lot of battery power.

Among all the cause of energy losses in WSN, the routing of redundant sensory traffics to base-station accounts for the number one cause of energy consumption in WSN; hence, reduction in their lifetime. This is due to the reason that radio activities i.e., transmission and receiving consumes the most battery power of the mote [2]. Our proposed approach models the routing optimization problem i.e., generating the best approximate route for Mobile Agent (MA) to transverse the WSN, as Euclidean Traveling Salesman Problem (TSP) [25]. Compared to traditional TSP, the scalable and the dynamic topology of sensor networks can result in an incomplete graph. Incomplete graph results when a source node has no direct communication link with other source nodes in a WSN. To allow the source nodes to communicate with each other, generating a complete graph becomes a necessity. To achieve this objective, our proposed approach uses Dijkstra's algorithm. Genetic Algorithm (GA) uses this complete WSN graph to generate best approximate route to send the sensory data to the sink.

In addition, we introduce the concept of mobile agent that can autonomously visit the source nodes and save WSN energy by suppressing the redundant sensory data along the routing path generated by GA. To be specific, the agent is dispatched according to the routing path generated by GA at base-station. Then, the data is aggregated at each source node and is brought back to sink during every data-gathering round. This allows substantial gain towards the network lifetime.

1.3. Contributions and Paper Organization

The main contributions of this paper are: 1) the generation of an optimal routing path in WSN by the use of both Dijkstra and Genetic algorithm, and 2) a successful demonstration of mobile agents for suppressing the redundant sensory data that is the cause of energy waste.

The remainder of this paper is organized as follows: Section II describes the literature review about the related work based on energy-efficient routing in WSNs. Section III discusses our proposed intelligent routing approach. Section IV provides the details of simulation. Section V lists the experimental results of our proposed approach. Finally, Section VI concludes the paper and provides a few directions for the future work.

2. Related Work

There have been several attempts by many researchers to propose several approaches to prolong node's lifetime [2][3][34][36][37] and to eliminate redundant sensory data [4][5][16]. Recently, software agent is used as a combined approach in wireless sensor networks [9][20][21][22][38][39]. Also, there have been attempts to reduce energy consumption based on mobile agents [8][9][23][33]. In wireless sensor network, mobile agent is an intelligent, autonomous program that periodically visits source nodes to collect and aggregate sensory data in a target region.

Due to the bandwidth constraints in WSN, the network's capacity may not satisfy the transmission of sensory data [28]. In order to handle the problem of overwhelming data traffic, Qi, et al. [10] proposed mobile agent-based distributed sensor network (MADSN) for multi-sensor data fusion. Their proposed approach, not only achieves data fusion, but also reduces energy expenditure. However, the application of this approach can only be applied on cluster-based topologies. Mobile agent-based directed diffusion (MADD) in [5] is also introduced an approach to deal with the same problem. The drawback of the approach presented in [7] is that it doesn't always guarantee the best sequence of nodes to be visited by a mobile agent for collecting sensory data from source nodes.

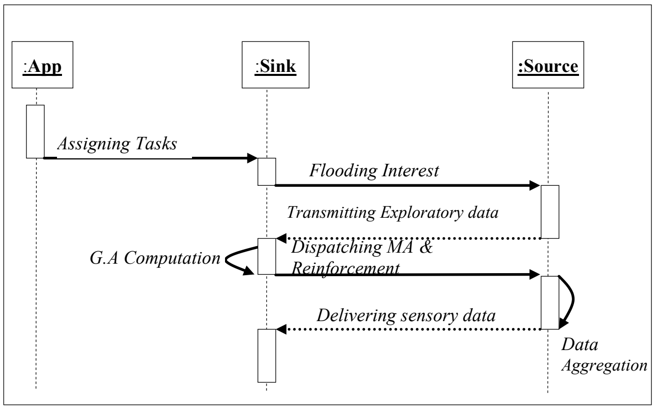


Fig.1. GA-based routing sequence diagram

Currently, most energy-efficient proposed approaches [4][5][18] are focused on data-centric model, such as the directed diffusion [6]. By selecting good path to drain quality data from source nodes, directed diffusion approach can achieve substantial energy gain.

However, it still allows redundant sensory traffic to flow back to sink. The main advantage of MADD is to reduce the redundant sensory data. Through using mobile agent, data is aggregated at each source node and is brought back to sink. This allows additional substantial energy gain toward the network lifetime.

Wu, et al. in [11] proposed agent routing problem (MARP), which is formulated as a combinatorial optimization problem. In this proposed approach, MARP is reduced to a variation of three-dimensional TSP and represented as an NP-complete problem. This work led into a two-level-based GA to compute an approximate solution with two assumptions: (1) wireless sensor network is a cluster-based, and (2) source nodes are within one-hop away from cluster-head.

To deal with the aforementioned limitations, we proposed a mobile agent approach for efficient routing approach by using both Dijkstra's algorithm and genetic algorithms. It should be noted that the order of source nodes to be visited by the mobile agent greatly affects the energy consumption. Although MADD approach presented in [5] allows the agent to autonomously select visit sequence of source nodes for achieving data aggregation, it does not always provide an optimal sequence. To address this shortcoming, our proposed approach introduces genetic algorithm to produce an optimal route.

Furthermore, many optimizations related routing problems can be modelled by a traveling salesman problem (TSP), including mobile agent routing. Compared with traditional TSP, the scalable and the dynamic topology of sensor networks may result in an incomplete graph. If there is an incomplete graph, it will turn out that the source node selected by the sink has no direct communication link with other selected source nodes. In order to allow the selected source nodes to communicate directly, generating a complete graph becomes a necessity. Therefore, our proposed approach utilizes Dijkstra's algorithm to achieve such conversion.

3. Proposed Routing Approach

In this section, we first provide an overview of our proposed GA assisted routing approach. Then, we explain the packet structure of the mobile agent required for our approach. Next, we explain in detail the scheduling technique required by our GA assisted routing approach to generate schedule for wireless sensor network. Lastly, we explain in detail the use of GA by our approach to generate best approximate route in a radio harsh environment for WSN.

3.1. Overview of Routing Approach

Our proposed approach is built upon the intuition of our previous software assisted routing research work to reduce the energy consumption of WSN [26]. In our proposed GA assisted routing approach, we show how to further reduce the energy consumption of WSN successfully by providing MA an optimal path to route the data from source nodes to sink.

The use of GA enables the generation of such optimal routing path. We provide an overview of our proposed routing approach by use of sequence diagram shown in Fig.1.

1) *Assigning Tasks:* An application requiring the sensory data, assigns sink the task of collecting it from the WSN. For example, a weather station application will require the data related to temperature and humidity, an environmental monitoring application used in agriculture (i.e., vineyard monitoring) may require additional data such as pressure, light (i.e. photonics) and in cases vibration; whereas, habitat monitoring application may require data related to sound and state.

2) *Flooding Interest:* When sink receives a task from an application, it sends a request to each of its neighbor nodes. This request is called interest as it is normally in the form of an SQL query as shown in Fig.2. The exemplary interest query in Fig.2 states that after 5 minutes from now (the time at which query is diffused into WSN), the motes in Area_C should measure the temperature, pressure, photonic, humidity attributes per second for 120 times, and the resulting temperature should be reported within 10 minutes after the last sensor sampling is completed. Delay in WHILE clause is as the QoS constraint, and the SAMPLE clause specifies the temporal information. From delay and sample time, we can easily get the latest report time. The sink normally uses a flat routing protocol for flooding interest (i.e., for dissemination of interest). Our GA assisted routing approach is built upon the directed diffusion framework for the dissemination of interest in WSN.

3) *Transmitting Exploratory Data:* Each WSN node is equipped with a stationary agent (SA). When the neighboring nodes receive an interest, SA adds the interest to the nodes cache, and sets up gradient

SELECT	Temperature, pressure, photonic, humidity
FROM	sensors as s
WHERE	s.location=area_C
WHILE	delay < 10 min
SAMPLE	now + 5 min
INTERVAL	1s
LOOP	120

Fig.2. An exemplary interest query

between the two nodes. A gradient is a direction state created between the sending and receiving nodes, from the receiver back to the sender with a specified data rate for information flow. Initially, this data rate will be low; the intention is to explore the network and set up a path to the nodes. Intuitively, these gradients are called exploratory gradients. If the neighbours are not in the region specified by the interest, they send exploratory events (i.e., an interest for a low rate event notification) to their neighboring nodes, until source nodes are found. When a node in the specified region i.e., source node receives an interest, it schedules its local sensors to start collecting data samples. The data is named by attribute-value pairs and sent it back to base-station via relaying the data through neighboring sensor nodes. MA on each neighboring node uses the gradients to choose the next hop towards sink for transmitting the data.

4) *Dispatching MA:* The exploratory data may reach to the sink from many neighbors. The data contains the id of every node that took part in relaying the information from source nodes to the sink, their residual energy i.e., the remaining battery power and the message latency values [16]. The message latency refers to the delay incurred in receiving the response on an interest from the neighboring nodes from the time a node broadcasts the interest. Once the time mentioned in the interest query expires, the sink which has a dedicated power supply uses:
   a) Genetic algorithm on the data reported back to the sink to construct a complete and connected graph of the WSN and
   b) Dijkstra algorithm to assist MA with the following:
      i. The shortest path to reach to the first source node.
      ii. The shortest path to visit all the source nodes.
      iii. The shortest path to reach from the last source node to the sink.

Mobile agent is a special kind of code that has the ability to process data autonomously [6][10] and have the capability to roam around in networks periodically.

Use of mobile agent helps to reduce the amount of data transmitted by source nodes. Eliminating redundant data is a very important task for establishing efficient routing scheme for WSN. Our proposed approach borrows some of the basic ideas introduced in our previous work [26] to eliminate redundant data in WSN.

For every source node that MA visits, it performs local processing to aggregate the raw data of the current source node. And that of the sensory data collected from previously visited source nodes, thereby eliminating data redundancy.

Let $\rho, (0 < \rho < 1)$, be the reduction ratio by the MA assisted local processing. Let S be the size of raw data at source i and let R_i be the size of reduced data. Then, Mobile agent does local processing using the above formula for reducing redundant data on every source node it visited, based on the source-visiting sequence generated by GA. The size of data S, accumulated by mobile agent can be calculated using the following formula [5]:

$$S_{ma}^{1} = R_1\,, \qquad (1)$$

$$S_{ma}^{2} = R_1 + (1-\rho)\times R_2\,, \qquad (2)$$

$$S_{ma}^{i} = S_{ma}^{i-1} + (1-\rho)\times R_i\,, \qquad (3)$$

$$= R_1 + \sum_{k=2}^{i} (1-\rho)\times R_k\,. \qquad (4)$$

During the mobile agent migration from one node to another, it aggregates the sensory data and removes redundant data at the same time. The aggregated data can be calculated using equation 6:

$$S^{i} = \sum_{k=1}^{i} R_k \qquad (6)$$

Moreover, the energy cost is reduced. In client/server based sensor network, all source nodes in target region individually transmit sensory data back to sink with a specific interval. In our proposed approach, the mobile agent carries both the processing code as well as the source-visiting sequence generated by GA.

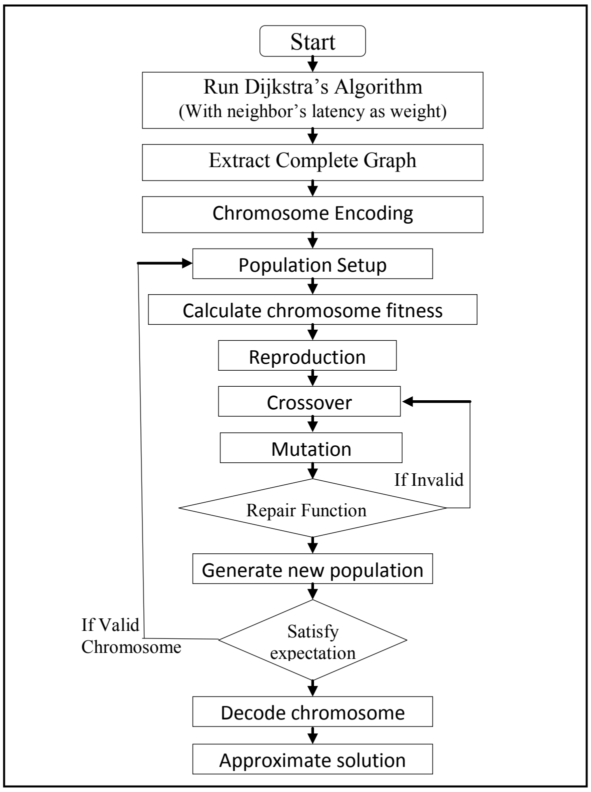


Fig.3. Generating approximate solution operation flow

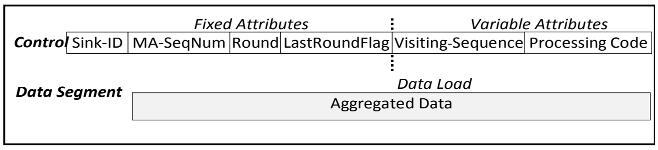


Fig. 4. Packet structure of mobile agent

Once the mobile agent arrives at the first source node, it identifies the second node to be visited and continues visiting other source nodes in sequence until it reaches the last source node.

We refer to this activity of the mobile agent as work cycle. At the end of each cycle, aggregated data is sent back to the sink along with the reinforced path.

Once the MA is back to the sink, the sink recalculates the optimal routing sequence based on the residual energy of the sensor nodes to ensure that nodes along the optimal path are aggressively used and depleted their energy quickly.

The calculation of the optimal path is based on the GA fitness function. After the decision is made on the optimal path i.e. source-visiting sequence, the sink reinforces a path to the last source. At the same time, it dispatches again the mobile agent to the first source node.

The sequence of operations demonstrating the phases and processes of our GA-based routing approach is shown in Fig.3.

3.2. Packet Structure

The mobile agent packet structure is shown in Fig. 4. This packet is composed of both control segment and data segment. The control segment is divided into two parts: fixed and variable attributes. Fixed attributes include: SinkID, MA-SeqNum, Round and LastRoundFlag. The pair of SinkID and MA-SeqNum identifies a mobile agent. When a sink dispatches a mobile agent, it will increase the value of MA-SeqNum, which initially is set to 1. The Round attribute specifies the number of times MA should periodically report data to the sink while LastRoundFlag indicates that there is no more data to report back to sink after the current round of the given task.

When a mobile agent with LastRoundFlag field travels to a source node, it will indicate the source node to unload the executable instructions and code required for the corresponding sensing task after the completion of the current round. The variable attributes are Visiting-Sequence and Processing code. The former describes a source-visiting sequence constructed by GA, which a mobile agent must follow within a cycle. The latter is a special program that enables any source node to perform local operations on the sensory data i.e. aggregation. The MA distributes a copy of this special program at every source node on its first visit. A data segment of the agent contains the accumulated data. When the mobile agent arrives at the first source node, the size of aggregated data is equal to zero. The data segment is incremented whenever the agent migrates from one source node to another.

3.3. GA Assisted Routing

When using GA to address mobile agent routing, the sink requires a proactive routing mechanism i.e. sink needs to know the complete network topology. It is difficult to keep up-to-date topology information in such highly dynamic and distributed environments [12]. These updated packets consume a large portion of the network bandwidth and more energy. In our proposed approach, we use Data Diffusion (DD) as energy efficient routing strategy. According to DD, the sink is only interested in collective information area of sensor network (i.e., interest region). Therefore, a sink needs to only maintain the topology of an interest region. As a node in interest region (we call it a target node) receives an interest packet from its neighbor, it stores the original interest message and to maintain the latency of this message. Flooding mechanism ensures each node in the interest region to cache its neighbors' latency. A target node sends its neighbor latency in the exploratory data.

Based on the exploratory data delivered by all the target nodes, the sink constructs a topology of target area and employs GA to compute an approximate path. In order to abstract sensor networks as a standard or symmetric TSP, we need to address the following issues:

1. TSP assumes each vertex in a given complete weighted graph, G=(V,E) containing $n\times(n-1)/2$ edges, is directly reachable. Thus, the question is how to convert an indirectly accessible sensor network into a complete weighted graph?
2. A sensor network is dynamic in open environment. In such environments, new sensor nodes may join existing nodes or may disjoin due to energy depletion. Therefore, the question is how to minimize GA computation interval and guarantee optimizations process not to converge quickly.
3. GA requires an appropriate fitness function to evaluate chromosomes and achieve crossover/mutation operation.

In the phase of receiving exploratory data, a sink recognizes a whole topology in target. This topology cannot guarantee the direct communication between source nodes. However, any source node is logically directly reachable because a shortest path from one source to another exists. Therefore, before transforming visiting-sequence issue to a TSP, we utilize Dijkstra's algorithm to generate the shortest path among a group of source sensors.

In Dijkstra's algorithm for finding shortest paths, we assume that minimum latency between the two nodes represent the shortest path. We also give a weight to each edge following the formula min(lij, lji), because the latency from sensor i to j is different from the latency from sensor j to i. Thus, a sink generates a dynamic symmetric metric $D(t)$ as follows:

$$D(t)=\begin{pmatrix} l_{11}(t) & l_{12}(t) & \ldots\ldots & l_{1n}(t) \\ l_{21}(t) & l_{22}(t) & \ldots\ldots & l_{2n}(t) \\ \ldots & \ldots & \ldots\ldots & \ldots \\ l_{n1}(t) & l_{n2}(t) & \ldots\ldots & l_{nn}(t) \end{pmatrix}_{n(t)\times n(t)} \quad ; \quad l_{ij}(t)=l_{ji}(t).$$

Where, l_{ij} represents the latency from the source sensor i to j and t represents the current time when the metric is generated by the base-station. In a distributed communication system, nodes may be added or removed from the system at any time. Suppose we have a finite time sample space Ω containing m distinct time samples. That is:

$$\Omega=\{t0, t1, t2, \ldots, tm\}.$$

Where, $t_0 = 0_s$, $t_m = T_s$ and $t_k(0\leq k\leq m)$ represents a specific time sample in this set. Therefore, the symmetric metric at a specific time is described as follows:

$$D(t_k)=\begin{pmatrix} l_{11}(t_k) & l_{12}(t_k) & \ldots\ldots & l_{1n}(t_k) \\ l_{21}(t_k) & l_{22}(t_k) & \ldots\ldots & l_{2n}(t_k) \\ \ldots & \ldots & \ldots\ldots & \ldots \\ l_{n1}(t_k) & l_{n2}(t_k) & \ldots\ldots & l_{nn}(t_k) \end{pmatrix}_{n(t_k)\times n(t_k)} \quad ; \quad l_{ij}(t_k)=l_{ji}(t_k).$$

At this instant, the MA routing is changed to a dynamic environment optimization, i.e. finding an approximate best solution in an interval defined by Δt. Where, $\Delta t = t_{k+1} - t_1$. The solution quality is depending upon the interval, i.e. Δt_α (solutionquality)-1. However, sensor network cannot be idle for a long time. Thus, sink has to consider the tradeoff between time and routing quality (minimizing energy expenditure) in order to achieve the best solution for our routing approach.

3.4. Scheduler

Our proposed approach is designed for dynamic environment, which means lots of sensors may die by running out of their power, or many new sensors may join the WSN after several data gathering rounds. It turns out that the topology for the entire WSN or specific interest region is changed. In our approach, genetic algorithm is used for creating a globally optimal routing path, which is based on the shortest path generated by Dijkstra's algorithm as an initial input. Nevertheless, Dijkstra's algorithm utilized the topology of interest region of WSN to produce the corresponding shortest path. Therefore, changes for the topology of WSN will greatly influence the accuracy of the result of GA.

According to the aforementioned reason, a scheduler is used to maintain relatively high accurate results of GA. The scheduler generates schedules in the sink, where the topology for the interest region is available. This scheduler is responsible for two main tasks, namely: schedule generation and dissemination.

3.4.1. Schedule generation

An online scheduling technique is used to generate schedule for wireless sensor network. The online scheduler generates a data aggregation tree based on the current energy level of sensor nodes, which is the shortest path produced by Dijkstra's algorithm according to the topology of interest region of WSN. In fact, this shortest path is the source-visiting sequence that will be used as an initial input of GA. The schedule is defined to be data aggregation tree along with its frequency that is defined as the number of data gathering rounds. For example, if an aggregation tree T_i is used for n number of rounds, then at the n^{th} round, all nodes send their energy information along with their data packets to sink. Once the scheduler receives energy information from all nodes, it will update the topology of interest region at sink. The new aggregation tree T_{i+1} is computed based on the current energy information received from the nodes.

Frequency (Number of round) 1 byte	Node ID 1 byte	Parent ID 1 byte	Child list (n-2) byte

Fig. 5. Scheduler structure

3.4.2. Schedule Dissemination

After a schedule is computed, the sink broadcasts the schedule information in a single packet to the network. This packet contains the routing information for all nodes in the data aggregation tree. We named this single schedule packet as *SchedulerPacket*. The content of the scheduler packet is shown in Fig.5.

The first three segments are of one byte each. The first segment contains the number of rounds, whereas the second and third segments represent the node's identification number and the node parent's ID respectively. The fourth segment is the maximum size that the child list can have with n-2 bytes in size, with n sensor nodes in the interest region.

3.5. Genetic Algorithm. With the increment of sensor devices in sensor network, conventional search methods (calculus-based, enumerative and random method) cannot meet our required robustness. As proposed in [13], GA is different from normal optimization and search procedures in four ways:

1. GA works with a coding of the parameter set, not with the parameters themselves.
2. GA searches from a population of points, not from a single point.
3. GA uses payoff (objective function) information, not derivatives or other auxiliary knowledge.

4. GA uses probabilistic transition rule, not deterministic rules.

In a word, GA is widely used in applications where the accurate results are not very important and search space is massive. The main advantage of GA is that the process is completely automatic and avoids local minima. Therefore, we utilized genetic algorithm to generate an approximate best solution rather than using the conventional search methods.

In a broad sense, a genetic algorithm performs fitness tests on new structures to select the best population [19]. Fitness is used to determine the quality of the individual based on the defined criteria. In our proposed approach, the WSN is implemented as a population of chromosomes, where candidate solutions (population) will be evaluated to obtain better solutions. GA is performed on the complete graph generated by Dijkstra's algorithm to produce best approximated route for an energy efficient data aggregation trees. Before adopting GA to compute an approximate best solution, we have to consider encoding method, selection approach, and fitness. The main components and core concepts of GA are as follows:

3.5.1. Chromosome

A chromosome is a collection of genes and represents a data aggregation tree for a given network. Each chromosome has the size of fixed length that is determined by the number of nodes in a given network. In other words, chromosome represents a solution in GA. Two main components constitute a chromosome are gene index and gene value. A gene index represents the node's identification number (ID). As for gene value, it indicates the source node's parent ID. In sensor network application, we choose permutation method to encode a chromosome, because mobile agent routing is an ordering problem. In permutation encoding, every chromosome is a string of numbers, described as follows:

Gene index	2	4	3	5	7	9	…	199
Gene value	30	45	25	30	80	16	…	99

These numbers represent the corresponding IDs for each sensor node. The whole chromosome represents a visiting sequence. To be specific, chromosome represents a possible solution of GA for defined problem. We use population to represent the complete solution to GA, where population is the collection of chromosomes.

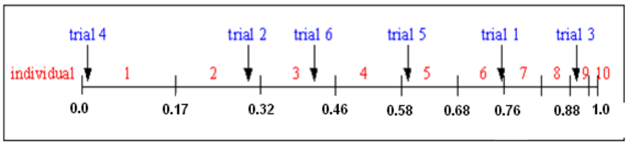
3.5.2. Selection

The process of selection decides which chromosomes in current population will mate (crossover) to produce a new chromosome. Then, these generated chromosomes will join the existing population. Those combined population will also be the basis for the next selection. A chromosome with a higher fitness value has a better chance of being selected. Several methods are used to select chromosomes to crossover, such as: *Rank* selection, *Tournament* selection and *Roulette-Wheel* selection. According to Darwinian survival of the fittest principle, Roulette-Wheel selection, also called stochastic sampling with replacement. In our proposed approach, the individuals (chromosomes) are mapped to contiguous segments of a line, such that each individual's segment is equal in size to its fitness. A random number is generated and the individual whose segment spans the random number is selected [15].

Table 1. Selection of Individuals

Individual index	1	2	3	4	5	6	7	8	9	10
Fitness value	1.9	1.7	1.5	1.3	1.1	0.9	0.8	0.7	0.3	0.1
Probability of selection	0.17	0.15	0.14	0.12	0.10	0.08	0.06	0.05	0.02	0.01

Table 1 shows the probability of selection for 10 individuals; linear ranking together with the fitness value. For example, individual 1 is the best-fit individual and occupies the largest interval. Individual 10 is the least-fit individual and has the smallest interval on the line. For selecting the mating (crossover) population, the appropriate number of uniformly random numbers (uniform distributed between 0.0 and 1.0) is independently generated sample of 6 random numbers; for example: 0.75, 0.30, 0.90, 0.02, 0.61 and 0.41 for trails 1 to 6. Then, the process of Roulette-Wheel selection can be demonstrated in the following figure.



The process is repeated until the desired number of individuals is obtained, which is called mating population for crossover operation. This technique is analogous to a *Roulette-Wheel* with each slice proportional in size to the fitness, as shown in Table 1.

After this selection of mating population, the individuals (chromosomes) will consist of 1, 2, 3, 5, 6, and 9. These generated chromosomes will join the existing population and will be the basis for the next selection.

3.5.3. Fitness function

In GA, we used a defined function (called fitness function) to solve a given problem by evaluating the fitness of a chromosome, where a chromosome with a higher value has the better chance of survival. GA solves the design problems similar to that of natural solutions for biological design problems [19]. In our proposed approach, the chromosome with a higher fitness function value will have a better chance to be selected for selection operation.

That chromosome will also have a better chance to be selected as the best solution.

In mobile agent routing problem, the objective is stated as the minimization of cost function rather than the maximization of some profit function. For a routing search work, we have to transform a minimization problem to a maximization problem. Therefore, the following cost-to-fitness transformation $f(x)$ is used.

$$f(x) = C_{max} - latency(x), \text{ where } latency(x) \le C_{max}$$

Where, C_{max} is taken as the largest *latency(x)* value in the current population and *latency(x)* is a sub-function which represents the amount of time it takes a mobile agent to visit all source sensors. The fitness parameter in the above fitness function is denoted as latency. The initial fitness parameters can be assigned arbitrary to the sensor nodes with the interest region as *latency* (x_{ij}). Then, after every generation the best-fit chromosome is evaluated and the fitness parameters are updated as follows:

$$\Delta f = f(x_{i+1}) - f(x_i).$$

The Δf expression represents the change in the fitness parameter value and the index i represents the number of generation. Therefore, the fitness function can be described as the subtraction of maximum latency of current population and the previous population.

Based on the encoding/selection approach and the fitness function given above, we could select a group of chromosomes from a given population according to the evaluated chromosome's fitness *f(x)*. For extending the search space and computing an approximate solution by producing the new generation, GA needs to accomplish crossover and mutation.

3.5.4. Crossover Operator

Crossover is a well-known genetic operator that acts on a pair of chromosomes. It recombines genetic material of two participating chromosomes. The result of crossover operation depends on the selection process made from the population.

In 1989, Goldberg [13] proposed a crossover method called partially matched crossover (PMX) to tackle a blind salesman problem with permutation representation. Under PMX, two chromosomes (permutation and their associate alleles) are aligned and two crossover sites are picked uniformly at random along the chromosomes.

These two points define a matching section that is used to affect a cross through position-by-position exchange operations. In other words, for crossover operation we randomly select crossover point and the gene values of participating parents are flipped to produce a pair of child chromosomes. Consider the following chromosomes (red color represents a crossing site):

Parent chromosome: (gene value)									
A:									
12	30	45	26	70	16	10	99	36	18
B:									
30	16	10	45	18	99	36	12	70	26

PMX proceeds by position-wise exchanges. First, mapping chromosome B to chromosome A: the gene value 45 and 26, 18 and 70, 99 and 16 exchange places. Similarly, mapping chromosome A to chromosome B, the gene value 26 and 45, value 70 and 18, and value 16 and 99 exchange places. Following PMX, two new offspring are generated: Where, each chromosome contains ordering information partially determined by its parent.

Offspring chromosome: (gene index)									
A*:									
12	30	26	45	18	99	10	16	36	70
B*:									
30	99	10	26	70	16	36	12	18	45

3.5.5. Mutation Operator

In order to avoid losing a feasible gene segment and maintain various solution populations, a sink has to implement mutation operation. Although mutation introduces a new sequence of genes into a chromosome, there is no guarantee that mutation will produce desirable features in the new chromosome. In addition, there exists the difference from the conventional mutation operators (like bit inversion); mutation for sensor network must be modified because a chromosome is encoded as a permutation method.

In our application, we adopt a two-point mutation operator, which randomly selects two sites and exchange the genes with each other in the chromosome. The following example describes this operation.

Original chromosome:									
12	30	45	**26**	70	16	**10**	99	36	18
Mutated chromosome:									
12	30	45	**10**	70	16	**26**	99	36	18

The bold and underlined node IDs represents the gene exchange locus. Usually, in the process of implementation, we have to consider parameters like crossover rate and mutation rate. Crossover rate generally should be high probability, about 80%~95% to guide the evolution direction. Unlike crossover, the mutation rate is very low and normally is around 0.5% ~ 3%, since the increment of the probability may destroy some superior gene segments. While performing our experiments, a slight variation of probability values does not have a significant impact on the quality of new population and still can keep a steady optimization.

3.5.6. Repair Function

Although crossover and mutation operation manipulate between two valid chromosomes, they may produce invalid chromosome(s), which would contain(s) cycles (loops). Since chromosome represents a data aggregation tree for a given network, it also represents the source-visiting sequence, there should not be cycle or loop in it. Hence, repair function is used to identify and remove the inclusion of invalid chromosomes in new generation. A pseudo-code for the repair function is shown in Fig.6, where *i* represent the gene index.

```
Algorithm RepairFunction
  for each gene_i in chromosome do
     While gene_i creates a loop do
        Randomly select a new parent for gene_i
     end while
  end for
```

Fig.6. Algorithm for repair function

In [24], a large number of experiments show that adding repair function to GA will generate much better solutions than the one without repair function. Furthermore, on terms of the efficiency, GA with repair function also generates solutions significantly faster than the one without repair function.

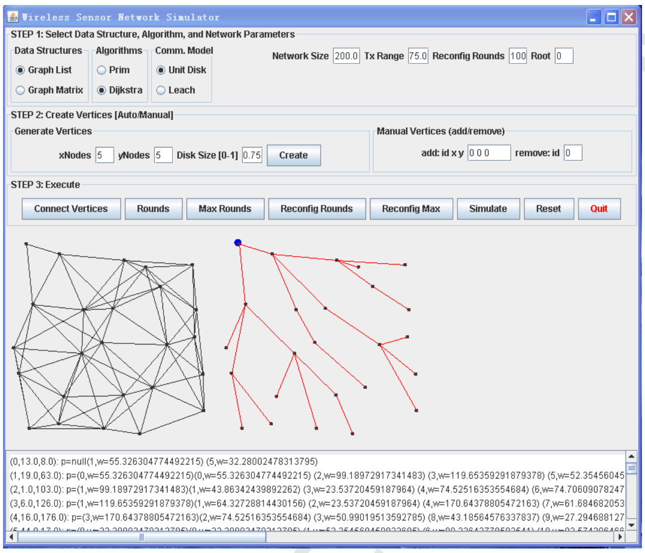


Fig.7. Simulator interface

Finally, a sink has to choose the first source sensor because GA generates a Hamilton Cycle [29]. The size of mobile agent becomes the maximum in the last source due to carrying aggregated data, while it is the minimum in last one. Thus, to minimize energy consumption, the first source sensor should be the furthest target sensor from the sink. Based on the choice of the first source node, the sink reinforces the last source node appeared in the Hamilton Cycle.

4. Simulation

The main objective of our simulation is to evaluate our proposed approach and to compare it with directed diffusion, a well-known routing approach for WSN. We detail our communication model and our custom simulator in subsequent sections.

4.1 Communication model

The communication model utilized in this paper is similar to that presented in [40]. The model employees Embedded Sensor Board (ESB), that is a prototype wireless sensor network device developed at Freie University Berlin [41].The ESB consists of a Texas Instruments MSP430 low-powered microcontroller with 2k RAM and 60k flash ROM, a TR1001 radio transceiver, a 32k serial EEPROM, an RS232 port, a JTAG port, a beeper, and a number of sensors (passive IR, active IR sender/receiver, vibration/tilt, microphone, temperature). In this model, the energy E_r spent while in a given radio state r to transmit or receive message is calculated using Eq. (7).

$$E_r = P_r \times T_r, \quad \text{.......................................(7)}$$

Where P_r is the consumed power during time T_r in a given radio state r. The P_r is calculated using Eq. (8).

$$P_r = V_r \times I_r, \quad \text{.......................................(8)}$$

Where V_r is the voltage applied and I_r is the current induced at a given radio state r. For a shorter distance transmission, the energy consumed by a transmit amplifier is proportional to d^2 [42], where d is the distance between nodes. The energy consumed ET_{ij} by an agent to carry a message of length l bit from a node i to a node j is given by Eq. (9).

$$ET_{ij} = lE_e + lE_s{}^{d2}{}_{ij} \quad \text{.......................................(9)}$$

Moreover, the energy consumed ER to receive the l bits message is given by Eq. (10).

$$ER = lE_e + lE_{BF} \quad \text{.......................................(10)}$$

Where E_e is the energy consumed in the electronics circuit to transmit or receive the signal, E_s is the energy consumed by the amplifier to transmit at a shorter distance, and EBF is the energy consumed for beam forming.

We used the variables and constants as described in Table 2 for our custom simulator. The network simulator (ns2) [44] implements three radio propagation models: free space, two-ray ground reflection and shadowing [43]. The first two are variations of the unit disc graph model, i.e. within a certain radius of the sender all nodes always have perfect reception. We borrowed *free space* i.e., unit disk model from ns2 and used it as beam propagation model in our custom simulator.

Table 2. Variable and constant for our custom simulator

Parameters	Units
Radio bandwidth	19,200 bits/s
Control packet length	10 bytes
Size of aggregated data	242 bytes
MAC header length	8 bytes
Sensed packet size	50 bytes
Sensed packet interval	1 s
Size of mobile agent	250 bytes
Size of static agent	900 bytes
Energy consumed in the electronic circuit to transmit or receive signal	0.10125mWs
Energy consumed for transmitting 1 data packet	13.5mWs
Basic energy consumption PCL	12.0mA
Additional energy consumed for sending PTX	12.0mA
Additional energy consumed for receiving PRX	4.5mA
Total energy needed in sleep mode PSL	0.008mA

4.2 Custom Simulator

The simulator is implemented using Java language in Eclipse development environment. Several modules are required in the simulation of a wireless sensor network environment, such as network configuration, communication and routing. The communication channel in our simulation is assumed to be ideal i.e., there is no data collision. In addition, we do not simulate the energy required to receive schedule from sink. In our simulation, we implement a random network layout for the WSN, since random deployment of sensor nodes is commonly used in military and security applications, where sensor nodes are dispersed in a random fashion by an airplane or other moving devices in a harsh environment.

We incorporated Unit Disk as our communication model for simulation. For Unit Disk model, the radius of circular communication is fixed at transmission range. We use Euclidean for calculating the distance.

Fig. 7 shows a graphical user interface for the implemented simulator. The network is configured to a network size of 200 sensor nodes with transmission range of 75 meters. The sink is placed as node 0, which is the default index for sink node. The data structure used in our simulation is Graph list. The routing method used is Dijkstra's algorithm. In the interface, the Round button's main function is to start generating the shortest path according to Dijkstra's algorithm, which will be used as an initial input for GA. Reconfig Rounds button is used to simulate the designed scheduler. Our default number of round for the scheduler is set to 100. The Max Round button function is used to simulate different deployment of sink node in the given network. Reconfig Max button is similar to Max Rounds that is used for finding the sink node that gives maximum rounds; however, it uses Reconfig Rounds instead of getRounds() function in Java code. We set the interest region to a size of 5×5 meters. With 1 meter as intervals between each sensor node, the corresponding connected graph is displayed for the interest region as shown in Fig. 7 with black lines. The shortest path to reach to source nodes based on this graph is displayed with colored lined within the area of GraphPanel of the simulator.

In Fig. 7, the text area in the lower part of the interface is reserved for displaying the desired information about the routing in the interest region of the network under test. The routing information consists of the node ID for each sensor nodes in the interest region with their x and y coordinates, and the node ID for which they are connected together with the corresponding weights for those links.

5. Experimental Results

This section discusses our several experimental results that include comparisons between our proposed approach and directed diffusion.

5.1. Experiment-1

The first experiment is performed to validate the effectiveness of the use of genetic algorithm for our proposed approach. From 'effectiveness', we mean if there exist significant difference between the best routing path proposed by GA as compared to its average recommendations per simulations. The GA outcome heavily depends on the fitness function. The fitness function performs cost-to-fitness transformation based on the latency parameter. We plotted the latency of the path based on the value for best fitness function along with the average fitness value for one thousand simulations as shown in Fig. 8.

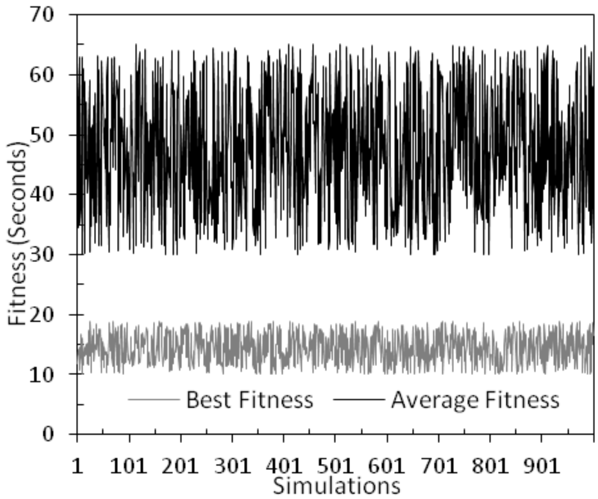


Fig.8. Fitness for various simulations

Each simulation was carried out for 5 minutes. The average fitness value i.e., latency in the graph shows the average of all the chromosomes in the population obtained in the final generation. The graph shows that the best fitness values are significantly better than the average fitness values for all the simulations.

The x-axis shows the number of simulation; whereas, fitness value (latency) is represented along y-axis.

The GA is effective in energy efficient routing for WSN. The best solution proposed by GA is 65% better than G.A's average recommendations.

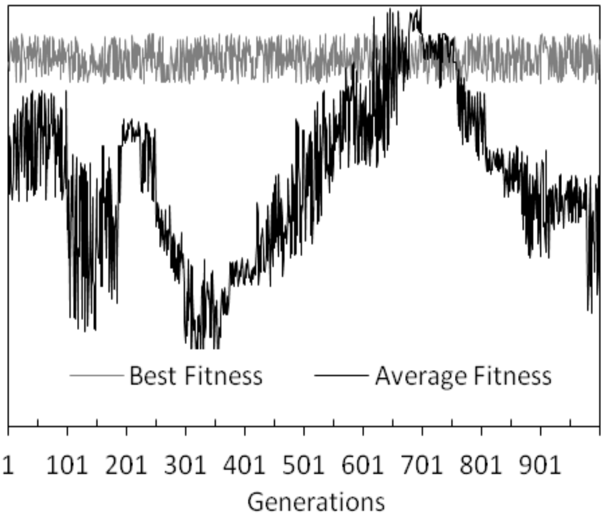


Fig.9. Fitness with respect to generations

5.2. Experiment-2

The second experiment is performed to understand the variation between best and average fitness with respect to the increase in number of generations as show in Fig.9. The Fig. 9 shares the same y-axis as that of the Fig. 8. However, the x-axis represents the generations. The result shows that there is a variation in the values of the best fitness of the chromosomes. Nevertheless, we do see a steep 'V' curve. This is due to the low fitness chromosomes, which were filtered during the selection criteria i.e., only the best fit chromosomes survived for the next generation. Also, mutation played the deception of having G.A stuck to its local maxima i.e., the steep 'V' curve.

The GA out performs DD by consuming 35% less network energy i.e., power consumption.

5.3. Experiment-3

The third experiment is conducted on a network size of 50 nodes, where initial energy parameter for each node was kept constant throughout the simulation. We used random rectangle areas to simulate the increase and decrease of sensing. The nodes in a rectangle area were selected and exposed to heavy load. It was observed that our proposed approach performed 25% better than directed diffusion in terms of energy consumption, as shown in Fig.10.

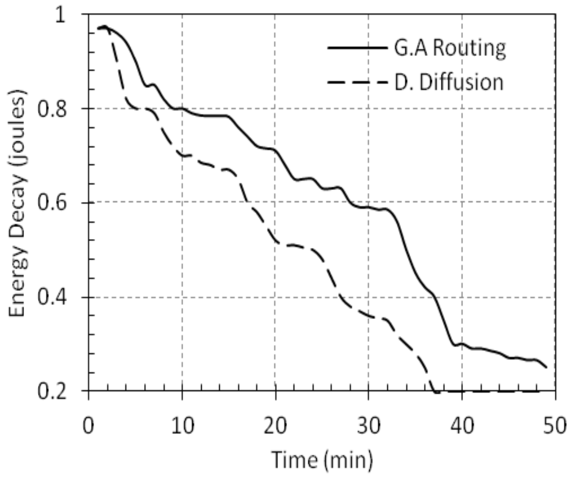


Fig.10. Energy consumption comparison between G.A routing and directed diffusion approach (routing protocol(s))

5.4. Experiment-4

The fourth experiment is conducted on a set of 200 nodes. Sink dispatches the identical interest for both directed diffusion and our proposed approach. The effect of latency was studied by varying the number of nodes for both approaches.

We define latency as the delay incurred since the first interest is dispatched from sink until the sink receives the first data packet from WSN. Fig. 11 clearly identifies that our proposed approach overrides the directed diffusion by 2 folds in terms of latency. We found out that with fewer nodes, up to 50 nodes, there is just one fold difference between both approaches. As the number of node density increases above 50, the latency for directed diffusion doubles in contrast with our proposed approach. This might be due to the fact that increasing the number of nodes in directed diffusion also increased the chance of collision, which adds to latency factor. In our approach, there is only one mobile agent to transverse through the network; hence, reducing the chances of collision to minimal.

The proposed approach overrides the directed diffusion by two folds in terms of latency.

5.5. Experiment-5

The fifth experiment was aimed to compute the amount of data reported for both approaches to sink, given the same interest query. In real world deployment of wireless sensor networks, the field engineers evaluate the performance on two criteria: (1) the ability to collect and deliver data to the base-station or backend PC, and (2) the ability to recover from loss or error, i.e. how much resilient sensor platform is to commonly occurring faults.

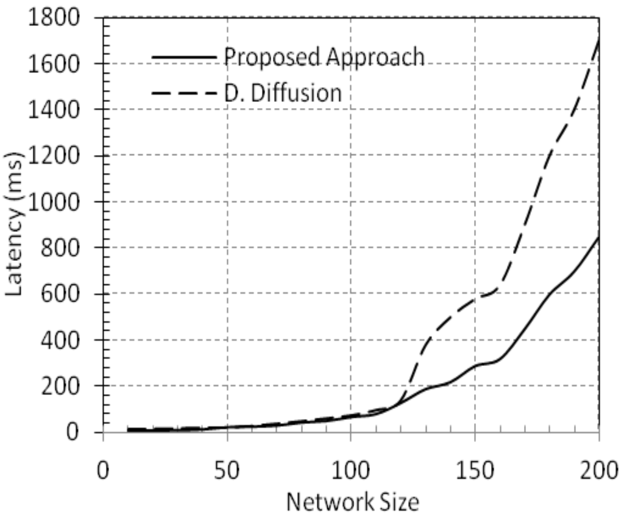


Fig. 11. Energy consumption comparison between G.A routing and directed diffusion approach (routing protocol(s))

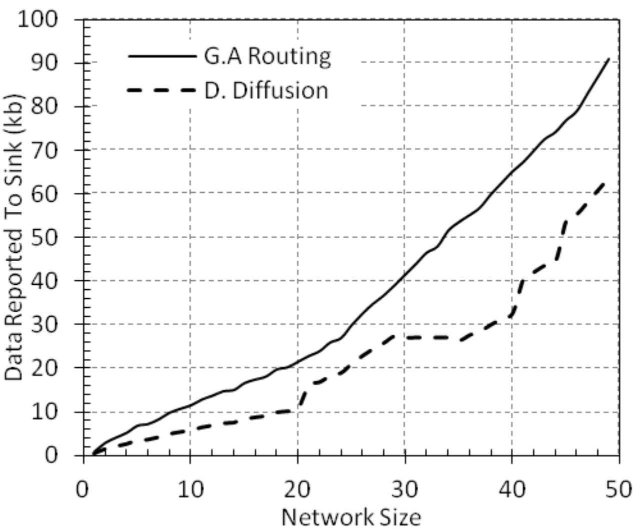


Fig.12. Data redundancy comparison between proposed and directed diffusion approach

On same note, we sought to seek if the choice of routing protocol has any considerable impact on the first criteria, i.e., ability to collect and deliver data to base-station. We did multiple experiments by increasing the size of the WSN (up to 50 nodes). We diffused the same query for both approaches in the WSN for all experiments and observed the amount of data reported to the sink over the period of 3 hours. It is observed from Fig. 12 that directed diffusion routing protocols draw more data from the WSN to sink as compared to proposed approach.

On closer look at the data reported to the sink, we found out that much of the data reported via directed diffusion accounts being redundant. With network size of 50 nodes, half of the data reported to the sink by directed diffusion (in comparison with proposed approach) accounts for being redundant which is important cause of energy decay in sensor network lifetime. We synthesize this claim by

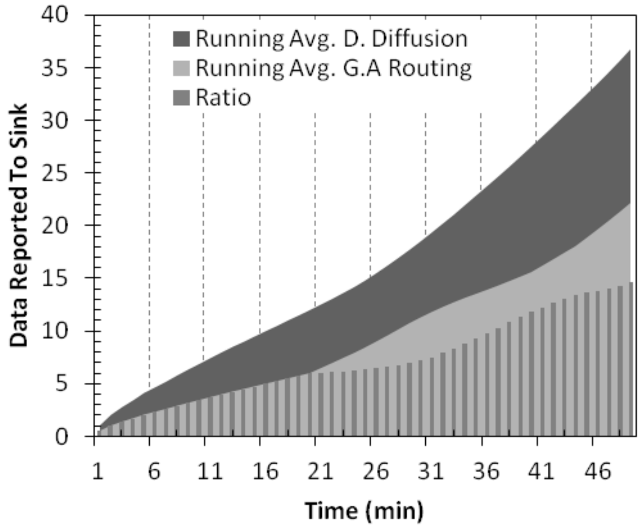


Fig.13. Average data reported to the sink for both approaches.

comparing the cumulative moving average of data reported to sink for 60 minutes and is shown in Fig.13. Cumulative moving average CMA of data reported at time interval t_n is the average of all the date reported to sink up until t_n and is calculated as:

$$\mathrm{CMA}_{t_n} = \mathrm{d}_1 + \mathrm{d}_2 + \cdots + \mathrm{d}_n/\mathrm{n}\ .$$

Where d is the amount of data reported to sink. Where, n refers to the sequence of time intervals, which is in our case is minutes. CMA for data reported to the sink for every new time interval is calculated as:

$$\mathrm{CMA}_{t_{n+1}} = \mathrm{d}_{i+1} + \mathrm{iCA}_i/\mathrm{i} + 1\ .$$

The bar plot in Fig. 13 is the ration between the cumulated average data reported to sink every minute. From this figure, it is evident that our approach banks more energy by suppressing redundant flow of sensory data to base station.

The proposed approach reduces the energy consumption of WSN by suppressing the flow of redundant sensory data to the sink.

5.6. Experiment-6

Our last experiment is conducted to compare the health of the WSN using our proposed approach and DD. For a healthy network, it is important that there is no sudden node decay; rather nodes should expire in a linear manner. Moreover, network traffic should be spread evenly as much as possible across the network.

Even spread of traffic in WSN lessens the creation of communication holes by reducing the hastily burnt energy of the nodes along the shortest/optimal path to sink, such as with application layer protocols, e.g. directed diffusion.

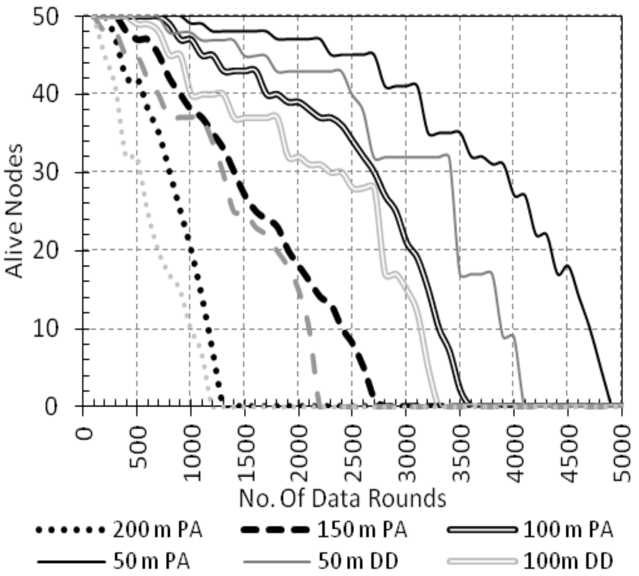


Fig.14. Comparison of received data

It also avoids wireless sensor network to experience disgraceful segregation and high latency by keeping more nodes alive in wireless sensor network to act as router(s) between sink and source.

In our sixth experiment, we check the health of the network by investigating the percentage of nodes alive in grid layout using a set of 50 nodes. We calculate and plot the number of alive nodes after every one thousand data round. A data round consisted of sinking dissemination interest query to the WSN and sensory nodes responding to that query. A round is considered finished when either:

1. All the data regarding to a query is reported to sink, or
2. The time allocated to nodes for responding to a query expires.

Fig. 14 shows the number of nodes alive for both approaches per hundred data rounds, and the total number of data rounds survived by each approach when the distance of the base station is varied (50 meters to 200 meters), i.e., placed further from the WSN.

With base station placed 200 meters away from the WSN, our proposed approach enables WSN to sustain a thousand extra data rounds.

The proposed approach stops the formation of communication holes in WSN; thereby prolonging the life of the WSN to sustain 20% extra data rounds as compared to directed diffusion.

6. Conclusion and Future work

We proposed a genetic algorithm-based solution in an agent-based sensor network. We described a sensor network as a dynamic latency metric and based on this an evolution algorithm is presented. By utilizing Dijkstra's algorithm, we were able to transform a topology of an interest region to a dynamic TSP. In a distributed system, mobile agent routing problem it turned out to be an NP-problem. Hence, we adopted genetic algorithm to compute an approximate routing. More study of genetic algorithm with improved fitness function is our next step to improve our approach.

References

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, A survey on sensor networks, in: IEEE Communications Magazine, vol. 40, no. 8, pp.102-114, August 2002.

[2] H. Zhang and H. Shen, Balancing Energy Consumption to Maximize Network Lifetime in Data-Gathering Sensor Networks, in: IEEE Trans. Parallel and Distributed Systems, vol. 20, no. 10, pp. 1526-1539, Oct. 2009.

[3] Xueyan Tang, Jianliang Xu, Adaptive Data Collection Strategies for Lifetime-Constrained Wireless Sensor Networks, in: IEEE Trans. Parallel Distrib. Syst. 19(6): 721-734 (2008)

[4] Fengyuan Ren, Jiao Zhang, Tao He, Chuang Lin, Sajal K. Das, EBRP: Energy-Balanced Routing Protocol for Data Gathering in Wireless Sensor Networks, in: IEEE Transactions on Parallel and Distributed Systems, vol. 22, no. 12, pp. 2108-2125, Dec. 2011.

[5] Min Chen, Taekyoung Kwon, Yong Yuan, Yanghee Choi and Victor C. M. Leung1, Mobile Agent-Based Directed Diffusion in wireless Sensor Networks, in: EURASIP Journal on Advances in Signal Processing, Article ID 36871, 2007.

[6] 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, in: Computer Journal, vol. 47, no. 4, pp. 448–460, 2004.

[7] H. Qi, S.S. Iyengar, and K. Chakrabarty, Multi-Resolution Data Integration Using Mobile Agents in Distributed Sensor Networks, in: IEEE Trans. Systems, Man, and Cybernetics Part C: Applications and Rev., vol. 31, no. 3, pp. 383-391, Aug., 2001.

[8] D. Massaguer, C. Fok, N. Venkatasubramanian, G. Lu, Exploring sensor networks using mobile agents, in: 5th International Conference on Autonomous Agents (AAMAS), pp. 323-325 2006.

[9] C.-L. Fok, G.-C. Roman, and C. Lu, "Mobile agent middleware for sensor networks: An application case study," In Proc. of the 4th Int. Conf. on Information Processing in Sensor Networks (IPSN'05), pp. 382-387. IEEE, April, 2005.

[10] Hairong Qi, Yingyue Xu, Xiaoling Wang, Mobile-agent-based Collaborative Signal and Information Processing in Sensor Networks, in: Proceeding of the IEEE, vol. 91, no. 8, pp.1172-1183, Aug. 2003.

[11] Q.Wu, N. S. V. Rao, J. Barhen, et al., On computing mobile agent routes for data fusion in distributed sensor networks, in: IEEE Transactions on Knowledge and Data Engineering, vol. 16, no. 6, pp. 740-753, 2004.

[12] R. Choudhury, S. Bandyopadhyay and K. Paul, MARP: A Multi-Agent Based Routing Protocol for Mobile Ad Hoc networks, in: Journal of Autonomous Agents and Multi-Agent Systems - Kluwer Publishers, 2004.

[13] E. Goldberg, Genetic algorithms, in: search, optimization, and machine learning, 1989.

[14] J. M. Rabaey, et al., PicoRadio supports ad hoc ultra low power wireless networking, in: IEEE Computer, Vol. 33, pp. 42-48, July

2000.

[15]E. Shakshuki and H. Malik, Multi-agent Based Clustering Approach to Wireless Sensor Networks, in: International Journal of Wireless and Mobile Computing (IJWMC), Inderscience Publisher, 2007.

[16]H. Malik, E. Shakshuki, T. Dewolf, M. Denko, Multi-agent System for Directed Diffusion in Wireless Sensor Networks, in: Third International Workshop on Heterogeneous Wireless Sensor Networks (HWISE 2007), in conjunction with the 21st International Conference on Advanced Information Networking and Applications, IEEE Computer Society, Niagara Falls, , Canada, 2007.

[17]S. Hussain, A. W. Matin, O. Islam, Genetic Algorithm for Hierarchical Wireless Sensor Networks, in: Journal of Networks, vol. 2, no. 5, Sep., 2007.

[18]J. Wu, Shengfang Yuan, S. Ji, Genyuan Zhou, Y. Wang, and Zilong Wang, Multi-agent system design and evaluation for collaborative wireless sensor network in large structure health monitoring, in: Journal of Expert Systems with Applications, pp. 2028-2036, June, 2009.

[19]BaiLi, Lynn Batten, Using mobile agents to recover from node and database compromise in path-based DoS attacks in wireless sensor networks, in: Journal of Network and Computer Applications, pp. 377– 387, Feb., 2008.

[20]Weilian Su and Tat L. Lim, Cross-layer design and optimization for wireless sensor networks, in: International Journal of Sensor Networks, vol. 6, pp. 3-12, January, 2009.

[21]E. Shakshuki, H. Malik, and T. Sheltami, Multi-agent-based clustering approach to wireless sensor networks, in: International Journal of Wireless and Mobile Computing, Inderscience, vol. 3, no. 3, pp. 165-176, March 2009.

[22]G.G. Mitchell, D. O'Donoghue, D. Barnes M. McCarville, GeneRepair - A Repair Operator for Genetic Algorithms, in: Proceeding (453) Artificial Intelligence and Applications, 2005.

[23]Web: http://robotics.eecs.berkeley.edu/~pister/SmartDust/ , downloaded July 2010.

[24]http://en.wikipedia.org/wiki/Sensor_Web#cite_note-Geosciences-0, downloaded July 2010.

[25]C. H. Papadimitriou, "The Euclidean Traveling Salesman Problem is NP-complete." Theoretical Computer Science, vol. 4, pp. 237-244, 1977.

[26]E. Shakshuki, H. Malik H, M. Denko, Software agent-based directed diffusion in wireless sensor network, in: Journal of Telecommunication Systems, Springer, vol. 38, pp. 161-174, 2008.

[27]Z. Guo, P. Chen, H. Zhang, M. Jiang, C. Li, IMA: An Integrated Monitoring Architecture With Sensor Networks, in: IEEE T. Instrumentation and Measurement , vol. 16, no. 5, pp. 1287-1295, 2012.

[28]C. Alippi, G. Anastasi, M. Di Francesco, M. Roveri, Energy Management in Wireless Sensor Networks with Energy-hugry Sensors, in: IEEE Instrumentation & Measurement Magazine, vol. 12, no. 2, pp.16-23, April 2009.

[29]B. Alspach, S. Locke, D. Witte, Hamilton spaces of Cyclic graphs on abelian groups, in: Discrete Mathematics, 82 (1990), pp. 113–126.

[30]E. A. Lee, Cyber physical systems: design challenges, in: Proceedings of the 11th IEEE Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing (ISORC '08), pp. 363–369, May 2008.

[31]E. A. Lee, Cyber-physical systems—are computing foundations adequate?, i Position Paper for NSF in: Workshop on Cyber-Physical Systems: Research Motivation, Techniques and Roadmap, Austin, Tex, USA, October 2006.

[32]J. Wan, H. Yan, H. Suo, and F. Li, Advances in cyber-physical systems research, in: KSII Transactions on Internet and Information Systems, vol. 5, no. 11, pp. 1891–1908, 2011.

[33] K. Lin, C. Min, Z. Sherali, Joel J.P.C., Rodrigues, Balancing energy consumption with mobile agents in wireless sensor networks, in: Future Generation Computer Systems, Volume 28, Issue 2, February 2012, Pages 446-456.

[34]Zhixin Liu, Qingchao Zheng, Liang Xue, Xinping Guan, A distributed energy-efficient clustering algorithm with improved coverage in wireless sensor networks, in: Future Generation Computer Systems, Volume 28, Issue 5, Pages 780-790, May 2012.

[35] Israel A. Wagner, Michael Lindenbaum, Alfred M. Bruckstein, ANTS: Agents on Networks, Trees, and Subgraphs, in: Future Generation Computer Systems, Volume 16, Issue 8, Pages 915-926, June 2000.

[36]A.D. Ahmed, O.K. Mohamed, M. Lewis, On balancing network traffic in path based multicast communication, in: Future Generation Computer Systems, pp. 805-811, 2006.

[37]H.L. Jason, B. Bobby, Y. Miao, L. Renato, A scalable key management and clustering scheme for wireless ad hoc and sensor networks, in: Future Generation Computer Systems 24 (2008) 860–869.

[38] S. Yashpal, K. Deep, S. Niranjan, Multiple Criteria Clustering of Mobile Agents in WSN, in: arXiv preprint arXiv:1207.3140 , 2012.

[39] C. Jiannong, S.K. Das. Mobile Agents in Networking and Distributed Computing. Vol. 3. Wiley-Interscience, 2012.

[40] Thomas, An FDL'ed Textbook on Sensor Networks (GNU FDL), 2006.

[41] D. Adams, D., April 2003. http://www.sics.se/~adam/contiki/contiki-2.0doc/a00435.html.

[42] W.R, Heinzelman, A. Chandrakasan, H. Balakrishnan, Energy efficient communication protocol for wireless microsensor networks. In: Proceedings of the IEEE Hawaii International Conference on System Sciences, January 2000.

[43] K. Fall, K. Varadhan, The ns Manual, May 2007.

[44] Network Simulator [ns2], http://www.isi.edu/nsnam/ns/, downloaded January 2013.

***Biographies (Text)**

Elhadi M. Shakshuki is a professor at the Jodrey School of Computer Science at Acadia University, Canada. He is the founder and the head of the Cooperative Intelligent Distributed Systems Group at the Computer Science Department, Acadia University. He received the BSc degree in computer engineering in 1984 from Tripoli University Libya, and the MSc and PhD degrees in systems design engineering respectively in 1994 and 2000, from the University of Waterloo, Canada. Dr. Shakshuki is an Adjunct Professor at Dalhousie University, Canada. He is on the editorial board of several international journals and contributed in many international conferences and workshops with different roles. He published over 150 research papers in international journals, conferences and workshops. He is the founder of the International Conference on Ambient Systems, Networks and Technologies. He is the co-founder of the International Conference on Mobile Web Information Systems. He is also a founder for other international symposia and workshops. He is a senior member of IEEE, and a member of ACM and APENS.

Haroon Malik received his B.S degree from Hamdard University, Pakistan in 1996 and MSc degree from Jodrey School of computer science, Acadia University, Canada in 2007. He has been studying at school of computing at Queen's University, Canada, as a PhD candidate. His current research interest lies in the intersection of Systems and Software engineering, Multi-agent systems and Wireless sensor network.

*Biographies (Photograph)

***Biographies (Photograph)**
Click here to download high resolution image

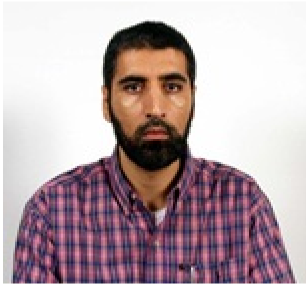
Highlights

1. An agent based routing approach for WSN to reduce the battery consumption.

2. Approach Increases the network lifetime by use of Dijkstra's algorithm and Agents.

3. Approach saves 20% more energy as compared to Directed Diffusion (DD).

4. The approach also decreases the network latency up to 50% as compared to DD.



Dr. Tarek R. Sheltami received the Ph.D. degree in electrical and computer engineering from the Department of Electrical and Computer Engineering, Queen's University, Kingston, ON, Canada, in 2003. He was with GamaEng Inc., Ottawa, Canada as a Consultant on wireless networks (in 2002–2004). He also worked in several joint projects with Nortel Network Corporation, Ottawa, Canada. He is currently an Associate Professor with the Computer Engineering Department, King Fahd University of Petroleum and Minerals (KFUPM), Dhahran, Saudi Arabia. He is also an Adjunct Professor with the Jodrey School of Computer Science, Acadia University, Wolfville, Canada. Before joining KFUPM, he was a Research Associate Professor with the School of Information Technology and Engineering, University of Ottawa, Ottawa, Canada. Dr. Sheltami has been a member of the Technical Program and Organizing Committees of several international IEEE conferences. He is the Co-founder of the IEEE International Symposium on Emerging Ubiquitous and Pervasive Systems, (EUPS) IEEE International Symposium on Applications of Ad hoc and Sensor Networks (AASNET). He is also the coordinator of the pervasive computing research group at KFUPM.