

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800

Open access books available

122,000

International authors and editors

135M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.

For more information visit www.intechopen.com



Calculation of an Optimum Mobile Sink Path in a Wireless Sensor Network

S. Chinnappen-Rimer and G. P. Hancke

Additional information is available at the end of the chapter

<http://dx.doi.org/10.5772/50182>

1. Introduction

A wireless sensor network (WSN) is a collection of spatially distributed, resource-constrained sensor nodes, deployed within an application area, to monitor a specific event or set of events. These sensor nodes are standalone devices without access to a continuous energy source and are located either within or close to the phenomena they are observing. The nodes communicate with one or more central control point(s), generally called a sink or base station. A typical sensor node comprises a sensing unit, a small processing unit to perform simple computations, a transceiver unit to connect nodes to the network and a power unit. Some nodes are also equipped with a location finding system [1]. A WSN application contains hundreds to thousands of sensor nodes. These sensor nodes are designed for unattended operation and are generally stationary after deployment.

One of the main criteria in designing a WSN application is prolonging network lifetime and preventing connectivity degradation through aggressive energy management. There is a trade-off between a node's energy, node range, size and cost. Due to the need to conserve battery lifetime, the sensor nodes operate with low duty cycles and communicate sporadically, over short distances with low data rates. In WSNs the flow of data is predominantly unidirectional, from nodes to sink [2]. The limited resources, non-renewable power supply and short radio propagation distances, (and hence large number required for deployment), of sensor nodes impose constraints on WSN applications not found in wired networks. A WSN differs from local area networks in the following key areas [3, 4]:

- Each sensor node communicates with one or more base stations (sinks). Traffic is mainly between individual sensor nodes and a base station.
- The network topology is a multi-hop star-tree that is either flat or hierarchical.
- They are used in diverse applications which may have different requirements for QoS and reliability.

- Most network applications require dense deployment and physical collocation of nodes.
- Individual sensor nodes have limited resources in terms of processing capability, memory and power.
- Power constraints result in small message sizes
- The placement of nodes in a WSN is application dependent and may not be pre-determined.

A WSN also differs from other wireless networks, such as cellular networks and mobile ad hoc networks (MANETS) because these networks are linked to a wired or renewable energy supply. In cellular networks and MANETS, the organising, routing and mobility management tasks focus on optimizing quality of service (QoS) and ensuring high bandwidth efficiency. There is a large amount of network traffic and the data rate is high to cater for the demand for multimedia rich data. These networks are designed to provide good throughput/delay characteristics under high mobility conditions [2]. Energy consumption is of secondary importance as the battery packs can be replaced or re-charged as needed.

As the term "wireless" implies, there is no fixed physical connection between sensors to provide continuous energy and an enclosed communication medium. This creates two problems, firstly, the sensor has a finite amount of energy, which once depleted, disables the sensor and hence reduces network lifetime. Secondly, all transmitted messages will be detected by any listening device within receiving range, which then has to decide whether to accept, forward or ignore the message. This signal transmission and reception has a power cost. In addition, many WSN applications do not have a pre-planned network topology and nodes are only aware of their immediate neighbours. When routing a message to a sink, the nodes closest to the sink receive a disproportionate amount of messages, resulting in their energy being consumed earlier.

Initial message routing protocols assumed the sink or destination node was in a fixed location, and that network nodes had no or limited knowledge of the network topology [5]. An area of active research for a number of years has been how to notify the central sink (or monitoring hub) about an event in real-time by utilising the minimum amount of power of sensor nodes. Strategies to improve node energy efficiency include using multiple sinks in the application area and the use of mobile sinks to collect data from stationary sensor nodes to prevent nodes close to a sink from having their energy depleted and hence decreasing network lifetime.

A model for optimum path movement of mobile sinks to reduce the number of messages transmitted and received by an individual sensor node is proposed. An investigation is conducted into the optimum route a mobile sink can travel that will reduce the number of messages transmitted within a network, allow equitable usage of all nodes to transfer an event message and still allow an event to be reported in real-time.

In the following sections a brief discussion of the use of mobile elements in WSNs as well as current research using mobile sinks and/or nodes to improve the energy efficiency of routing protocols is provided. The algorithm to transmit data from a sensor node to a mobile sink is discussed and the results analysed.

2. Mobile entities

The application and routing challenges presented by static nodes in a dense, multi-hop WSN has led to the investigation of the use of mobile elements in WSNs for data collection and/or dissemination. The advantages of using mobile entities in WSNs include [6, 7]:

1. Improved reliability as there is less contention and collisions within the wireless medium because data can now be collected directly through single or limited hop transmissions.
2. Reduced reliance on nodes located close to a static sink to route messages to the sink, resulting in increased energy efficiency and network lifetime.
3. Improved connectivity as mobile nodes can enable the retrieval of collected measurements from isolated regions of the sensor application area.
4. Sparse network architecture implies reduced application cost as fewer nodes are required and nodes can utilise mobile elements already present in the application area such as trains, cars, wildlife, and livestock etc.

The use of mobility in WSNs introduces complications not found in static WSN applications, such as detecting when nodes are within transmission range of a mobile sink, ensuring reliable data transfer as nodes may move as messages are exchanged, tracking sink location and design of a virtual backbone to store data reports so that the mobile sink can easily collect them, and managing sensor nodes to support sink mobility [6, 7].

Current strategies for data collection and dissemination using mobile elements include a rendezvous-based virtual infrastructure which uses limited and unlimited multi-hop relays to route data messages, or a backbone-based approach where mobile sinks only communicate with pre-defined cluster heads or gateways, or passive data collection where there is direct communication between the source and sink [7, 8].

The mobility patterns of mobile elements (sinks and relays) are dependent on the type of WSN application, its data collection requirements and the controllability of the mobile elements. Current mobility patterns can be classified into the following categories [6, 8]:

1. *Random mobility*: no network information required because communication does not occur regularly but with a distribution probability. This method does not provide optimal increases in network lifetime due to the need for continuous sink position updates and route reconstruction.
2. *Predictable or deterministic mobility*: mobile elements enter range of sensor nodes at regular, periodic times to collect data, and allow the sensor nodes to predict arrival of mobile entities.
3. *Controlled mobility*: the mobile elements movements are not predictable but are controlled by network parameters such as maximum and minimum residual energy of sensor nodes on a data route, event location, and the mobile elements trajectory and speed. In addition, the mobile entities can be instructed to visit individual nodes at specific times, and stop at nodes until they have collected all buffered data.

3. Related work

According to Akkaya, Younis and Bangad [9], finding an optimal location for the sink in a multi-hop network is a complex problem, NP hard in nature. The complexity results mainly from two factors. The first factor is the potentially infinite possible positions that the gateway can be moved to. Second, for every interim solution considered during the search for an optimal location, a new multi-hop network topology needs to be established in order to qualify that interim solution in comparison to the current or previously picked location in the search. A mathematical formulation of the problem would involve a huge number of parameters including the positions of all deployed sensors, their state information such as energy level, transmission range, etc., and the sources of data in the networks. The authors propose moving the sink to the top relay nodes location. The sink is assumed to know the geographical location of deployed sensors. In the solution proposed in this article, an optimum location is not sought but an optimum path for a mobile sink that will ensure equitable usage of all nodes to transport data messages to the mobile sink node.

Research undertaken by Somasundara et. al [10] shows that the energy consumption in a network using a mobile base station is significantly less than that of a static network. The authors propose moving the base station around the application area. When the base station is within range of sensor nodes, it collects event data. This is not an optimum real-time solution as the sensor nodes have to wait for the base station to arrive before transmitting event information but is feasible in delay-tolerant applications such as environmental monitoring. A key difference between this researcher's proposed ideas and the model presented here is that in the model presented here an optimum path within the application area, along which one or more mobile sinks travel is calculated.

Huang, Zhai and Fang [11] consider a wireless network where the sensors are mobile, (applications such as tracking free-ranging animals, both wild or farm livestock). The problem focused on in this paper is on improving the robustness of routing when there are path breakages in the communication channel due to node mobility. The suggested solution is the use of a cooperative, distributed routing protocol to combat path breakages. The writers assume that the intended path or route between the source and destination is already known and neighbouring nodes can be used if the communication channel on the intended path fails. Our primary research focus in this paper is the calculation of an optimum path for a mobile sink to reduce the number of messages required to be re-transmitted when sending a message to a sink in the WSN. However, we will have to take cognisance of possible path breakages that may occur during the development of optimal routes.

Vupputuri, Rachuri and Ram Murthy [12] use mobile data collectors to achieve energy efficient and reliable data communication. When an event occurs, sensor nodes inform the nearest data collector. The data collector aggregates the event information and with a specified reliability factor (R) informs the base station. The primary focus of the authors' investigation is determining a mobile strategy for the data collectors to ensure reliable and energy efficient event reporting. The mobility strategy does not consider how to optimise the changing locations of the data collectors. The authors focus on reducing the number of

messages sent and received by nodes close to the base station to improve network lifetime as well as ensuring that multiple paths are used to improve network reliability.

Gu, Bozdag and Brewer [13] use a partitioning-based algorithm to schedule the movements of mobile sinks in order to reduce data loss due to buffer overflow while waiting for a sink to arrive. This aspect is ignored in our proposed solution. Other recent research activity in this field, include the work of Marta and Cardei [14] where mobile sinks change their location when the nearby sensors' energy becomes low, and determines the new location by searching for zones where sensors have more energy. Heinzelman, Chandrakasan and Balakrishnan [15] have proposed a combination hierarchical and cluster based scheme that groups sensors and appoints a cluster head to transmit messages to the sink, thus saving the surrounding nodes energy (LEACH). The small percentage of cluster heads are randomly re-selected to improve node longevity of nodes located close to cluster heads. Patel, Venkatesan and Chandrasekaran [16] propose a Lexicographic Maximum Lifetime Vector routing scheme to maximise the first, second and so forth set of nodes time until their battery energies are depleted.

The use of a mobile relay to route all traffic passing through a static node for a specified period of time, is discussed by Wang et. al. The mobile relay traverses a concentric circle that stays within a two-hop radius of the sink. The authors show that the use of a mobile relay can improve a WSN's lifetime by 130%. Additional experiments show that a mobile sink, moving around the perimeter of a large and dense network, can best optimise WSN lifetime compared to a mobile relay or using resource rich static relays located close to a static sink [17]. The results of this paper indicate that the mobile relay should be a maximum of two hops from a static sink and that only nodes within a maximum of 22 hops from the sink need to be aware of the location of the mobile relay. The use of both a mobile sink and a mobile relay prevent over-utilisation of static nodes located close to the sink to route messages to the sink and hence increase overall WSN lifetime. We do not consider the use of a mobile relay in the solution discussed in this chapter and focus exclusively on an optimum path for a mobile sink to follow within a WSN application area.

A multi-sink heuristic algorithm (HOP) is proposed by Ben Saad and Tourancheau to find the best way to move mobile sinks in order to improve the lifetime of large scale sensor networks. Sinks are relocated to nodes located the maximum number of hops from a sink as it is assumed that these node will have higher residual energy as the nodes will not be required to re-transmit messages destined for a sink [18]. The minimum amount of time a sink will spend at a specific location is 30 days. The proposed algorithm is compared against schemes using static sinks, sinks moving along the periphery of the network, sinks moving randomly and sinks moving according to an Integer Linear Programming algorithm, in terms of network lifetime and residual energy at each sensor node. The results of simulations indicate the HOP algorithm achieves significant improvement in network lifetime over the other algorithms and that there is more even distribution of residual energy per sensor node. The HOP algorithm differs from the solution proposed in this paper, because HOP assumes that the sinks are not continuously mobile but are moved after a specified number of days to different locations within the building.

4. Algorithm design

To reduce the number of messages received and re-transmitted by nodes closest to the sink, it is proposed that one or more mobile sinks follow a path in the application area based on the calculated number of hops from a sink. The path should (1) ensure reliable communication between nodes and sink(s), (2) ensure the even distribution of messages received and transmitted within the application area to reach a sink destination, and (3) enable real-time processing of event messages. Consider the following definitions in Table 1:

| Variable | Description |
|-------------|--|
| X | Width of application area |
| Y | Length of application area |
| R | Node and mobile sink communication range |
| X_b | Minimum starting X point on the mobile path |
| X_e | Maximum ending X point on the mobile path |
| Y_b | Minimum starting Y point on the mobile path |
| Y_e | Maximum ending Y point on the mobile path |
| H | Number of hops to nearest node within communication range of the mobile sinks path. |
| d | Distance between each sink broadcast "hello" message. |
| N_{hello} | Number of times the "hello" message is broadcast to complete one loop around the calculated path |
| a | The constant acceleration of the mobile sink |
| d_{stop} | The constant deceleration of the mobile sink |
| v_i | Initial velocity of mobile sink |
| v_f | Final constant velocity of mobile sink |
| s_{av} | Distance the mobile sink has to traverse after accelerating from zero velocity to reach required velocity. |
| s_{dv} | Distance the sink has to traverse after decelerating from constant velocity to when the sink stops (zero velocity) |
| s_{cv} | Distance the sink has to traverse moving at constant velocity before next "hello" type message is broadcast |
| t_{av} | Time sink to accelerate from zero to constant velocity |
| t_{dv} | Time for sink to decelerate from constant velocity to zero |
| t_{cv} | Time for sink moving at constant velocity to traverse the required distance before next "hello" message is broadcast |
| T_{total} | Total time sink takes to complete one loop of its calculated path transmitting messages at required intervals |
| t_{stop} | Time a mobile sink will stop, broadcast a "hello" type message and wait for responses from surrounding nodes |

Table 1. Definitions of variables used in calculations of mobile sink path

4.1. Calculation of optimum path for one mobile sink

For one sink, the optimum path must be equidistant from any furthest node in the application area. The maximum distance a message from a node on the perimeter of the application area travels before reaching a node within communication range of the mobile sink must be the same as the maximum distance from a node at the centre of the application area to a node within communication range of the mobile sink. If one sink is located in the centre of the application area, then for a square or rectangle shaped area, the number of hops a message from a node at the farthest end of a square or rectangular application area has to travel to reach the sink is approximately:

$$H_{sr_num} = \left(\frac{X}{2 * R}\right) + \left(\frac{Y}{2 * R}\right). \quad (1)$$

For a circle shaped application area, nodes at the perimeter are distance r (where r is the radius) from the centre. Thus the number of hops for nodes on the perimeter is:

$$H_{circle_num} = \frac{r}{R}. \quad (2)$$

To ensure equidistance between nodes at the centre of the application area and nodes at the perimeter of the application area, the number of hops should be almost the same, i.e. $\frac{H}{2}$.

Since the application area dimensions (X and Y for square and rectangular shapes or r for circular shapes), and the range of the nodes (R) is known, the maximum number of hops a message $\left(\frac{H}{2}\right)$ has to be re-transmitted before reaching a node that is within communication range of the mobile sink's path can be calculated as follows:

Square or rectangular shape

$$H_{sr} = \left(\frac{X}{4 * R}\right) + \left(\frac{Y}{4 * R}\right) \quad (3)$$

Circular shape

$$H_{circle} = \frac{r}{2 * R} \quad (4)$$

where, $H_{(outer)} = low(H) = floor(H)$

and, $H_{(inner)} = high(H) = ceil(H)$

Once the number of hops has been calculated, the optimum path of a mobile sink can be calculated as shown below:

Calculation of optimum path:

Square or rectangular shape:

$$X_b = R * \left(\frac{H_{sr}}{2}\right) \quad (5)$$

$$X_e = X - \left(R * \left(\frac{H_{sr}}{2}\right)\right) \quad (6)$$

$$Y_b = R * \left(\frac{H_{sr}}{2}\right) \tag{7}$$

$$Y_e = Y - \left(R * \left(\frac{H_{sr}}{2}\right)\right) \tag{8}$$

Circular shape:

$$r_{path} = H_{circle} * R \tag{9}$$

Consider the nodes placed in a 300mx300m WSN as shown in Figure 1. The nodes' range is assumed to be 30m and thus each node is placed 30m from the previous node. The optimum path for the mobile sink calculated based on Equations [5], [6], [7] and [8]. Nodes within the immediate communication range of the mobile sink node act as temporary stores for any message destined for the sink. As the sink passes along the path, these nodes pass the message to the sink. This results in a short delay between the time an event occurs and the time the sink receives the message. If the sink needs to be notified immediately, the node can calculate where in the mobile path the sink currently is and re-route the message to the sink.

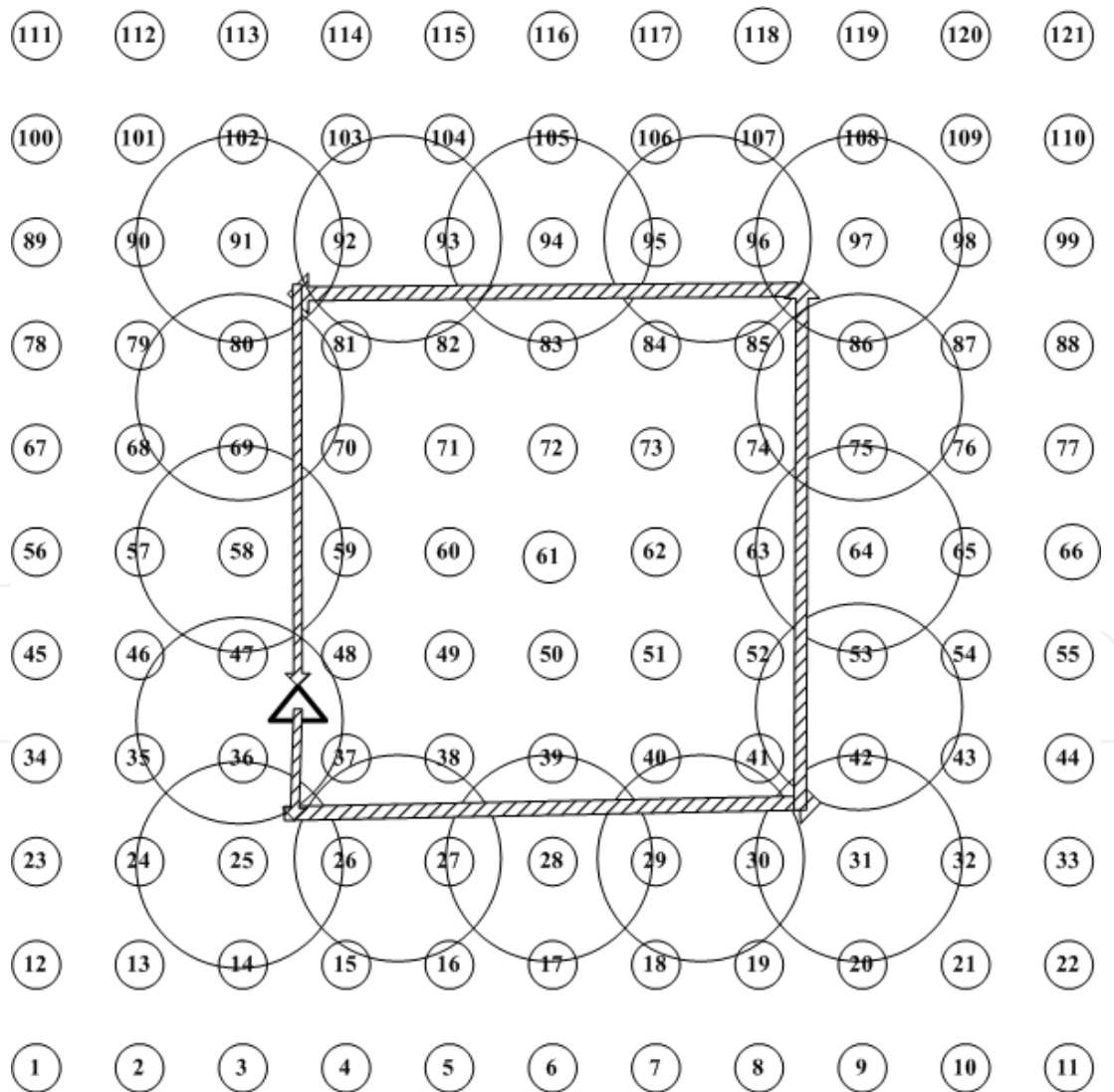


Figure 1. Path for a mobile node to follow in a 300mx300m application area.

4.2. Optimum path for multiple mobile sinks

For multiple sinks, the WSN application area should be sub-divided optimally. Thus the number of sinks and number of squares must be a square of a positive integer number, i.e. $N_{sink} = \{1^2, 2^2, 3^2, 4^2 \dots\}$. The size of each square is calculated as follows:

$$x = \frac{X}{\sqrt{N_{sink}}} \text{ and } y = \frac{Y}{\sqrt{N_{sink}}} \tag{10}$$

Using Equations [5], [6], [7], [8] and [10] the mobile path for each sink can be calculated. Figure 2 shows the number of sub-divisions and mobile paths calculated for the same 300mx300m WSN application area shown in Figure 1 for one mobile sink. The size of the application area is small, so for four square sub-divisions, each node in the WSN application area will be within communication range of the mobile path.

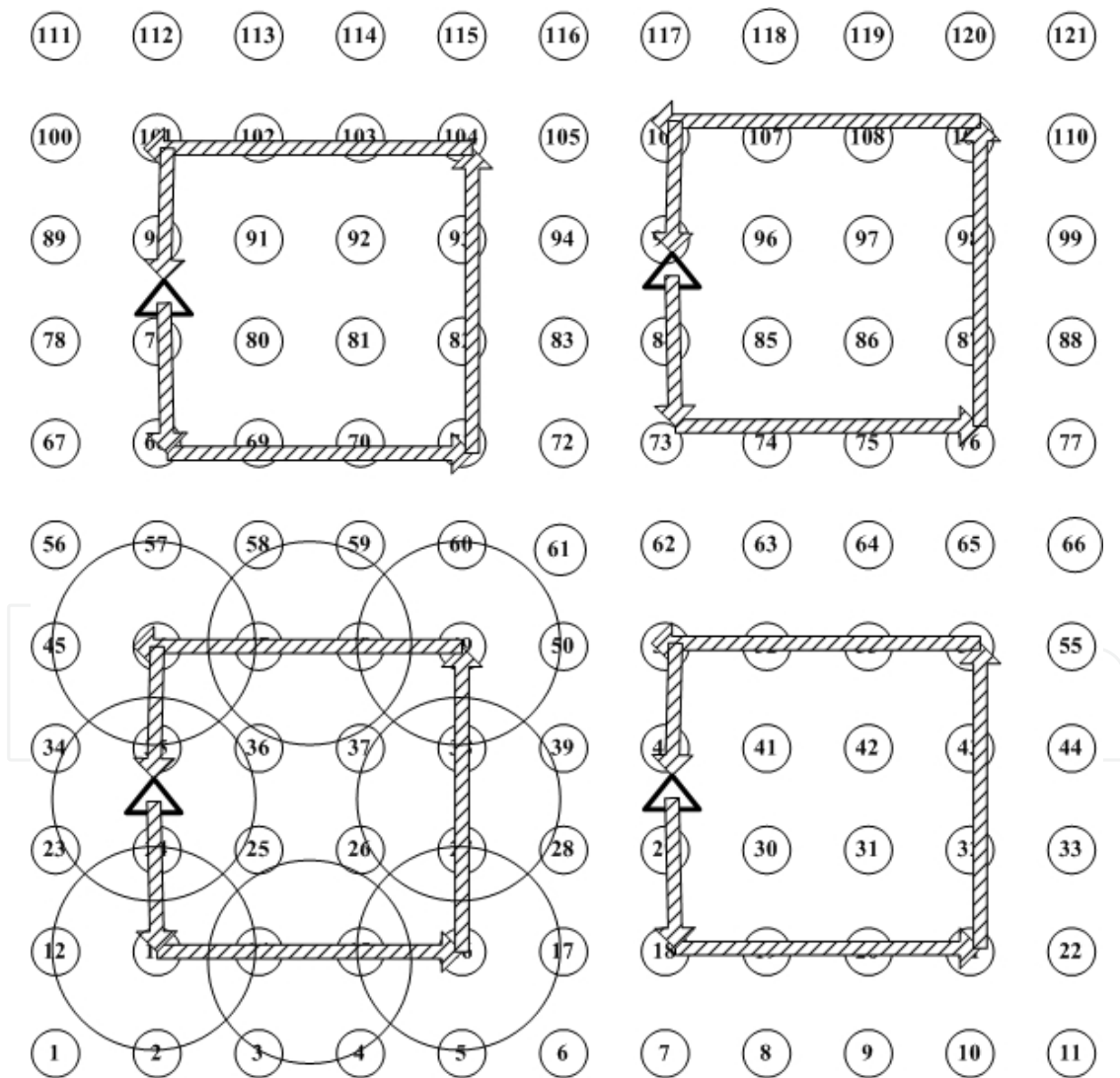


Figure 2. Four mobile sinks and each sinks path in a 300mx300m application area.

If there are multiple sinks, then the actual load is spread among more nodes. In Figure 2, the number of sinks and the optimum mobile path can be calculated to ensure that all nodes are within communication distance of a mobile sink's path, with the possible exception of nodes at the perimeter of the WSN application area. For example, nodes 1, 6, 11, 56, 61, 66, 111, 116 and 121 may require an intermediate node to pass the message on in Figure 2. To ensure connectivity, this set of nodes can be moved closer to the sink node's path, as shown in Figure 3. The path each sink has to travel is even shorter and hence the calculated time to complete one loop is less.

When an event occurs, the sensing nodes aggregate the data and elect a single node to forward the message to the sink. In Figure 2, as each node is one hop from the path of the mobile sink, the message will be stored by the elected node until the sink passes by and requests messages. In Figure 1, the message is stored by any node in direct communication range of the mobile sink as it moves along the path. Most nodes in the WSN application area of Figure 1 are two hops away from the path of the mobile sink. Nodes at each corner are at most three hops from the path of the mobile sink because it is assumed that the corner nodes are moved slightly into the application area as shown in Figure 3 to be within communication range of at least three nodes.

Only nodes which have a minimum of four immediate neighbours will re-transmit the event message. This ensures that nodes on the perimeter of the application area do not unnecessarily re-transmit the message. The event message is only re-broadcast until it is received by an intermediate node that is in direct communication range of the path of the mobile node. The message is stored and when the mobile node passes the intermediate node, all stored messages are transmitted to the mobile sink. Real-time event messages can be forwarded to nodes that will be closer to the sink's path based on the calculations described at the end of this section.

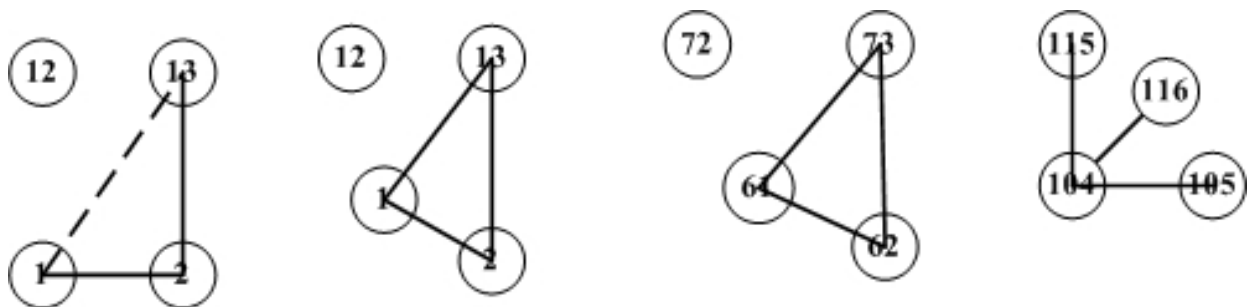


Figure 3. Moving corner nodes within communication range of mobile sink path

4.3. Calculation of distance between each “hello” broadcast message from mobile sink

To ensure complete coverage of all nodes neighbouring the path, the calculation of the distance between transmitting a “hello” broadcast message and waiting for responses from surrounding nodes is shown below:

$$d = R + \frac{R}{2}$$

$$d = \frac{3R}{2} \quad (11)$$

The number of times the sink stops and broadcasts a “hello” type message is given by the following formula:

$$N_{hello} = \frac{2*(X_e - X_b) + 2*(Y_e - Y_b)}{d} \quad (12)$$

4.4. Time for a mobile sink to complete one loop around the path

The mobile sink first moves along the path and greets all nodes within communication range. The mobile sink transmits a “hello” greeting message at every d metres requesting any of the surrounding nodes to return any data messages they may have temporarily stored while waiting for the sink to return. The message contains the mobile sink’s ID, velocity and acceleration, sink direction, its intended path, and when it calculates it will return to its current position as well as a list of all nodes that have responded to its greeting thus far. Initially, during the first loop of the mobile sink, the path list will be incomplete, as the sink is not yet aware of all nodes in its path range. When the mobile sink completes its first loop, it will have obtained a reasonably accurate network topology of all nodes within communication range of its path and their locations. The mobile node will re-broadcast this list as it continues to loop around its path, so that even if some nodes were asleep during previous cycles, these nodes can still obtain the list to update their records.

In the event that a real-time event message needs to be reported to the mobile sink, the initial node that is elected to receive the event message, as it is within communication range of the sink or the actual node that detected the event, can transmit the message to the sink, using this list and its knowledge of the mobile sink's velocity and intended path, to determine the optimum nodes to use to route the message to the sink. The messages transmitted between the nodes will travel faster than the mobile sink so the message will be delivered to the sink faster than waiting for the sink to pass by again.

4.5. Calculation of total time it takes a sink to complete one loop across the mobile path

4.5.1. Sink stop-start movement with non-uniform velocity

Initially the sink will have to move from a state of rest or initial velocity of zero to a constant, specified velocity. The time it takes a mobile sink to accelerate to a constant velocity can be calculated using the following equation:

$$a = \frac{v_f - v_i}{t_{av}}$$

The initial and final speed as well as the acceleration of the mobile sink is known. Thus, the time it takes the mobile sink to accelerate from zero and reach constant velocity is:

$$t_{av} = \frac{v_f - v_i}{a} \quad (13)$$

The distance the mobile sink has to traverse after accelerating from zero velocity to when the sink reaches the required constant velocity is:

$$s_{av} = \frac{1}{2} a t_{av}^2 \quad (14)$$

The sink will have to decelerate to stop before it broadcasts another “hello” type message. The time it takes a mobile sink to decelerate to a stop is similar to equation [13], i.e.

$$t_{dv} = \frac{v_{zero} - v_f}{a_{stop}} \quad (15)$$

The distance the mobile sink has to traverse after decelerating from constant velocity to zero velocity, i.e. to when the sink comes to a standstill, can be calculated as follows:

$$s_{dv} = \frac{1}{2} a t_{dv}^2 \quad (16)$$

Now, the time the mobile sink will spend at constant velocity can be calculated based on the distance between each time the sink node broadcasts a “hello” type message:

$$s_{cv} = d - (s_{av} + s_{dv}) = v_f t + \frac{1}{2} a t_{cv}^2$$

Since at constant velocity, $a=0$,

$$t_{cv} = \frac{s_{cv}}{v_f} \quad (17)$$

If t_{stop} is the time a mobile sink will stop, broadcast a “hello” type message and wait for responses from surrounding nodes, the total time for a node to complete one loop along the calculated path is given by the following formula:

$$T_{total} = N_{hello} * (t_{stop} + t_{av} + t_{cv} + t_{dv}) \quad (18)$$

Each node within communication range of the mobile sink’s path must be able to perform the above calculations. When one of these nodes receives an event message that it has to re-transmit to the mobile sink, it can calculate the time delay before the sink will again pass by, based on the above equations. The node can then determine, based on the message status and urgency, whether to wait for the mobile sink to pass within communication range or whether to route the message to a node closer to the mobile sink. The sink path information contained in the previous “hello” message is used to determine which node to request to forward the message to the sink. As electromagnetic waves travel much faster than the mobile sink, this will ensure that the event message reaches the sink in real-time.

4.5.2. Sink movement with uniform velocity

The previous calculations are based on the mobile sink stopping before it broadcasts a “hello” type message. The stopping and re-starting by the mobile sink will increase the time it takes a mobile sink to complete a loop around the calculated mobile path. A variation on

the above calculations is to assume that the mobile sink moves at constant velocity without stopping. When the mobile sink reaches a “hello” broadcast point it will transmit a “hello” type message to all nodes and continue moving at constant velocity. Because electromagnetic waves travel much faster than the mobile sink, the mobile sink should be able to send and receive all responses from surrounding nodes before it moves out of radio range. Then Equation [18] becomes:

$$T_{total} = \frac{2*(X_e - X_b) + 2*(Y_e - Y_b)}{v_f} \quad (19)$$

Of course the mobile node will have to decelerate when it approaches a corner to turn, but within the experimental simulation it is assumed that this time to turn is negligible.

5. Experimental simulation

The experimental setup used the Network Simulator (NS-2). In NS-2 the mobile nodes move at constant velocity. As this was a simulation environment, the mobile node did not require time to accelerate to a constant final velocity or to decelerate when turning a corner. Therefore, the time calculations are based on the node moving at constant velocity around a square path.

Changes were made to certain C++ programs in the NS-2.3.5 version to enable the node to move along the specified path and periodically send “hello” type messages. A Tcl script defined the parameters of the path the node travelled on and stored the event messages received by nodes along the mobile node’s path. When a mobile node passed by a node with stored event messages, the node would pass these messages onto the mobile sink.

Experiments were run to determine the time it takes to complete one loop around the calculated path as shown in Figure 1. This time was verified with the calculated time, using Equation (15). Thereafter an event message was broadcast from a node on the perimeter of the application area, and the effect on surrounding nodes was analysed. The velocity of the mobile sink was set at 10 m/s. The range of the nodes was assumed to be 30m.

Using Equation [3], the number of hops was calculated to be:

$$\begin{aligned} H_{sr} &= \left(\frac{300}{(4 * 30)} \right) + \left(\frac{300}{(4 * 30)} \right) \\ &= \frac{20}{4} \\ &= 5 \text{ hops} \end{aligned}$$

Using Equations [5], [6], [7] and [8] the path can be calculated as follows:

$$\begin{aligned} Y_b = X_b &= R * \left(\frac{H_{sr}}{2} \right) \\ \text{(use the integer value, i.e. floor())} &= 30 * \left(\frac{5}{2} \right) \\ &= 60 \end{aligned}$$

$$Y_e = X_e = 300 - \left(R * \left(\frac{H_{sr}}{2} \right) \right)$$

(use the integer value, i.e floor()) $\quad = 300 - \left(30 * \left(\frac{5}{2} \right) \right)$

$$= 240$$

The corner node (node 1 from Figure 1) is moved slightly into the application area (i.e. x = 10 and y = 10) to enable a message from node 1 to reach a node with at least 4 neighbours (in this case node 13). Figure 4 shows the experimental network topology for a 300 by 300 application area with a node range of approximately 30m.

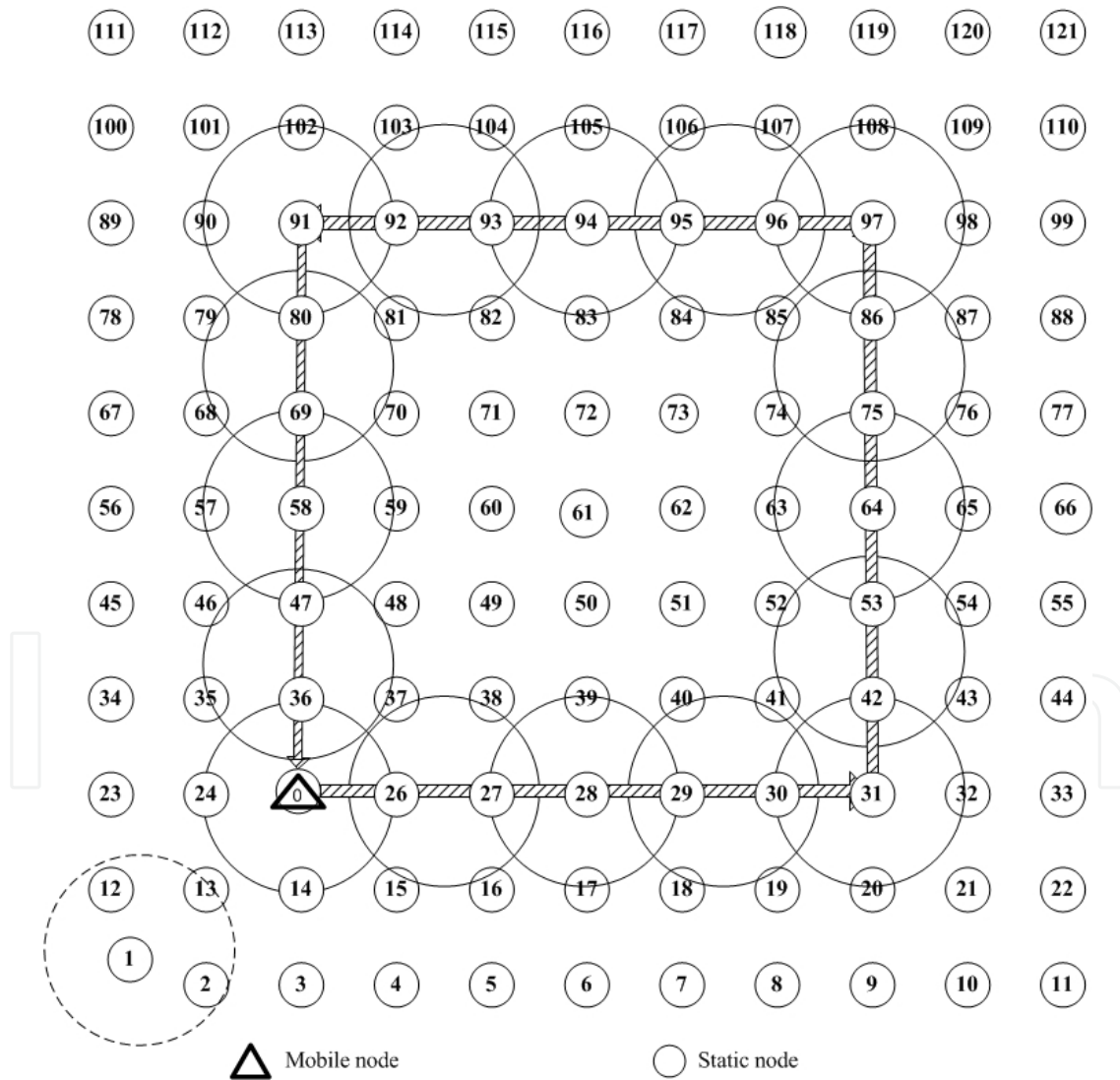


Figure 4. Experimental Setup with node 1 moved slightly into the application area.

6. Results and analysis

Using Equation [19] the time it will take a mobile node moving at a constant velocity of 10m/s to complete one loop along the calculated path is calculated.

$$T_{total} = \frac{2 * (240 - 60) + 2 * (240 - 60)}{10}$$

$$T_{total} = 72 \text{ seconds}$$

The NS-2 Tcl script was run and the time taken for the mobile node to complete one loop around the calculated path as shown in Figure 1 is **72 seconds**.

An analysis of the number of messages received by nodes neighbouring the mobile node's path is shown in Figure 5. As can be seen, certain nodes receive more than one message. These nodes are on the perimeter of two intersecting "hello" type messages sent from the mobile sink as shown in Figure 1. Thus nodes 36, 37, 26, 27 etc. receive a "hello" message from the mobile node twice. To prevent this duplication of received messages from the mobile node, the researcher suggests that the neighbouring nodes go into sleep mode for a specified time period after receiving the first "hello" type message from the mobile node. This should ensure that all nodes neighbouring the mobile node's path only receive one message per complete loop of the circuit.

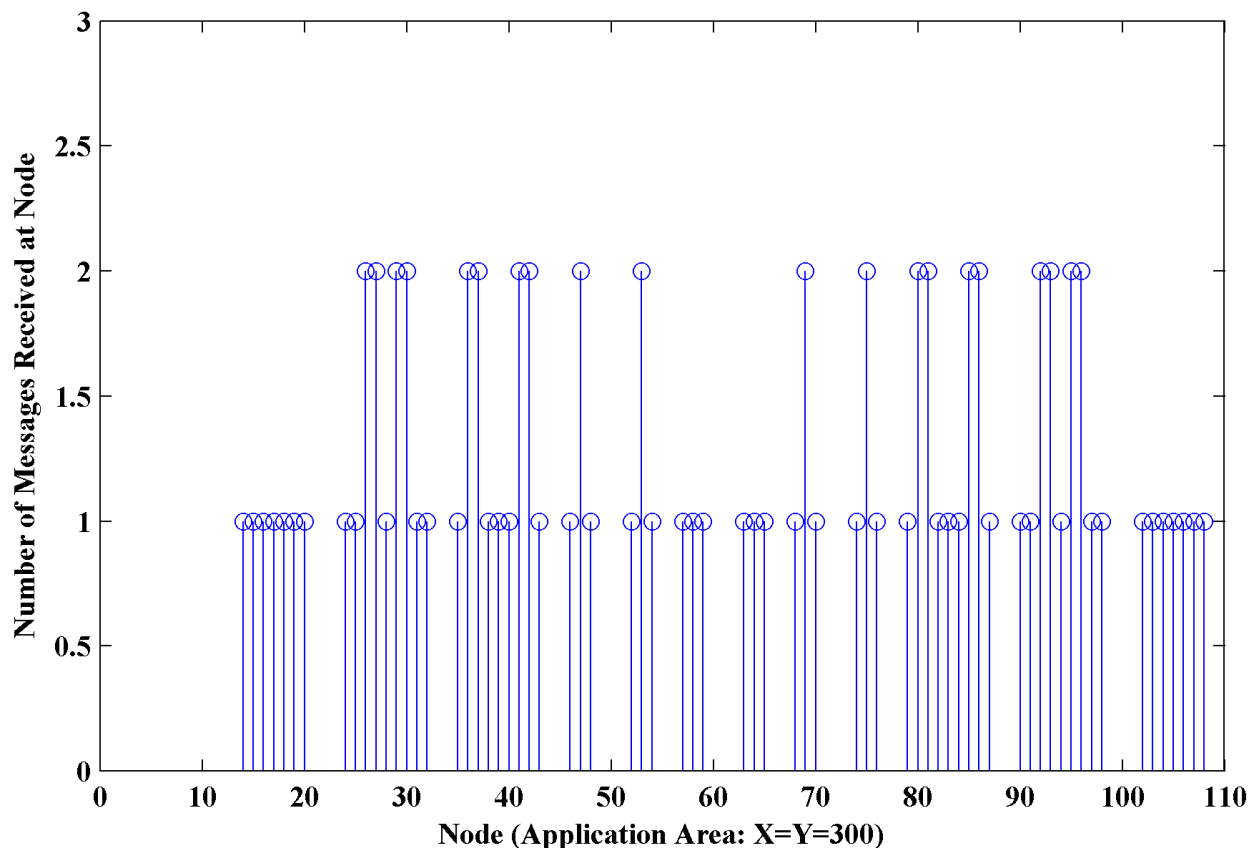


Figure 5. Number of messages received by nodes neighbouring mobile node's path

To conserve energy further, the number of times a mobile node will circumvent the path can be application-specific. For example, if sensor nodes are required to send updates to a sink periodically, the mobile node can traverse the path only during this time period. However, if the application requires the mobile node to monitor the area continuously for events and respond in real-time, the mobile sink has to move along the path and send “hello” type messages continuously.

However, the continuous sending of “hello” type messages at periodic intervals by the mobile node, does incur a cost. To reduce the number of messages transmitted within the application area further, (depending on the type of WSN application); a “hello” type message can be sent once at initialisation when the mobile sink first completes a loop along the path. All nodes along the perimeter will be able to calculate when the sink will pass by again and ensure that the node is awake during that time, if the node has event messages to relay. At the calculated time the perimeter node can proactively send a message to the mobile node informing it that it will begin transmitting event messages.

To determine the effect of the mobile node on reducing the total number of messages within the application area and received per individual node a series of tests were run for a message destined to the mobile node 0 (Figure 4) for both flooding and the mobile algorithms with various hop counts.

The effect of sending a message from a node on the perimeter of the application area to one of the nodes on the perimeter of the mobile node’s path is analysed. For example, a message is sent from node 1 to nodes 14 and 24 (refer to Figure 4). As can be seen in Figure 6, only those nodes used to pass the event message receive more messages than those shown in Figure 5.

To further reduce the total number of messages sent and received when reporting an event message, the mobile node algorithm only allows nodes with four or more neighbours to re-broadcast the received event message. When a node that is within range of a mobile node receives a message (in this experiment, node 14 and node 24), it stores the message for collection by the mobile node 0. For flooding all nodes re-broadcast the message. The event message was sent at 3 seconds and the TCL script was set to end at 10 seconds, i.e., before the mobile node completes one full cycle around its predetermined path. The experiment was to compare flooding and using the mobile route algorithm in terms of the time to reach the destination and the total number of messages transmitted within the network as well as the number of messages received per node. Thus, the worst case scenario of assuming that the mobile node has just left the range of node 24 and node 14 and started on the path and will only return in approximately the time it takes to complete one full circumnavigation of the specified path is considered.

Figure 7 shows the number of messages received by node 14 and node 24 for the mobile algorithm and flooding. Node 14 and node 24 receive the same number of messages (2) for the mobile algorithm. In flooding the number of messages per node varies widely but node 14 tracks node 24 in terms of the number of messages received per hop count. In the mobile algorithm, node 14 and node 24 receive **two** messages, one from node 13 and one from the mobile node 0. As can be seen even for low hop counts, the number of messages when using flooding exceeds the number of messages for the mobile algorithm.

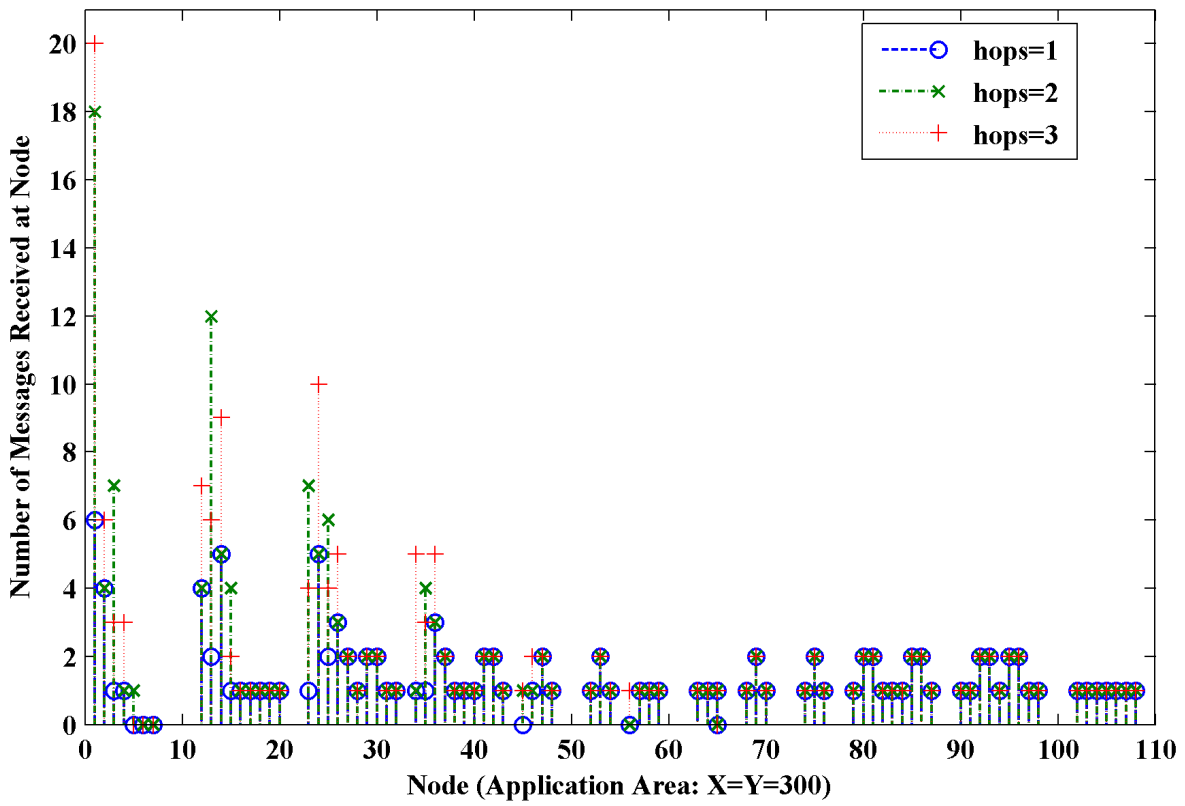


Figure 6. Number of messages per node when event message sent to node on mobile nodes perimeter

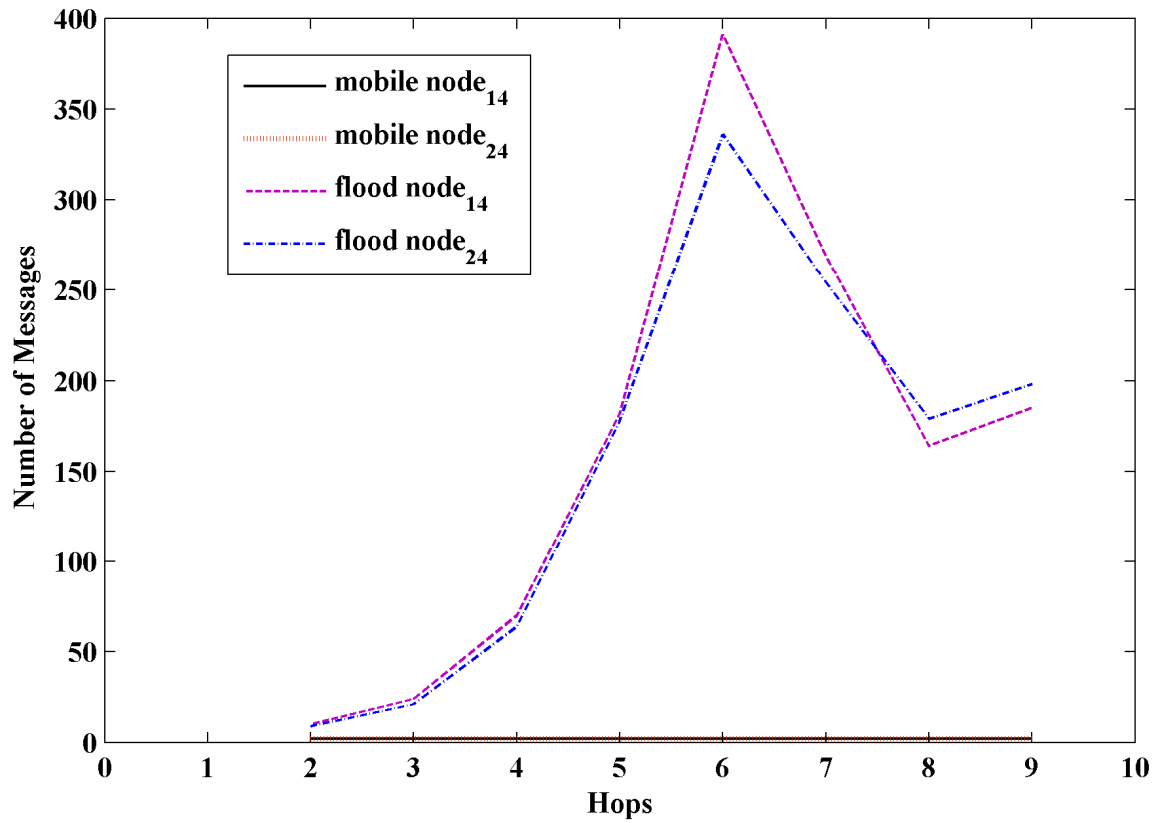


Figure 7. Number of messages for node 14 and node 24 for varying hop count

Figure 8 shows the number of messages per individual node when only nodes whose neighbours are equal to or greater than 4 re-transmit an event message. When compared to the number of messages received per node in Figure 6, it is obvious that by restricting which nodes re-transmit an event message, there is a significant decrease in the number of messages received or re-transmitted amongst individual nodes. Once the event message reaches a node that can convey the event message directly to the mobile node (in this case node 14 and node 24), the algorithm stops retransmitting the event message.

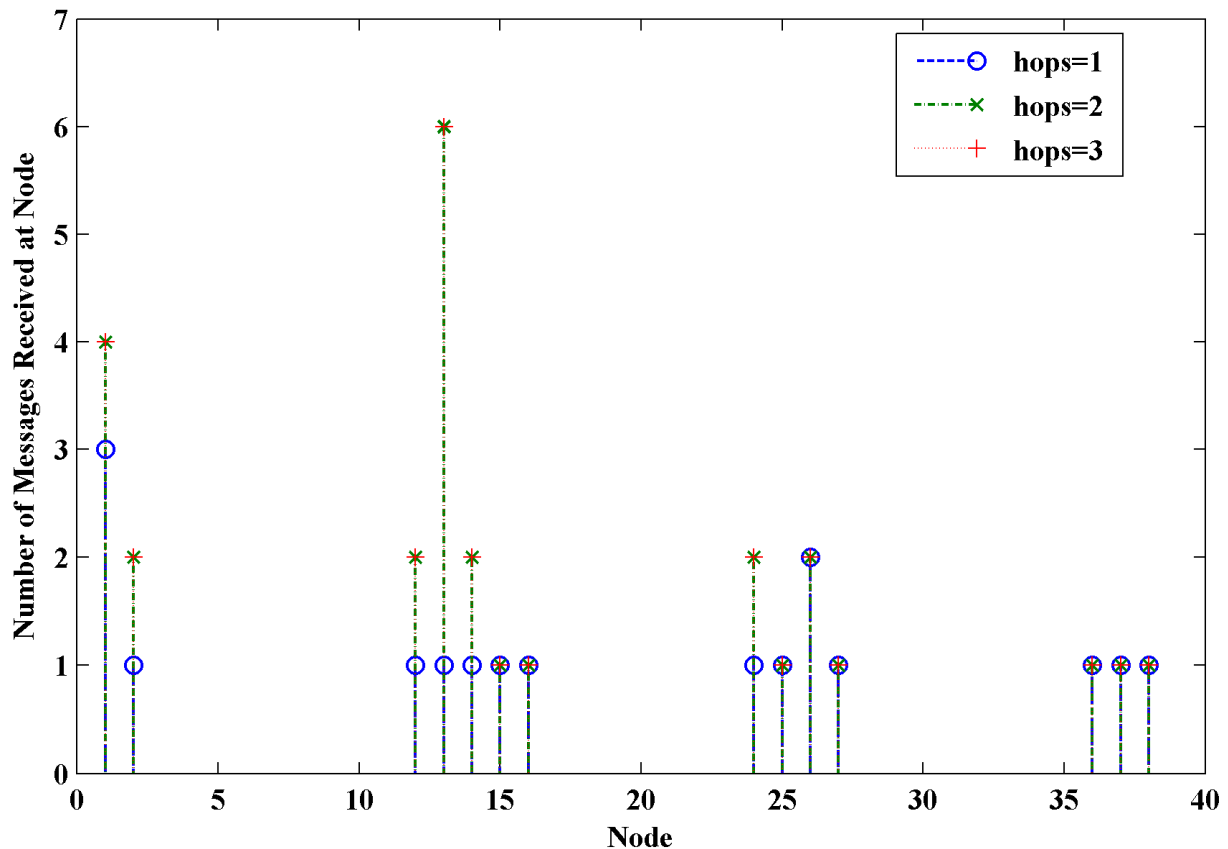


Figure 8. Restricting re-transmission of event messages to nodes with 4 or more neighbours

In Figure 6, the TCL script is run for 72 seconds whereas in Figure 8 the TCL script is only run for 10 seconds. Thus, not all the nodes along the perimeter of the mobile nodes path are shown receiving messages from the mobile sink in Figure 8. The number of messages for node 13 and node 24 are reduced by half when the number of neighbour nodes restriction rule applies.

Figure 9 shows the number of messages per individual node if a typical flooding message is sent from node 1 to a static node 0 located at the same X,Y coordinates as node 25. Flooding a message to the sink is the worst case scenario as more nodes receive (and possibly have to re-transmit) messages which depletes an individual node's limited energy and reduces node and network lifetime.

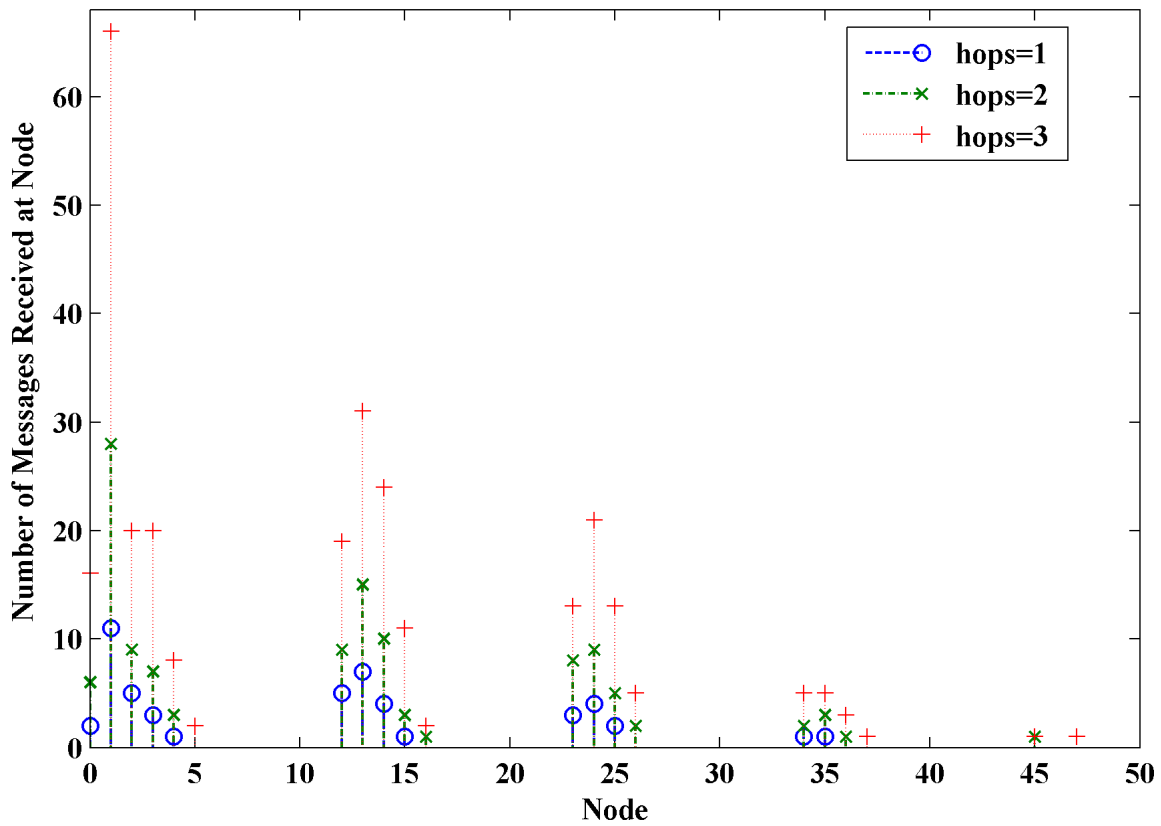


Figure 9. Number of messages per node for a flooding message sent to a static node 0 located at node 25.

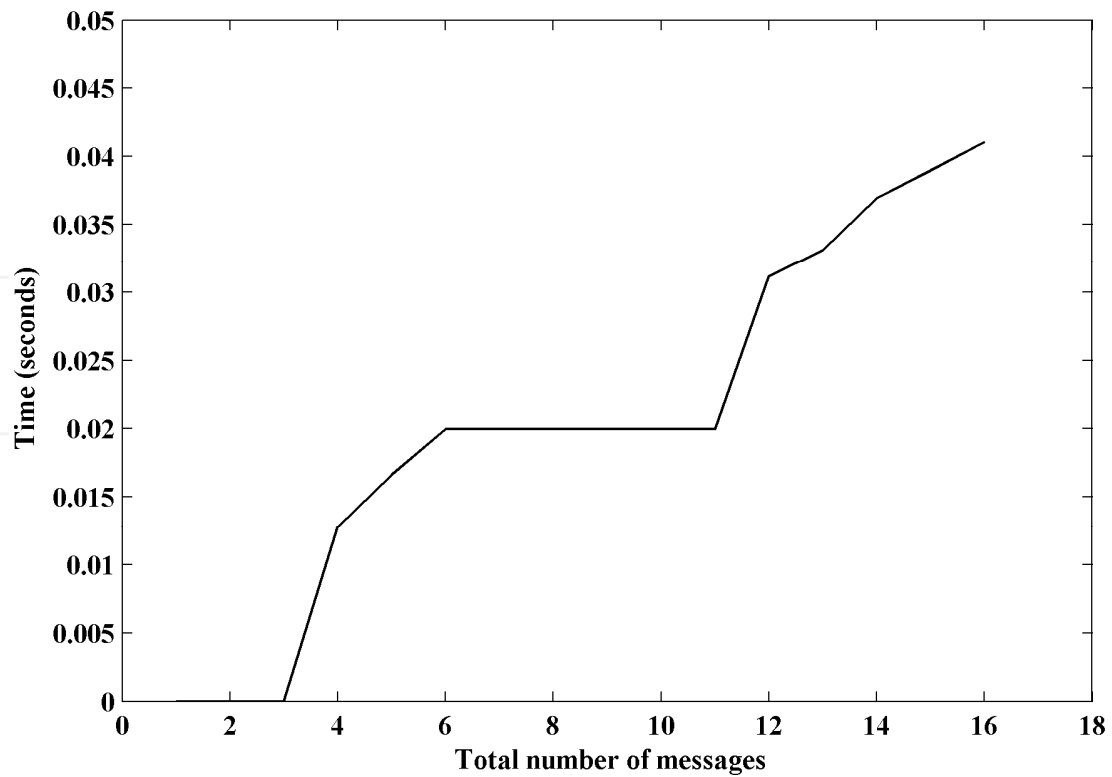


Figure 10. Time it takes for an event message sent from node 1 to reach node 14 or node 24

The time it takes for an event message to reach node 14 and node 24 using the mobile algorithm with re-transmissions of event messages restricted to nodes with four or more neighbours is shown in Figure 10. If the mobile node has just passed out of range of node 24 and node 14, then these nodes must wait for approximately 72 seconds before the mobile node is within range again. Alternatively, depending on the real-time requirement of the application, the perimeter nodes can retransmit the event message as electromagnetic waves travel faster than the mobile sink and the message should reach the sink in less than 72 seconds.

7. Conclusion

An optimum path for a mobile sink is calculated so that the number of hops that the message has to be re-transmitted is small. Because all neighbouring nodes can pass an event message to the sink, no specific set of nodes is overloaded with the task of routing event messages to the sink. This ensures more equitable usage of all sensor nodes in the network and hence increased node lifetime.

It has been shown that the number of messages received per node can be reduced by using a specific path for the mobile node/sink to move along. All neighbouring nodes can store messages when an event occurs, and if the sensor detecting the event is not an immediate neighbouring node along the path of the mobile sink, the number of hops that the message has to be re-propagated is small. By restricting the nodes that re-transmit a message to nodes with four or more neighbours, the number of messages received per individual node is further reduced.

Thus the use of a mobile sink moving along a calculated path around the application area can significantly reduce the number of messages received per individual node and hence increase node lifetime.

Author details

S. Chinnappen-Rimer
*Department of Electrical and Electronic Engineering Science,
 University of Johannesburg, Johannesburg, South Africa,*

G. P. Hancke
*Information Security Group, Royal Holloway, University of London, Egham,
 United Kingdom and Department of Electrical, Electronic and Computer Engineering,
 University of Pretoria, Pretoria, South Africa*

8. References

- [1] I. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, "Wireless sensor networks: A survey," *Computer Networks Journal*, vol. 38, no. 4, pp. 393-422, 2002.

- [2] P. Rentala, R. Musunuri, S. Gandham and U. Saxena, "Survey of Sensor Networks.," 2002.
- [3] C. Wang, K. Sohraby, B. Li, M. Daneshmand and Y. Hu, "A survey of transport protocols for wireless sensor networks," *IEEE Network*, vol. 20, no. 3, pp. 34-40, 2006.
- [4] M. Katz and S. Shamai, "Transmitting to colocated users in wireless ad hoc and sensor networks," *IEEE Transactions on Information Theory*, vol. 51, no. 10, pp. 3540-3563, 2005.
- [5] K. Akkaya and M. Younis, "A survey on routing protocols for wireless sensor networks," *Elsevier Journal of Ad Hoc Networks*, vol. 3, no. 3, pp. 325-349, 2003.
- [6] M. D. Francesco, S. K. Das and G. Anastasi, "Data Collection in Wireless Sensor Networks with Mobile Elements: A Survey," *ACM Transactions on Sensor Networks*, vol. 8, no. 1, pp. 7:1-7:31, 2011.
- [7] E. B. Hamida and G. Chelius, "Strategies for data dissemination to mobile sinks in wireless sensor networks," *IEEE Wireless Communications*, vol. 15, no. 6, pp. 31--37, 2008.
- [8] Y. Faheem, S. Boudjit and K. Chen, "Data dissemination strategies in mobile sink Wireless Sensor Networks: A survey," in *Proceedings of the 2nd IFIP conference on Wireless days (WD'09)*, 2009.
- [9] K. Akkaya, M. Younis and M. Bangad, "Sink repositioning for enhanced performance in wireless sensor networks," *Computer Networks*, vol. 49, pp. 512-534, 2005.
- [10] A. Somasundara, A. Kansal, D. Jea and D. Estrin, "Controllably mobile infrastructure for low energy embedded networks," *IEEE Transactions on Mobile Computing*, vol. 5, no. 8, pp. 958-973, Aug. 2006.
- [11] X. Huang, H. Zhai and Y. Fang, "Robust Cooperative Routing Protocol in Mobile Wireless Sensor Networks," *IEEE Transactions on Wireless Communications*, vol. 7, no. 12, pp. 5278- 5285, Dec. 2008.
- [12] S. Vupputuri, K. Rachuri and C. S. Ram Murthy, "Using mobile data collectors to improve network lifetime of wireless sensor networks with reliability constraints," *Journal of Parallel and Distributed Computing*, vol. 70, no. 7, pp. 767-778, July 2010.
- [13] Y. Gu, D. Bozdog, R. Brewer and E. Ekici, "Data harvesting with mobile elements in wireless sensor networks," *Computer Networks*, vol. 50, no. 17, p. 3449-3465, 2006.
- [14] M. Marta and M. Cardei, "Using sink mobility to increase wireless sensor networks lifetime," in *Proceedings of the 9th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM'08)*, 23-26 June 2008.
- [15] W. Heinzelman, A. Chandrakasan and H. Balakrishnan, "Energy-efficient communication protocol for wireless sensor networks," in *In Proceedings of the Hawaii International Conference on System Sciences*, Hawaii, 2000.
- [16] M. Patel, S. Venkatesan and R. Chandrasekara, "A Network-flow based Integral Optimal Algorithm for Lexicographic Maximum Lifetime Routing in Wireless Sensor Networks," in *Proceedings of 16th International Conference on Computer Communications and Networks*, (ICCCN 2007), 2007 .
- [17] W. Wang, V. Srinivasan and K.-C. Chua, "Using mobile relays to prolong the lifetime of wireless sensor networks.," in *Proceedings of the 11th annual international conference on Mobile computing and networking (MobiCom '05)*, Cologne, Germany, 2005.

- [18] L. Ben Saad and B. Tourancheau, "Towards an Efficient Positioning of Mobile Sinks in Wireless Sensor Networks inside Buildings," in *3rd International Conference on New Technologies, Mobility and Security (NTMS)*, 2009.

IntechOpen

IntechOpen