Computer Communications 33 (2010) 398–408

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

# Efficient path planning and data gathering protocols for the wireless sensor network

Jang-Ping Sheu [a], Prasan Kumar Sahoo [b,*], Chang-Hsin Su [c], Wei-Kai Hu [c]

[a] Department of Computer Science, National Tsing Hua University, Hsinchu 30013, Taiwan, ROC
[b] Department of Information Management, Vanung University, Chungli 32061, Taiwan, ROC
[c] Department of Computer Science and Information Engineering, National Central University, Chungli 32001, Taiwan, ROC

ARTICLE INFO

*Article history:*
Received 6 February 2008
Received in revised form 18 July 2009
Accepted 17 October 2009
Available online 25 October 2009

*Keywords:*
Wireless sensor network
Data gathering protocols
Path planning
Mobile sink

ABSTRACT

Wireless sensor network is commonly used to monitor certain region and to collect data for several applications. Normally, in wireless sensor network, data are routed in a multi-hop fashion towards a static sink. In this scenario, nodes closer to the sink become heavily involved in packet forwarding and their battery power is exhausted quickly. In this paper, an Infrastructure based Data Gathering Protocol (IDGP) and a Distributed Data Gathering Protocol (DDGP) are proposed to plan the data gathering path for a mobile sink. A *k*-hop relay mechanism is introduced to limit the number of hops for routing data to a mobile sink. In order to increase the efficiency of the data gathering, a cooperative environment among the sensor nodes and the mobile robot is proposed to formulate the data gathering path. Simulation results show that our data gathering protocols enable the nodes to transmit data with least number of hops and simultaneously reduce the data gathering path length traced by the mobile sink.

© 2009 Elsevier B.V. All rights reserved.

## 1. Introduction

Wireless sensor network (WSN) is composed of many small, inexpensive and battery-powered nodes with limited computing and wireless communication capabilities. Typical applications of wireless sensor network are environmental monitoring, military surveillance, health monitoring, target tracking, earthquake monitoring and many more [1–3]. It is envisioned that the nodes in WSN are used to collect useful information in the physical environments over a long time period for scientific data analysis, where battery is the main source of energy. The nodes are typically less mobile due to their unique application needs, substantially more resource constrained and more densely deployed than mobile ad hoc networks. In wireless sensor network, one or more sinks are present to subscribe specific data streams by expressing interest or queries. The sensors in the network act as sources, which detect environmental events and push relevant data to the appropriate subscriber sinks. A typical application in this network is gathering of sensed data at a distant base station (BS). It is observed that the power consumption due to communication among several degrees of nodes is usually a significant component of the total power consumption in a sensor network [4]. There is an energy cost for transmitting or receiving a packet in the radio electronics and there is a variable energy cost depending on the distance in transmission. Due to the $r^2$ or larger radio signal attenuation for a range $r$, it is important to limit transmission distance to conserve energy. Even though, there have been significant advances in recent years, more energy-efficient solutions are required within the communication stack for the conservation of the battery power.

Data aggregation and in-network processing techniques have been investigated recently as efficient approaches to achieve significant energy savings in WSN by combining data arriving from different sensor nodes at some aggregation points enroute, eliminating redundancy, and minimizing the number of transmission before forwarding data to the sinks. Hence, data fusion or aggregation has emerged as a useful paradigm in sensor networks. The key idea is to combine data from different sensors to eliminate redundant transmissions, and provide a rich, multidimensional view of the environment being monitored. In [5], authors argue that this paradigm shifts the focus from address centric approaches i.e. finding routes between pairs of end nodes to a more data centric approach i.e. finding routes from multiple sources to a destination that allows in network consolidation of data. Directed diffusion [6] is based on a network of nodes that can coordinate to perform distributed sensing of an environmental phenomenon. Such an approach achieves significant energy savings when intermediate nodes aggregate responses to queries. More recently, use of mobile robot has been explored for improving the networking facilities in the system. For dynamic deployment of sensor nodes [7], autonomous flying robots may be used for deploying sensor nodes and measuring the network connectivity. If some desired connectivity is missing, additional nodes are deployed by the robot to repair the connection. As proposed in localization method [8], the nodes

* Corresponding author. Tel.: +886 3 4515811 66614; fax: +886 3 4621348.
*E-mail addresses:* sheujp@cs.nthu.edu.tw (J.-P. Sheu), pksahoo@mail.vnu.edu.tw (P.K. Sahoo), suesh@axp1.csie.ncu.edu.tw (C.-H. Su), cathie@axp1.csie. ncu.edu.tw (W.-K. Hu).

0140-3664/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2009.10.011

in WSNs can use detection of a moving target to update their position estimations. This will help to decrease the mean network localization error and enhance the accuracy in position estimation.

Relocation of the sink can also be beneficial in applications involving real-time traffic. In such applications, data paths are carefully established so that certain end-to-end delay requirements are met. The quality of service achieved in these applications can start to diminish with increased volume of real-time data and most of the packets can miss their specified deadlines. In order to improve the timeliness in such situations, one of the solutions could be considered to move the sink to a location where volume of real-time data is high. In this case the traffic load among multiple nodes could be balanced and thereby decrease the rate of real-time packet loss. However, relocating the sink during regular network operation is very challenging. The basic issues are when it would make sense for the sink to be relocated, where the sink should go and how the data traffic will be handled during the movement of the sink. The relocation of the sink first has to be motivated by inefficient pattern of energy depletion or a non-tolerable increase in the number of missed deadlines when real-time packets are involved, even if it is the best possible network operation given the traffic distribution and network state at that time. Once the sink detects such conditions, it should identify its most suitable location in order to enhance network performance. It is worth noting that in some setups it may also be desirable to continually adjust the sink's location to better serve the application, even if its current position is not causing concerns. Hence, it is highly essential to plan the most suitable path for repositioning the sink time to time. In the path planning of the mobile sink, it is intuitive that we can let the sink to visit all nodes and to gather data directly from them. But, it is an NP-hard problem to find the shortest path [9], and in large-scale WSNs, latency of the data will be large, too. Moreover, there is much research about the problem of planning a path, which can completely cover an environment by a mobile sink. Commonly, the methods are spiral path and straight rows path [10] with backtracking to assure the whole network is visited.

In this paper we design algorithms to develop the distributed and infrastructure based path for a single mobile sink that can efficiently collect data from the static nodes of the network, which are deployed uniformly over a monitoring region. The rest of the paper is organized as follows. Related work, motivations and contribution of our work are presented in Section 2. Assumptions and system model of our algorithms are given in Section 3. Algorithms of our Infrastructure based Data Gathering Protocol is presented in Section 4 and the Distributed Data Gathering Protocol is described in Section 5. Performance analysis and simulation results of our work are presented in Section 6 and concluding remarks are made in Section 7 of the paper.

## 2. Related work

A mobile-assisted localization method [11,26], employs a mobile user or robot to assist in measuring the distance between node pairs until these distance can guarantee a unique localization. For improving communication performance in the network, Goldenberg et al. [12], study mobility as a control primitive in mobile networks and propose a distributed mobility control scheme to adjust node positions such that energy consumption in communication is minimized. To overcome network partitions, the Message Ferrying scheme [13] exploits the controlled mobility to transport data. They consider the situation where the network is composed of a set of static nodes, and then compute a ferry route, which is a tour of all the nodes minimizing the delay and meeting some bandwidth requirements. In order to minimize redundancy and load on the wireless sensor networks with a goal to conserve energy, data gathering and routing schemes for data fusion are extremely important. In [31], authors have analyzed the routing for the data gathering and fusion [30] and classify the wireless sensor networks into three categories, such as routing-driven, coding-driven, and fusion-driven data fusion.

There is a wide range of data gathering applications in target tracking, hazard detection, environmental monitoring and battlefield surveillance. Data gathering [14,27,32] is another application of the mobile robot, which describes the use of a predictable mobile agent. A network node is mounted on a commuter bus and acted as a base station for collecting data from several static sensor nodes along its navigation path. In [15], the author exploits mobile entities called Mobile Ubiquitous LAN Extensions (MULEs) to pick up data from sensor nodes. In the MULEs architecture, sensor nodes transmit data only over a short range that requires less transmission power. The fluid infrastructure proposed in [16], uses the mobile robot to gather sensor node's data. Also, they discuss the motion control of the mobile robot and communication protocol between the sensor nodes. The objective of such research in [14–16] is meant for reducing the communication energy required at the sensor nodes and to maximize the sensor network lifetime. In [28], the authors propose data gathering protocols, where the communicating sensor nodes have to be sent their messages to a distant central node, called the base station or sink over shortest path. The authors in [29] propose a hierarchical grid structure to reduce the total energy consumption, and utilizes a tree architecture to decrease the transmission delay. In this work, the energy consumption is reduced by scheduling the redundant nodes in a grid into sleeping mode.

The authors in [17] propose methods to collect data efficiently and to transmit them to the base station, taking location of sensors and the base station and the available energy at each sensor, such that the system lifetime is maximized. They present polynomial time algorithms to solve the data gathering problem, with and without data aggregation. However, how to attain the desired tradeoff between the delay experienced by the sensors and the lifetime achieved by the system is not studied. A spanning tree method that is based on breadth-first search [18] proposes to reduce the communication path length and to turn off some sensor nodes radio components when they do not participate in data forwarding. But, the nodes communicate with the base station by delivering data across multiple hops, and thereby consume more energy by communication. A distributed routing protocol that implements the energy and latency aware data gathering paradigm and explicitly considers the routing under both the constraints of energy efficiency and path length is proposed in [19]. However, this protocol uses a longer data routing path to forward packets to the base station and the nodes around the sink become hot-spots and run out of energy quickly. The authors in [20] design a hierarchical and scalable data gathering protocol to demonstrate the data correlations in sensor readings and to minimize communications cost in the data gathering process towards the sink. This protocol adopts the clustered strategy using useful principles of the ant colony behavior and achieves the objective of uniform cluster formation by exploiting foraging and brood sorting.

A distributed data gathering algorithm is proposed in [21] that allows to avoid the use of global knowledge of the system in order to collect all the information disseminated among the sensor nodes deployed in a region. The work evaluates the effects on the energy consumption of uniformly distributed nodes in a given area and the effect of having a different number of cluster heads transmitting to the base station. Although, the work demands the improvement in network longevity as compared to similar data gathering protocols, it still uses the longer data routing path and does not consider the fault-tolerant aggregation process. The impacts of different features and behavior of mobile sinks on the hybrid wireless sensor networks is analyzed in [22]. Their simulation results show that, instead of deploying mobile sinks as much as possible, choosing appropriate number, transmission range, velocity and gathering mode of the sink nodes can significantly decrease the average end-to-end data delivery delay and improve the energy conservation. Though the paper proposes the mobility and traffic model to analyze end-to-end data delivery delay, data gathering methods are not discussed in the work. Moreover the work considers the presence of multiple sink nodes in the wireless sensor network. In [23], authors investigate the potential of gateway repositioning and address issues related to when should the gateway be relocated, where it would be moved to and how to handle its motion. They use the search heuristics approach to move the gateway towards the sources of largest traffic. But significant challenges are encountered how to design the data gathering path for the mobile robot and how to improve the efficiency of data gathering with help of a mobile robot. In this paper, we propose the efficient methods to plan the data gathering path and motivations and contributions of our work are summarized as follows.

### 2.1. Motivations and contributions

Normally, sensor nodes are deployed over a large geographic area randomly, where they can not transmit the sensing data directly to the sink and need other nodes to relay their data. Under this situation, sensor nodes have to not only send their data, but also relay the data for other nodes repeatedly. Due to this multi-hop relays, nodes located nearby the sink have to forward several packets to the sink than others. Consequently, those nodes form the routing hot spots and run out of energy quickly. However, replenishing energy via replacing batteries on hundreds of nodes and mainly in possibly harsh terrains is infeasible. The basic operation in such a system could be the systematic gathering of sensed data to be eventually transmitted to a base station for processing. Further, the key challenge in such data gathering is conserving the sensor energies, so as to maximize their lifetime. Hence, it is one of the major challenges to reduce the high energy expenditure in multi-hop routing and to extend the sensor network's lifetime by gathering data from the nodes of the network with shortest routing path. In this paper, we propose protocols to navigate the data gathering path for a mobile robot that collects data from the nodes with limited number of hop counts. The main contributions of our work can be summarized as follows.

- An infrastructure based data gathering protocol (IDGP) is proposed, which plans the data gathering path for a mobile robot to collect data from limited number of hops, thereby reducing the longer routing path.
- A distributed data gathering protocol (DDGP) is proposed, which helps the mobile robot to navigate its path through the network in a distributed fashion. In this protocol, the routing of data is also limited with few number of hops.
- Our data gathering and path planning protocols use $k$-hop relay mechanism to limit the number of hops, where $k$ is a user defined number and both protocols ensure a significant tradeoff in terms of node energy balancing and data gathering path length.
- In our protocols, the data gathering path lengths can be made as small as possible. Thus, end-to-end packet transmission delay is restricted to match several real time applications like collecting temperature and moisture readings of the nodes frequently. Besides, our protocols prevent too many nodes to take part in routing, and thereby to reduce the overall energy consumption.

## 3. System model

Consider a large-scale sensor network of $n$ static nodes distributed uniformly over a rectangular region and a mobile sink (throughout the paper mobile sink is referred as mobile robot) that acts as the base station with unlimited energy. The locations of the sensors and vertices of the monitoring region are fixed and known a priori through GPS or some localization systems [24,25]. The sensor nodes are homogeneous and each sensor node has constrained battery energy. Each sensor node senses environment and produces some information as it monitors its vicinity. It is assumed that each node generates one data packet per time unit to be transmitted to the base station. For simplicity, we refer to each time unit as a round. Further, each sensor $i$ has a battery with finite, non-replenishable energy $E_i$. Whenever a sensor transmits or receives a data packet, it consumes some energy from its battery. The energy consumed to transmit $n$ bits for $d$ unit distance is given by:

$$E_T = nE_e + n\epsilon d^4 \tag{1}$$

Similarly, taking the cost of beam forming approach that reduces energy consumption, the energy consumed to receive $n$ bits is given by:

$$E_R = nE_e + nE_{BF} \tag{2}$$

where, energy consumed for short range transmission ($E_T$) $\epsilon = 10$ pJ/bit/m$^2$, energy consumed in the electronics circuit ($E_e$) to transmit or receive the signal is $E_e = 50$ nJ/bit and energy consumed for beam forming $E_{BF} = 5$ nJ/bit.

### 3.1. k-hop relay mechanism

$k$-hop relay mechanism is defined as the maximum number of hops required by a node to transmit data to the mobile robot. The value of $k$, which is informed to all nodes prior to formation of data gathering path is user-defined and also application oriented. As shown in Fig. 1, the mobile robot, which is $k$ hops away from node $A$ collects data from nodes $A, B, \ldots, D$ and this procedure is known as $k$-hop relay mechanism. It is to be noted that though smaller value of $k$ can minimize the communication cost, it makes the data gathering path longer. Conversely, larger value of $k$ can increase the communication cost, but produces a shorter data gathering path. Hence, it is a tradeoff between the communication cost and data gathering path length to decide the value of $k$, which could be application oriented. For example, if user wants to collect data more frequently and within short span of time by the sensors such as temperature or moisture reading, a shorter data gathering path could be considered and thus large value of $k$ may be taken.

### 3.2. Layer value

To comply with the $k$-hop relay mechanism, each node of the network determines its layer value between 0 and $k - 1$. If $l$ is layer value of a node, for $l < k$, it relays its data to a node having layer value more than $l$, which is ultimately sent to the mobile robot.

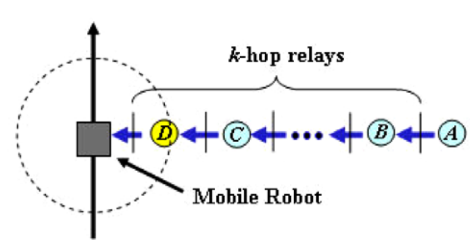


**Fig. 1.** An example of $k$-hop relay mechanism.

As shown in Fig. 2, when a mobile robot travels along the data gathering path, represented by the solid lines, it collects data from the nodes within its communication range. Nodes $A$ and $B$ that are on the $(k-1)$th layers, collect data from the nodes of layers 0 through $k-2$, combine their own data and forward it to the mobile robot.

### 3.3. Hop distance

Normally, radius of a circle centered with a sensor is known as the communication range ( $R_c$) of the sensor. However, in practice, distance between two nodes with exact value of communication range $R_c$ is not realized and can create communication problem. Hence, we consider the worst case in wireless communication and define a most suitable value of *hop distance* between any two nodes. As shown in Fig. 3, distance between nodes $A$ and $C$ is $R_c + d$ units, where $d$ approaches to zero. Since, node $A$ and $C$ cannot communicate with each other, they need node $B$ to be the relay node. Hence, we derive the *hop distance* between any two nodes is equal to $\frac{R_c}{2}$, as given in Eq. (1), where $R_c$ is communication range of a node.

$$2 \times hop\_distance = \lim_{d \to 0}(R_c + d) \approx R_c \tag{3}$$

## 4. Infrastructure based data gathering protocol (IDGP)

In order to reduce the number of hops and to gather data from the nodes deployed over a rectangular monitoring region, we propose an infrastructure based data gathering protocol (IDGP). In IDGP, a mobile robot decides its own data gathering path and broadcasts the control message to all nodes of the network, which contains the value of $k$ and data gathering path information. Upon receiving the control message, each node determines its layer value according to the data gathering path information. Detail formulation of data gathering path by the mobile robot and determination of layer value by the static nodes are described as follows.

### 4.1. Data gathering path formulation

Let us consider a rectangular deployed region, as shown in Fig. 4, whose coordinates of vertices $C_1$, $C_2$, $C_3$ and $C_4$ are known. Taking, $L_1$ and $L_2$ be the internal angle bisectors of $C_2C_1C_4$ and $C_1C_4C_3$, respectively, equation of $L_1$ and $L_2$ can be expressed as:

$$L_1 : \frac{a_1x + b_1y + c_1}{\sqrt{a_1^2 + b_1^2}} = -\frac{a_2x + b_2y + c_2}{\sqrt{a_2^2 + b_2^2}} \tag{4}$$

$$L_2 : \frac{a_3x + b_3y + c_3}{\sqrt{a_3^2 + b_3^2}} = -\frac{a_2x + b_2y + c_2}{\sqrt{a_2^2 + b_2^2}} \tag{5}$$

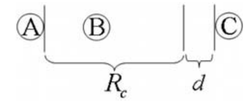

**Fig. 3.** Estimation of *hop distance* between any two nodes.

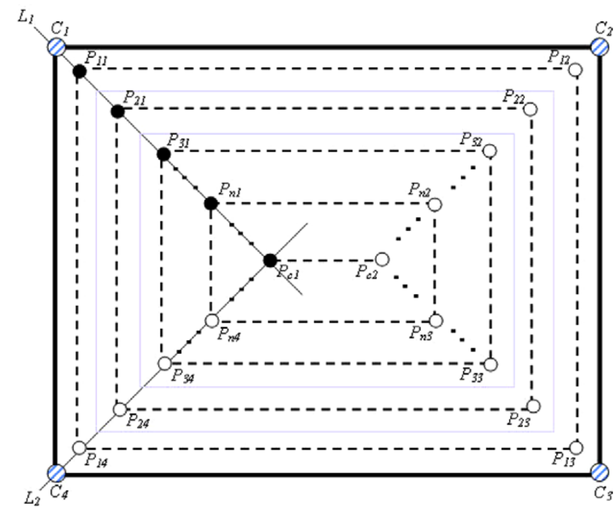

**Fig. 4.** Data gathering path of a mobile robot in IDGP.

It is assumed that sensors are regularly deployed including boundary of the monitoring region. Assuming the dotted lines as the data gathering path of the mobile robot as shown in Fig. 4, it is obvious that the first data gathering path closer to the boundary lines are $k$ hops away, whereas the next paths are $2k$ hops away from them. Let, $P_{c1}$ be the point of intersection of lines $L_1$ and $L_2$, and $(x_{P_{c1}},\ y_{P_{c1}})$ be the coordinate of $P_{c1}$. If $P_{11}$ is any point on the line $L_1$, the total hop distance $\frac{R_c}{2} \times k$ of those $k$ hops could be equal to $\overline{C_1P_{11}}$, where $R_c$ is communication range of a node. Hence, coordinate $(x_{P_{11}}, y_{P_{11}})$ of $P_{11}$ can be estimated using the coordinate of $C_1$ and length of $\overline{C_1P_{11}}$. Since, total number of hops between any two consecutive points among $P_{21}$, $P_{31}$, ..., $P_{n1}$ is $2k$, the total hop distance between any two consecutive points must be $R_c \times k$. Taking coordinate of $P_{11}$ as $(x_{P_{11}}, y_{P_{11}})$, and the hop distance as the ratios of division, coordinate of any point $P_{ij}$ can be obtained from the following equation:

$$P_{ij}(x_{P_{ij}}, y_{P_{ij}}) = (2i-1)(x_{P_{1j}}, y_{P_{1j}}) - 2(i-1)(x_{C_j}, y_{C_j}) \tag{6}$$

where, $i$=2,3,4,...and $j$=1,2,3,...

After formulating equation of the data gathering path, the mobile robot floods the control message as described in the next subsection so that nodes can determine their layer values.

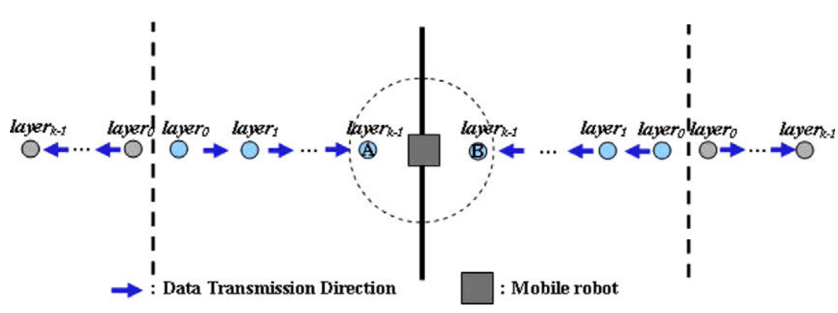

**Fig. 2.** Example of layer values based on the $k$-hop relay mechanism.

### 4.2. Control messaging

The objective of control messaging is to inform all nodes of the network about the proposed path planned by the mobile robot. As shown in Fig. 5(a), the control message has three fields, which contains equation of each data gathering path, value of $k$ and *layer_msg*. The *layer_msg* field is again subdivided into two fields as *source ID* and *layer_value*. The *source ID* stores *ID* of a sensor that broadcasts the message, and *layer_value* gives layer number of the source node that it belongs to. Upon receiving the control message, each sensor node records respective information of the control message to its *Layer Message table*(LM Table) to determine its own layer. The format of the *LM table* is shown in Fig. 5(b) and determination of layer by each node is given in the *Layer Determination Algorithm*.

### 4.3. Layer determination algorithm

Upon receiving the control message, each node derives the equation of data gathering path $\overline{P_{i1}P_{i2}}$, $\overline{P_{i2}P_{i3}}$, $\overline{P_{i3}P_{i4}}$, and $\overline{P_{i4}P_{i1}}$, where $i$ is an integer and $i > 0$. Let, $ax + by + c = 0$ be the equation of the path, and $d_i$ be the distance of the $i$th node from the data gathering path. A node $i$ assigns *layer_valuek* $- 1$, if it can communicate with the mobile robot directly. In this case, $d_i < R_c$. After determining the layer, a node $i$ broadcasts the *layer_msg*$(i, k-1)$. Nodes, which cannot communicate directly with the mobile robot wait for the *layer_msg* from other nodes, who have already determined their layers. If a node $j$ has not determined its layer yet, but receives the first *layer_msg*, it waits $T$ units time to collect all *layer_msg* of its 1-hop neighbors, and then records the information of the *layer_msg* to its *LM table*. We assume that the waiting time $T$ is equal to the propagation time of a packet between two farthest nodes of the network. After time $T$ is out, it selects a node $q$ as a relay node, which has maximum *layer_value* in its *LM table*. The layer of sensor node $j$ is always less by one from the layer of node $q$. The reason of selecting a node with maximum *layer_value* is that the node is nearest from the data gathering path among all of its 1-hop neighbors. The detail algorithm of Layer Determination procedure is given in Table 1. It is to be noted that the value of hop counts $k$ must affect the length of data gathering path, which is proved in Lemma 1.

**Lemma 1.** *In IDGP, length of data gathering path must be shorter for larger number of hop counts.*

**Proof.** Let $a$ and $b$ be the length and breadth of the monitoring region, respectively and $g$ be the length of each data gathering path. It is obvious that the optimum value of data gathering path length must be equal to perimeter of the monitoring region, i.e. $2(a + b)$.

Let, $k_1$ be the hop count corresponding to data gathering path $g_1$ with length $a_1$ and breadth $b_1$ and $k_2$ be the hop count corresponding to data gathering path $g_2$ with length $a_2$ and breadth $b_2$, such that $k_1 > k_2$. It is to be proved that $g_1 < g_2$.

From our assumptions, it is clear that $g_1 = 2(a_1 + b_1)$ and $g_2 = 2(a_1 + b_1)$.

Since, $k_1 > k_2 \Rightarrow a_1 < a_2$ and $b_1 < b_2 \Rightarrow 2(a_1 + b_1) < 2(a_2 + b_2)$ $\Rightarrow g_1 < g_2$ □

**Table 1**
ALGORITHM 1: IDGP Layer Determination Algorithm.

**Initialization:**
1. Mobile robot defines value of $k$;
2. Determines its data gathering path based on value of $k$;
3. Floods the control packet that contains equations of end point of the path;

**Step 1:** Upon receiving the control message, a node checks,
1. If it is within communication range of the robot;
2. **If** (it is within communication range)
3. {
4. Set *layer_value* to $k - 1$;
5. Broadcast *layer_msg*;
6. }
7. **Else:**
8. {
9. Wait for $T$ units to collect all *layer_msg* from its 1-hop neighbors;
10. Record information of *layer_msg* to its *LM table*;
11. }

**Step 2:** After time $T$ is out,
1. Select another node from its *LM table*, having maximum *layer_value*;
2. Set itself as a relay node;
3. Set *Layer_value*=selected node's ( *layer_value* $- 1$);
4. Broadcast *layer_msg*;

## 5. Distributed Data Gathering Protocol (DDGP)

The Distributed Data Gathering Protocol (DDGP) proposes to construct a distributed data gathering path by the nodes of the network and describes how to navigate the paths by the mobile robot. As opposed to IDGP, the nodes in DDGP first determine their own layers and design the data gathering path for the mobile robot. The robot then navigates the data gathering path based on the layer value of the nodes. It is assumed that each sensor node keeps the location information of its 1-hop neighbors and the mobile robot selects the value of $k$ in advance. DDGP comprises two parts, namely *Layer Determination Algorithm*, that lets the sensor nodes to determine its layer and *Path Guiding Algorithm* that guides the data gathering path in advance. Before going through these algorithms, we define here few terms and notations.

### 5.1. Notations and definitions

$k$: Maximum number of hops, up to which data should be relayed.
$N_1(i)$: Set of one-hop neighbors of node $i$.
$UN\_N_1(i)$: Set of one-hop neighbors of node $i$, which has not determined its layer yet.
$d(i, j)$: Euclidean distance between any two nodes $i$ and $j$.
$AP_c$: Line connecting to any node $A$ with the central point $P_c$.
*D_border*: Shortest distance of a node from the four boundaries of the deployed region.
*layer_x*: $x$th layer of a node.

**Definition 1. Central Point** ($P_c$): Point of intersection of two diagonals of the rectangular deployed region. It is to be noted that unlike IDGP, central point of DDGP is only one.

| Equations | k | layer_msg |
|---|---|---|

(a)

| LM table (node i) | |
|---|---|
| source | layer_value |

(b)

**Fig. 5.** (a) Format of control message in IDGP and (b) records of the LM table in IDGP.

**Definition 2. Layer Cycle**: Total range of the layers from which a mobile robot gathers the sensed data. Since, value of $k$ is predefined by the mobile robot and value of the layers range from 0 to $k-1$, layer cycle spans from $k-1$ to 0 along the left and right sides of the data gathering path, making $2k+1$ number of layers in total. Thus, each layer in the layer cycle is represented by a value from 0 to $2k$, as shown in Fig. 6.

**Definition 3. Message Format**: There are three types of messages used in DDGP. They are: *start_msg*, *layer_msg*, and *assign_msg*. Each sensor node determines its layer value from the contents of the message that it receives.

*start_msg*: This message contains value of $k$, central point $P_c$ and four corner points of the rectangular deployed region.
*layer_msg*: Format of this message is same as the *layer_msg* of IDGP. However, we add a new element *next_host* to the *LM table*, as shown in Fig. 7, which is recommended by the *source* to be in the next layer.
*assign_msg*: This message contains list of nodes, which are assigned to $layer_k$. In order to assure that $layer_k$ is connected, each sensor node in $layer_k$ uses this message to assign one or more nodes present in $layer_k$.

### 5.2. Layer Determination Algorithm

The *Layer Determination Algorithm* comprises the *Candidacy Condition* and *kth layer assignment rule*, as described in the following subsections.

#### 5.2.1. Candidacy condition

This condition is used by the nodes to determine their layers between border of the rectangular deployed region and central point $P_c$. For any node $i$, which has not determined its layer value yet, if $d(i,P_c) > d(j,P_c)$, for all node $j \in UN\_N_1(i)$, node $i$ satisfies the *Candidacy Condition*. Nodes, which are one-hop neighbors to each other and belong to the set of undetermined nodes can execute the candidacy condition. As per this condition, nodes farthest from $P_c$ can satisfy this and follow the *Layer Determination Algorithm*. As shown in Fig. 8, nodes with gray color have already determined their layer. Then question arise, which node should go first to determine its layer out of the nodes $A$, $B$ and $C$. Suppose, node $A$ first checks its undetermined neighbors $B$, $C \in UN\_N_1(A)$. Since, $d(A,P_c) > d(B,P_c) > d(C,P_c)$, node $A$ satisfies the candidacy condition and follows the *Layer Determination Algorithm*.

Along with the candidacy condition, we describe here how to find the *next_host* and *next_source* to form the next layer.

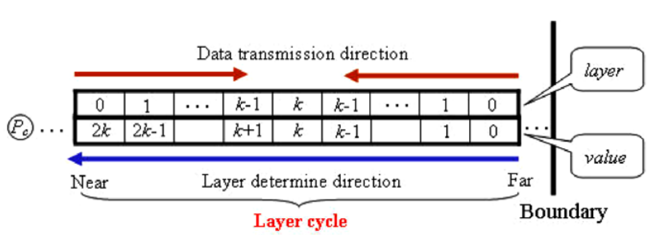


**Fig. 6.** Explanation of *Layer Cycle* range.

| LM Table (*node i*) | | |
|---|---|---|
| source | layer_value | next_host |

**Fig. 7.** Format of the *LM table* in DDGP.

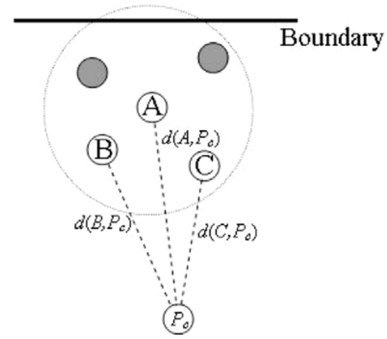


**Fig. 8.** An example of *Candidacy condition*, which verifies the candidate among nodes $A$, $B$ and $C$ to go for executing the Layer Determination Algorithm.

***next_host*** and ***next_source***: If a node $i$ has determined its layer, and recommends another node $p \in UN\_N_1(i)$ to be in the next layer, then $p$ is called the *next_host* of node $i$, if $D\_border_p$ is the maximum value among *D_border* of all nodes of $UN\_N_1(i)$. Besides, node $i$ is the *next_source* of node $p$. For example, as shown in Fig. 9, let $A$ be a node, which has already determined its layer value and nodes $B$, $C$ and $D$ are its one-hop neighbors, and have not determined their layer values yet. Since, $D\_border_D > D\_border_C > D\_border_B$, node $A$ recommends node $D$ as its *next_host*, and node $A$ is the *next_source* of node $D$. It is to be noted that node, which has already determined its layer has to recommend only one node as its *next_host*. Finally, in order to let the $k$th layer be connected, we use the *kth layer assignment rule*, as follows.

#### 5.2.2. kth layer assignment rule

As per this rule, each node $A$ in $layer_k$, checks both sides of the line $L$ to verify, if there exists any node in $N_1(A)$, as well as in $layer_k$. If there is no node in $layer_k$ on both sides of line $L$, node $A$ assigns $B$ and $D$, whose *D_border* is smallest along both sides, and $B, D \in UN\_N_1(A)$, as shown in Fig. 10(a). If node $A$ has only one node in $N_1(A)$ of $layer_k$ along one side of the line $L$, node $A$ simply assigns a node having smallest *D_border* in $UN\_N_1(A)$, as shown in Fig. 10(b) and (c). After *kth assignment rule* is executed, node $A$ broadcasts the *assign_msg* to inform about the node, which has assigned to $layer_k$. Then, the *Layer Determination Algorithm* is executed step by step upon receiving the following messages and implementing them as follows.

*Receiving start_msg*: If sensor node $i$ receives the *start_msg*, it checks if it satisfies the *candidacy condition*. If so, it is obvious that node $i$ is at the start of the *Layer Cycle*. Hence, layer of node $i$ is set to $layer_0$ and *layer_value* is set to 0. After determining the layer, node $i$ broadcasts the $layer\_msg(i, 0, next\_host)$. However, if node $i$ does not satisfy the *candidacy condition*, it assumes that at least one of its one-hop neighbors still belong to $UN\_N_1(i)$, which is farther than node $i$ to the central point $P_c$. Hence, node $i$ waits until that node determines its layer, and then node $i$ determines its own layer.

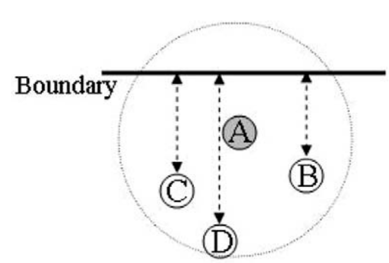


**Fig. 9.** Sensor node $D$ is the *next_host* of sensor node $A$.

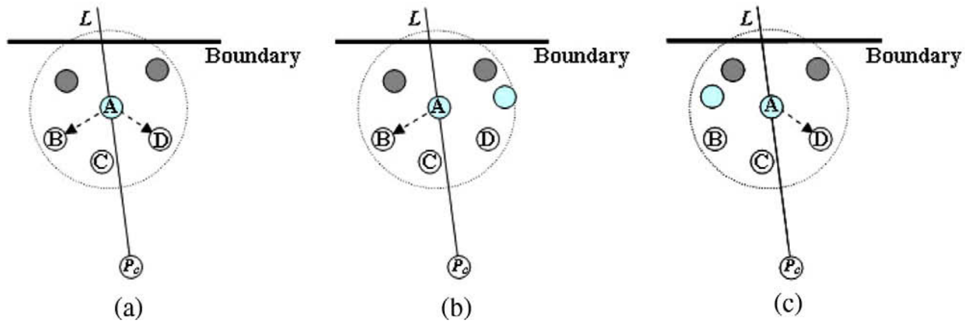


**Fig. 10.** The $k$th *layer connected assignment rule* (a) sensor node $A$ assigns $B$ and $D$, since there is no sensor node in $layer_k$ on both sides of the line $L$, and *D_border* of both $B$ and $D$ are smallest on both sides of $L$; (b) sensor node $A$ just to assign one sensor node $B$ on one side of the line $L$, which does not have any sensor node in $layer_k$; (c) sensor node $A$ just to assign one sensor node $D$ on one side of the line $L$, which does not have any sensor node also in $layer_k$.

*Receiving layer_msg*: When sensor node $i$ receives the *layer_msg*, it records the information in the *layer_msg* to its *LM table*. If node $i$ has not determined its layer yet, it verifies the *source ID* of *layer_msg* from which it has received the message. If the *layer_msg* is from $layer_k$ and node $i$ has not been assigned to $layer_k$ yet, the *layer_value* of node $i$ will be $k+1$, and the *layer_msg*($i, k+1$, *next_host*) is then broadcast. Else, node $i$ must verify the *candidacy condition*. If node $i$ satisfies the candidacy condition, in order to avoid layer-determined sensor nodes exceeding $k$ hops in relaying sensed data, it selects the *next_source j* from the *LM table*, having maximum *layer_value*. Under this condition, three cases can be considered as follows.

*Case 1*: *next_source j*'s *layer_value* $< k-1$ or *next_source j*'s *layer_value* $> k+1$. In this case, *layer_value* of sensor node $i$ just equals to *next_source j*'s *layer_value* $+1$. After that, sensor node $i$ broadcasts the *layer_msg*.
*Case 2*: *next_source j*'s *layer_value* $= k-1$. In this case, *layer_value* of node $i$ is also equal to node $j$'s *layer_value* $+1$. In order to let the nodes be connected in $layer_k$, node $i$, which is in $layer_k$ executes the *kth layer assignment rule* and then broadcasts the *layer_msg*.
*Case 3*: If there is no *next_source*, in order to enlarge the *Layer Cycle*, minimum *layer_valuex* is selected from the *LM table*. If that value is more than $k$, node $i$'s *layer_value* is set to be $x+1$, else it is set to be $x$.
*Receiving assign_msg*: If sensor node $i$ receives the *assign_msg*, it is bound to be in $layer_k$ and *layer_value* is set to be $k$. Then, it must execute the *kth layer assignment rule*, after which, it broadcasts the *assign_msg* and *layer_msg* in order.
*Special case:* In DDGP, since the final destination of mobile robot is $P_c$, nodes that are nearer to $P_c$ and within communication range of each other, for them *layer_value* is set to be $k$. Then, the *layer_value* is transformed into layer. Since, the *layer_value* lies between 0 to $2k$, in every layer cycle, it can be transformed as follows:

$$layer = \begin{cases} layer\_value, & \text{if, } layer\_value \leqslant k \\ 2k - layer\_value, & \text{otherwise.} \end{cases}$$

The algorithm of DDGP is given in Table 2. Besides, in order to justify the performance of DDGP is better than IDGP, Lemma 2 is given as follows.

**Lemma 2.** *Total data gathering path length of DDGP must be less than the data gathering path length of IDGP.*

**Proof.** Let $l_I$ and $l_D$ be the layer values and $p_I$ and $p_D$ be the total data gathering path length of IDGP and DDGP, respectively. It is to be proved that $p_D < p_I$.

From the assumptions of both protocols, maximum number of hops in both IDGP and DDGP are fixed.$\Rightarrow l_D \leqslant l_I$.

If $l_D < l_I \Rightarrow p_D < p_I$

Now the relation between the path lengths is to be verified for $l_D = l_I$. It is to be noted that the central point, ($P_c$) in DDGP is the point of intersection of two diagonals of the rectangular deployed region.

$\Rightarrow$ the last layer of DDGP must be a point, whereas the last layer in IDGP must be a straight line.

$\Rightarrow$ for $l_D = l_I, p_D < p_I$.

Hence, for any case we can conclude that $p_D < p_I$. □

### 5.3. Path guiding algorithm

Once the distributed data gathering path is constructed by the sensors, the mobile robot has to navigate it to collect data from each $k$-hop of nodes. The mobile robot first navigates the path along the nodes in $layer_k$ after each node's *Layer Determination Algorithm* is executed. In order to get each node's location information and corresponding layer value, the mobile robot exchanges the *path_search_msg* among all nodes of the network. Upon receiving this message, sensor nodes present in $layer_k$ have to reply. The reply message includes the *source ID* and location information of the node. Initially, the mobile robot navigates towards the central point $P_c$, and broadcasts the *path_search_msg* at every navigating distance, which is equal to the communication range, until it receives response from the nodes in $layer_k$. Since, the mobile robot may receive multiple reply packets, it chooses the nearest node as its destination. Upon arriving at the destination, the mobile robot starts gathering data from the nodes, which are in $layer_k$. When gathering data from $layer_k$ is finished, the mobile robot selects the next node as its destination, which is nearest to it and is along the left or right side separated by the line connecting to the mobile robot and $P_c$. If the next destination has already been visited, the mobile robot navigates towards $P_c$ again and the procedure is repeated until the mobile robot is arrived at $P_c$.

## 6. Performance evaluation

In order to evaluate the performance of IDGP and DDGP, our simulation is performed in a rectangular deployed region of fixed area of $1000\,\text{m} \times 500\,\text{m}$. The simulation is done using C++and nodes are uniformly deployed over the monitoring region. To avoid communication holes among the nodes, distance between any two nodes is maintained at $R_c + d$, where $R_c$ is communication range of each node and $d$ is a parameter that closes to zero. The hop distance is considered to be $2 \times hop\_distance = \lim_{d\to 0}(R_c + d) \approx R_c$, so that total range of $k$-hop relays is equal to $\frac{R_c}{2} \times k$. Besides, the distance between any two layers of nodes is considered to be $\frac{\sqrt{3}R_c}{2}$ so

**Table 2**
ALGORITHM 2: DDGP Layer Determination Algorithm.

**Initialization:**
1. Mobile robot selects the value of $k$;
2. Floods *start_msg*;

**Step 1:** Upon receiving *start_msg*, each node executes the candidacy condition;
1. **If** (Condition is satisfied)
2. { Set *layer_value* to 0;
3. Broadcast *layer_msg*; }
4. **Else:**
5. Wait to receive the *layer_msg*;

**Step 2:** Upon receiving *layer_msg*
1. **If** (Node has determined its layer)
2. Records the *layer_msg* to its *LM table*;
3. **Else:**
4. Checks the *layer_value* in *layer_msg*;
5. **If** (*layer_value*=$k$)
6. { Set *layer_value* to $k+1$;
7. Broadcast *layer_msg*; }
8. **Else:**
9. It verifies the candidacy condition;
10. **If**(candidacy condition is satisfied)
11. Select the *next_source* having maximum *layer_value*;
12. **If**(*next_source*'s *layer_value* $< k-1 ||$ *next_source*'s *layer_value* $> k+1$)
13. {
14. Set it's *layer_value*=*next_source*'s *layer_value* $+1$;
15. Broadcast *layer_msg*;
16. }
17. **Else:If**( *next_source*'s *layer_value* $= k-1$)
18. {
19. Set it's *layer_value* to $k$;
20. Executes $k$th *layer assignment rule*;
21. Broadcasts *layer_msg* in order;
22. }
23. **Else:**
24. Selects the minimum *layer_value* from *LM table*.
25. **If**(MIN( *layer_value*) $< k$)
26. {
27. Set it's *layer_value* to MIN(*layer_value*);
28. Broadcast *layer_msg*;
29. }
30. **Else:**
31. {
32. Set *layer_value* to MIN( *layer_value* $+1$);
33. Broadcast *layer_msg*;
34. }

**Step 3:** Upon receiving *assign_msg*
1. Set node's *layer_value* to $k$;
2. Execute $k$th *layer assignment rule*
3. Broadcast *layer_msg* in order;

that each three nodes of the network can form vertices of an equilateral triangle. Thus, the deployment strategy totally ensures our assumption that there is no communication hole in the network. All nodes use the CSMA/CA channel access mechanism and broadcasts the control packets in each 10 ms. In IDGP, simulation is performed by pre-assigning the data gathering paths for different densities of nodes, whereas, in DDGP, the layer values of different nodes are determined first. Various parameters like mobile robot's data gathering path length, average hop counts of the nodes and node density are simulated and the results are obtained as discussed below.

### 6.1. Data gathering path tracing

In order to get physical perception of the nature of data gathering paths, we simulated both IDGP and DDGP for fixed number of nodes and fixed value of $k$ to trace different paths, as shown in Fig. 11(a) and (b) for IDGP and DDGP, respectively. It is to be noted that the data gathering path of IDGP is totally regular, whereas it is irregular in case of DDGP. From the simulation results, we find the nature of path as per our expectation. In IDGP, since, we consider the first data gathering path $\frac{R_c}{2} \times k$ units away from each corner of the deployed region, all paths in the simulation result are regular. Since, the data gathering path in DDGP is estimated distributively based on the $k$ number of layers, irregular nature of the path is found in the data gathering path tracing.

### 6.2. Data gathering path length

In this section, the effects of the layer values ($k$) and node numbers on the total path length are analyzed. By implementing both IDGP and DDGP algorithms in the simulation, the total data gathering path length covered by the robot is measured. Initially, several paths are generated for different values of $k$ and node numbers. As

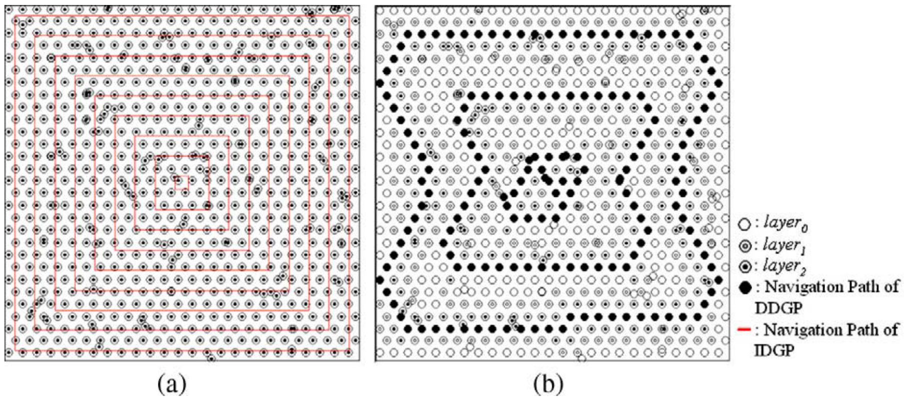


**Fig. 11.** Data gathering path of mobile robot for (a) IDGP and (b) DDGP, when $k = 3$.

shown in Fig. 12, the total data gathering path length of both IDGP and DDGP for different number of layers is shown for different number of nodes. From Fig. 12, it is noticed that the total path length in IDGP decreases with increase in value of $k$. In DDGP, the node numbers affect the total path length, though no such effect is seen in case of IDGP. Besides, it is observed that the total path length in DDGP is shorter than that of IDGP, irrespective of the value of $k$. It is due to its distributed and irregular nature of the path, based on the location of nodes in different layers. Whatever it may be, DDGP outperforms over IDGP in terms of total distance covered by a mobile robot to gather data from the nodes of the network. Hence, it is a tradeoff between the length of data gathering path and number of nodes. If a user never cares for the length of the data gathering path, we observe from our simulation results that IDGP should be considered over DDGP, as less number of nodes are required to deploy in the network. However, if a user wants to collect data more frequently such as temperature or moisture readings by the sensors, it is suggested from our analysis that DDGP is the best protocol to use.

### 6.3. Average hop counts

In order to save energy consumption of the nodes and thereby the whole network lifetime, the main goal of our protocol is to reduce the number of hops to transmit data to the sink. If the sink is located either at any corner or center of the monitoring region, the nodes have to route data through several hops, thereby consuming more energy. Hence, we have simulated the IDGP and DDGP to estimate the average number of hop counts of the nodes present in the network. Accordingly, as shown in Fig. 13, performance of our protocols is compared with the simulation results obtain by putting the sink at the corner and then at the center of the rectangular deployed region. From Fig. 13, it is observed that both IDGP and DDGP outperform in terms of average hop counts, irrespective of the position of the sink. It is to be noted that all nodes in the network use greedy relaying mechanism, in which node having largest layer value or nearest to the sink will be the relay node. Hence, the average hop counts are not affected by the node numbers.

The average hop counts in IDGP and DDGP are analyzed for different values of $k$, as shown in Figs. 14 and 15. It is observed that for fixed value of node density, IDGP outperforms over DDGP. It is due to the difference in hop distance between those two protocols. It is to be noted that the hop distance in IDGP is fixed and equal to $\frac{R_c}{2}$, whereas different hop distance is considered in DDGP. Hence, the average hop count in DDGP is affected by this hop distance as compared to IDGP. Though, the average hop counts increase with increase in the number of layers ($k$), it is not substantially affected by node densities for both of the protocols. Hence, node density in our protocols has very small impact on average hop counts. Thus, in our protocols, increasing or decreasing the node numbers of the network will not have much effect on the latency, as it has not much impact on the average hop counts.

### 6.4. Control packet overhead

In this subsection, we analyze the control packet overhead of generating the data gathering paths for both IDGP and DDGP. Accordingly, we simulated both protocols for higher number of nodes to measure the required number of control packets. To determine the control packet overhead of DDGP, each node broadcasts *Hello* message and *layer_msg* and the corresponding simula-

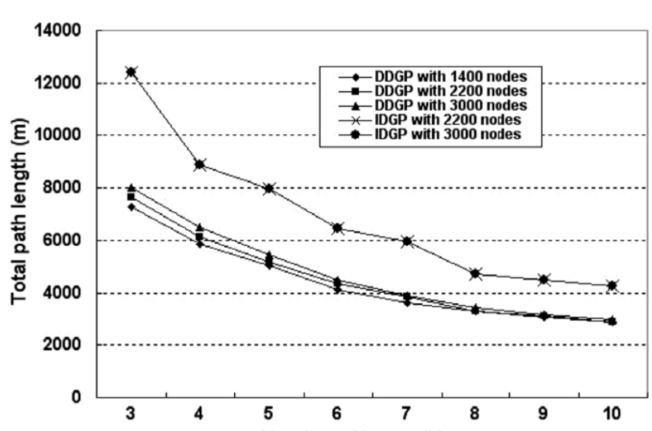


**Fig. 12.** Total data gathering path length in IDGP and DDGP for different value of layers($k$) and node numbers.

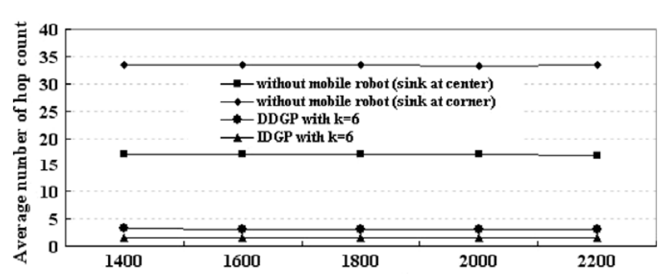


**Fig. 13.** Average hop counts in different position of sink, DDGP and IDGP with different node numbers.

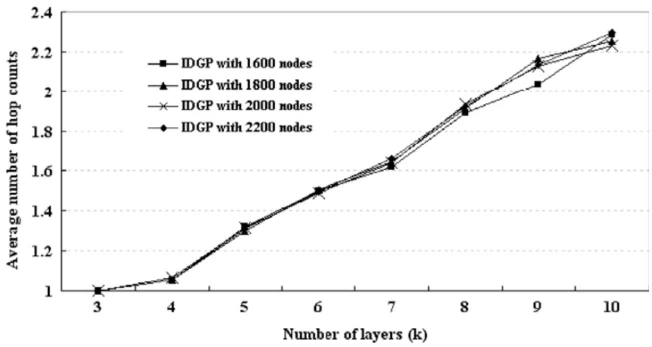


**Fig. 14.** Average hop counts in IDGP for different value of layers ($k$) and node numbers.

tion result is shown in Fig. 16. It is observed that DDGP requires more control packets than IDGP. Even if, the requirement of number of control packets increases with node numbers. It is to be noted that in IDGP, the mobile robot plans the data gathering path by itself and path information is flooded to sensors. But, in DDGP, each node does not know the data gathering path. Hence, they need the 1-hop neighbor information to determine their layer and then form the data gathering path of the mobile robot. We feel that the control packet requirements are attributed due to these characteristics and differences between the IDGP and DDGP algorithms.

### 6.5. End-to-end delay and delivery ratio

In this section we study the packet delivery ratio and end-to-end delay by delivering packets from different sensors to the mobile sink. Both scenarios are simulated for IDGP and DDGP with different layers and node numbers. As shown in Fig. 17, the end-to-end delay increases with increase in number of deployed nodes. If more number of nodes are deployed, more nodes may compete with each other to transmit packets and there is higher chance of collision and thereby end-to-end delay in transmitting packets to the mobile sink is increased. It is observed that the end-to-end delay of IDGP is less than DDGP as the data gathering path in IDGP is fixed. Due to dynamic data gathering path creation in DDGP, some packet may not be delivered on time, which may increase the ene-to-end delay. Besides, the end-to-end delay gradually decreases with decrease in the number of layers ($k$), which is obvious.

The simulation result of packet delivery ratio of the nodes to the mobile sink is shown in Fig. 18. It is observed that the successful packet delivery ratio of the nodes decreases with increase in the number of nodes. Similar to the case of end-to-end delay as shown in Fig. 17, the packet delivery ratio increases for less number of layers. It is found that IDGP outperforms DDGP in terms of delivery ratio, as the data gathering path in IDGP is fixed.

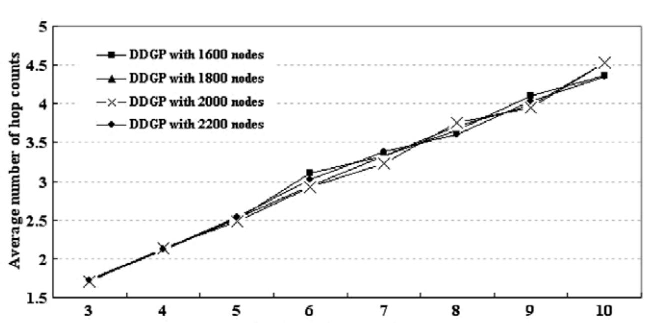


**Fig. 15.** Average hop counts in DDGP for different value of layers ($k$) and node numbers.

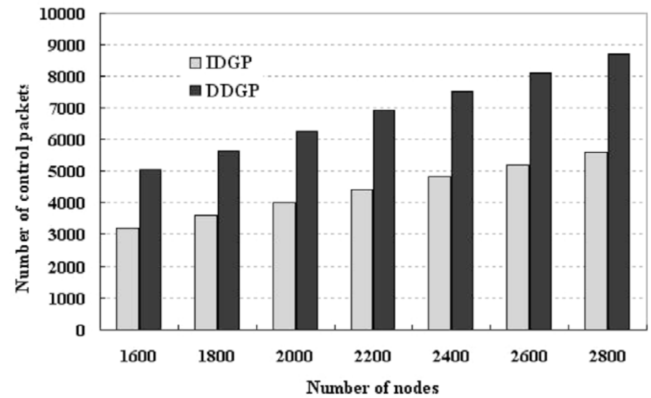


**Fig. 16.** Comparison of control packet overhead between IDGP and DDGP in 1000 m × 500 m network area.

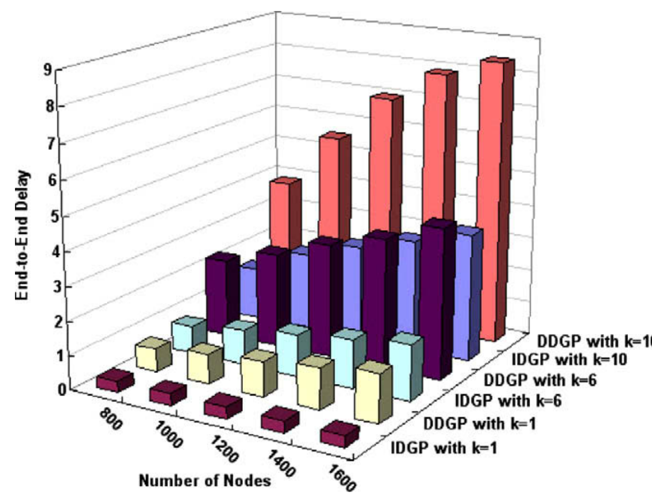


**Fig. 17.** Comparison of end-to-end delay of IDGP and DDGP for different network size and layer numbers.

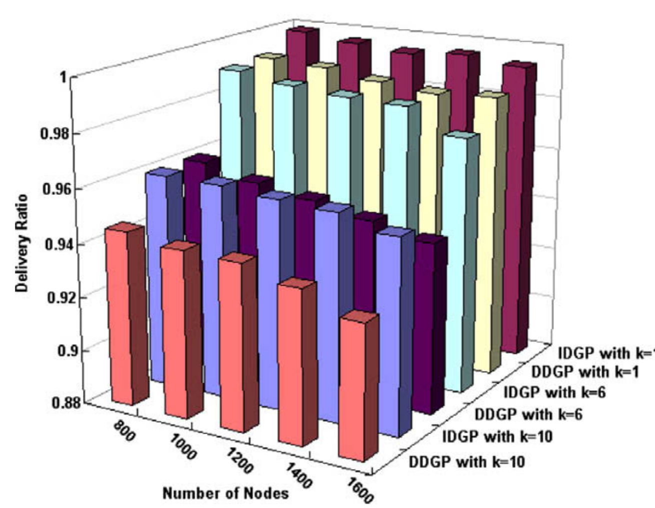


**Fig. 18.** Comparison of packet delivery ratio of IDGP and DDGP for different network size and layer numbers.

## 7. Conclusion and future work

In this paper we propose how to design the data gathering path for the mobile robot in infrastructure based or distributed network. We use a $k$-hop relay mechanism and find that the average hop counts are not affected by the node numbers, either it is IDGP or DDGP, whereas the total data gathering path length in DDGP increases with increase in the number of nodes, though it has no effect on the total path length of IDGP. Above all, our data gathering protocols enable the sensor nodes to transmit data with less number of hops and simultaneously satisfy a desired value of path length to cover the network by a mobile robot. Hence, implementation of our protocols can save energy consumption by the sensor nodes and thereby can enhance the network lifetime.

However, the performance of network in terms of end-to-end delay and packet delivery ratio may be degraded due to node redundancy and routing path loss. Hence, it is worth to design protocols to analyze the effect of link failure and thereby to study the energy consumption due to mobility of a sink. In our future work, we plan to develop algorithms for a mobile sink to study the effect of link failure for different possible routing protocols to improve the performance of data gathering.

## References

[1] H. Yang, B. Sikdar, A protocol for tracking mobile targets using sensor networks, in: Proceedings of the First IEEE International Workshop on Sensor Network Protocols and Applications, USA, 2003, pp. 71–81.

[2] J. Culpepper, L. Dung, M. Moh, Hybrid indirect transmissions (HIT) for data gathering in wireless micro sensor networks with biomedical applications, in: Proceedings of IEEE 18th Annual Workshop on Computer Communications, USA, 2003, pp. 124–133.

[3] T. He, B. Krogh, S. Krishnamurthy, J.A. Stankovic, T. Abdelzaher, L. Luo, R. Stoleru, T. Yan, L. Gu, J. Hui, Energy-efficient surveillance system using wireless sensor networks, in: Proceedings of the 2nd International Conference on Mobile Systems, Applications, and Services, USA, 2004, pp. 270–283.

[4] R. Mi, A. Chandrakasan, Energy-efficient communication for ad-hoc wireless sensor networks, in: Proceedings of the 35th Asilomar Conference on Signals, Systems and Computers, vol. 1, 2001, pp. 139–143.

[5] B. Krishnamachari, D. Estrin, S. Wicker, Modelling data-centric routing in wireless sensor networks, in: Proceedings of IEEE Infocom, 2002.

[6] R. Govindan, C. Intanagonwiwat, D. Estrin, Directed diffusion: a scalable and robust communication paradigm for sensor networks, in: Proceedings of 6th ACM/IEEE Mobicom Conference, 2000.

[7] P. Corke, S. Hrabar, R. Peterson, D. Rus, S. Saripalli, G. Sukhatme, Autonomous deployment and repair of a sensor network using an unmanned aerial vehicle, in: Proceedings of the IEEE International Conference on Robotics and Automation, USA, 2004, pp. 3602–3609.

[8] A. Galstyan, B. Krishnamachari, K. Lerman, S. Pattem, Distributed online localization in sensor networks using a moving target, in: Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks, USA, 2004, pp. 61–70.

[9] P. Cheeseman, B. Kanefsky, W.M. Taylor, Where the really hard problems are, in: Proceedings of International Joint Conference on Artificial Intelligence, 1991, pp. 331–337.

[10] E. Gonzalez, M. Alarcon, P. Aristizabal, C. Parra, BSA: A coverage algorithm, in: Proceedings of International Conference on Intelligent Robots and Systems, 2003, pp. 1679–1684.

[11] N.B. Priyantha, H. Balakrishnan, E.D. Demaine, S. Teller, Mobile-assisted localization in wireless sensor networks, in: Proceeding of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies, 2005.

[12] D.K. Goldenberg, J. Lin, A.S. Morse, B.E. Rosen, Y.R. Yang, Towards mobility as a network control primitive, in: Proceedings of the 5th ACM International Symposium on Mobile Ad hoc Networking and Computing, Japan, 2004, pp. 163–174.

[13] W. Zhao, M. Ammar, E. Zegura, A message ferrying approach for data delivery in sparse mobile ad hoc networks, in: Proceedings of the 5th ACM International Symposium on Mobile Ad Hoc Networking and Computing, Japan, 2004, pp. 187–198.

[14] A. Chakrabarti, A. Sabharwal, B. Aazhang, Using predictable observer mobility for power efficient design of sensor networks, in: Proceedings of the 2nd International Workshop on Information Processing in Sensor Networks, USA, 2003, pp. 129–145.

[15] S. Jain, R.C. Shah, W. Brunette, G. Borriello, S. Roy, Exploiting mobility for energy efficient data collection in wireless sensor networks, in: Proceedings of IEEE Workshop on Modeling and Optimization in Mobile Ad hoc and Wireless Networks, 2004.

[16] A. Kansal, A.A. Somasundara, D.D. Jea, M.B. Srivastava, D. Estrin, Intelligent fluid infrastructure for embedded networks, in: Proceedings of the 2nd International Conference on Mobile Systems, USA, 2004, pp. 111–124.

[17] Maximum Lifetime Data Gathering and aggregation in WSN <www.csee.umbc.edu/kalpakis/homepage/papers/tr-02-12.pdf>.

[18] K. Lu, L. Huang, Y. Wan, H. Xu, Energy-efficient data gathering in large wireless sensor networks, in: Proceedings of the Second International Conference on Embedded Software and Systems, 2005.

[19] P. Zeng, H. Yu, W. Liang, A new data gathering paradigm for wireless sensor networks, in: Proceedings of the IEEE, Sarnoff Symposium on Advances in Wired and Wireless Communication, 2005, pp. 85–89.

[20] S. Selvakennedy, S. Sinnappan, Yi Shang, T-ANT: a nature-inspired data gathering protocol for wireless sensor networks, Journal of Communications 1 (2) (2006).

[21] B. Ren, J. Ma, C. Chen, The hybrid mobile wireless sensor networks for data gathering, in: Proceedings of ACM IWCMC, 2006.

[22] M. Goyeneche, J. Villadangos, J.J. Astrain, M. Prieto, A. Cordoba, A distributed data gathering algorithm for wireless sensor networks with uniform architecture, in: Proceedings of 15th EUROMICRO International Conference on Parallel, Distributed and Network-Based Processing, 2007, pp. 373–380.

[23] K. Akkaya, M. Younis, M. Bangad, Sink repositioning for enhanced performance in wireless sensor networks, Computer Networks 49 (2005) 434–512.

[24] D. Niculescu, B. Nath, Ad hoc positioning system (APS) using AOA, in: Proceeding of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies, 2003, pp. 2037–2040.

[25] J. Bachrach, R. Nagpal, M. Salib, H. Shrobe, Experimental results for and theoretical analysis of a self-organizing global coordinate system for ad hoc sensor networks, Journal of Telecommunications Systems 26 (2–4) (2004) 213–233.

[26] A. Dvir, M. Segal, Placing and maintaining a core node in wireless ad hoc sensor networks, Networking (2007) 13–24.

[27] C.-L. Wang, S.-J. Syue, An efficient relay selection protocol for cooperative wireless sensor networks, in: Proceeding of the WCNC, 2009, pp. 1–5.

[28] Y. Revah, M. Segal, Improved algorithms for data-gathering time in sensor networks II: ring, tree and grid topologies, in: Proceeding of the ICNS, 2007.

[29] S.-F. Hwang, K.-H. Lu, T.-H. Chang, C.-R. Dow, Hierarchical data gathering schemes in wireless sensor networks, International Journal of Pervasive Computing and Communications 4 (3) (2008) 299–321.

[30] M. Simek, D. Komosny, P. Moravek, R. Burget, R. Silva, J. Silva, Data gathering model for wireless sensor networks based on the hierarchical aggregation algorithms for IP networks, International Journal of Computer Science and Network Security 8 (11) (2008).

[31] H. Luo, Y. Liu, S.K. Das, Routing correlated data in wireless sensor networks: a survey, IEEE Network 21 (6) (2007) 40–47.

[32] H. Gupta, V. Navda, S. Das, V. Chowdhary, Efficient gathering of correlated data in sensor networks, ACM Transactions on Sensor Networks 4 (1) (2008) 1–31.