

CHAPTER 6

LOCALISATION IN A WSN

In this chapter, the ideas from Chapter 3, (i.e. placing multiple sinks at specific locations to model a WSN as a small world network) and Chapter 4 (i.e. a node re-transmits an IM if the received signal strength of the message indicates that a node is on the perimeter range of the transmitting node), is extended to calculate a node's location. The locations of three or more sinks and the number of hops from these sinks are used to calculate the nodes location within a WSN.

6.1 INTRODUCTION

Data received from sensors can only be useful if the data contains information about where the event occurred, i.e. the sensor location must be associated with measured sensor data. A message may have travelled many hops before it reaches its destination sink and the human interface will require the message to indicate the location of the event. Localisation is a technique to determine the location of a node within a WSN application area. Location estimation must be done in an energy efficient manner, especially for networks of sensors with small batteries that must last for years (Patwari & Kasera, 2011).

Accurate and low-cost sensor localisation is a critical requirement for the deployment of WSNs in a wide variety of applications, such as bush fire surveillance, water quality monitoring and precision agriculture. Low-power wireless sensors may be many hops away from any other sensors with *a priori* location information. Also, sensor location information (if it is accurate enough) can be extremely useful for scalable, "geographic" routing algorithms

In many WSN applications, the sensor device costs will need to be low, sensors will need to last for years or even decades without battery replacement, and the network will need to organise without significant human intervention. Traditional localisation techniques are not well suited for these requirements. Including a global positioning system (GPS) receiver on each device is cost and energy prohibitive for many applications, not sufficiently robust to jamming for military applications, and limited to outdoor applications (Patwari et al.,

2005). In cooperative localisation, sensors work together in a peer-to-peer manner to take measurements and then form a map of the network.

Methods used to estimate the location of sensors typically assume that small portions of sensors, called anchors or beacons, have *a priori* information about their coordinates. The anchor node coordinates may be obtained by using GPS or by installing the anchor nodes at fixed points with known coordinates (Mao et al., 2007). Non-anchor sensor nodes do not have *a priori* information about their location and location coordinates have to be estimated by a sensor network localisation algorithm.

An algorithm, the Perimeter Echo Algorithm (PEA), to determine the location of sensor nodes in a wireless network is discussed. The algorithm extends the ideas discussed in Chapters 3, 4 and 5 to optimise routing of messages within a network by using the ideas of multiple sinks with known location (as in the small world model) and an initialisation message to calculate a node's location.

The algorithm assumes a limited number of nodes (anchor nodes, beacons or sinks) in the network are equipped with a location finding mechanism such as GPS. Note that these anchor nodes can be sinks also, but are not required to be a sink. Nodes on the perimeter range of a beacon echo the beacon's virtual ID to all nodes within range. This approach reduces the number of messages required to achieve network localisation. The advantage of PEA is reduced power consumption and reduced flooding of the network because the beacon nodes only transmit an initialisation message once within a specific range and only those sensor nodes on the edge of a transmitting node's range re-transmit the message. This reduces the flooding that typically occurs during iterative and collaborative multilateration approaches.

6.2 ALGORITHM DESIGN

PEA is a distance-based localisation algorithm based on graph theory, where the Euclidean coordinates of all non-anchor nodes in the network are calculated from the known Euclidean coordinates of a small number of beacon nodes (Mao et al., 2007). PEA requires at least three known distances to beacon nodes to form a trilateration graph. The known

distance and coordinates of three beacon nodes are used to estimate the location of a sensor node.

Assumptions

- The algorithm assumes that the sensor antennas are omnidirectional.
- Each node has sufficient memory to store a location table that can store at least three rows, containing the virtual ID of a beacon node and the number of perimeter nodes a message has to hop across to reach the specified beacon node.
- The size of this location table is dependent on the node density.

Consider a WSN with n nodes and m beacon nodes. The beacon nodes have some form of location-finding system or are given their location a priori. The sensor nodes store the virtual IDs of three beacon nodes and their distance from the beacons (measured in terms of number of hops from a perimeter node). This information is appended to every sensing data message the node transmits. The receiving server can use this information to perform a simple trilateration algorithm to calculate the position of the node in the network.

Since energy is used to receive and transmit messages, the objective of the PEA algorithm is to limit the number of messages in the system to achieve network localisation of all nodes. Also, data messages from sensor nodes are generally evaluated in a central server location (which is most probably connected to a power source), and therefore there are sufficient resources to perform calculations.

Network localisation messages are transmitted on initialisation of the network. When the nodes are deployed, only the beacons (i.e. sensors with known positions) transmit a broadcast message containing the following information:

- ID: unique node identification number,
- latitude,
- longitude,
- altitude, and
- hops: number of hops from the beacon measured in terms of number of messages re-transmitted from nodes located on the perimeter of the transmitting node's range.

All nodes within range of a beacon node, update their location table with the above data. Those nodes located on the perimeter of the transmitting nodes range, increment the number of hops and re-broadcast the above message to all sensors within range. This process continues iteratively until all nodes have three beacon nodes' data in their location tables.

In practical applications the radio signal strength can be used to determine whether a node is on the perimeter. In the computer simulation, all nodes with a distance greater than 0.8 of the nodes range (radius) but which were still within the transmitting nodes range were considered to be perimeter nodes. The perimeter nodes will re-broadcast the received message. The simulation assumes that signals received by the nodes in range $0.8 * R \leq \text{signal} \leq R$, (where R is the node radio receiving range), can be correctly decoded.

A receiving sensor node checks its location table and if it is within immediate range of the specified beacon, the message is ignored as the data already exists in the location table. A receiving sensor that is not within immediate range of the beacon node, updates its location table. The location table data is stored in a sorted table with the nearest beacons at the top of the table. The algorithm determines a loop-free route to a beacon, since the shortest path to a beacon is always chosen.

In summary, the rules of the PEA are described below:

REQUIRE All beacons transmit ID message on deployment

IF {node in range}

Add data to its location table

IF {node on perimeter}

Increment number of hops

Re-transmit message

ENDIF

ENDIF

Figure 6.1 illustrates the algorithm. A beacon node (1, depicted as a triangle) transmits an initialisation message. The sensor nodes (2, 3, 4, and 7, depicted as rectangles) are within range. The sensor nodes (5, 6, 8, and 9) are out of range. Sensors 4 and 7 are on the perimeter of node 1 and retransmit the message.

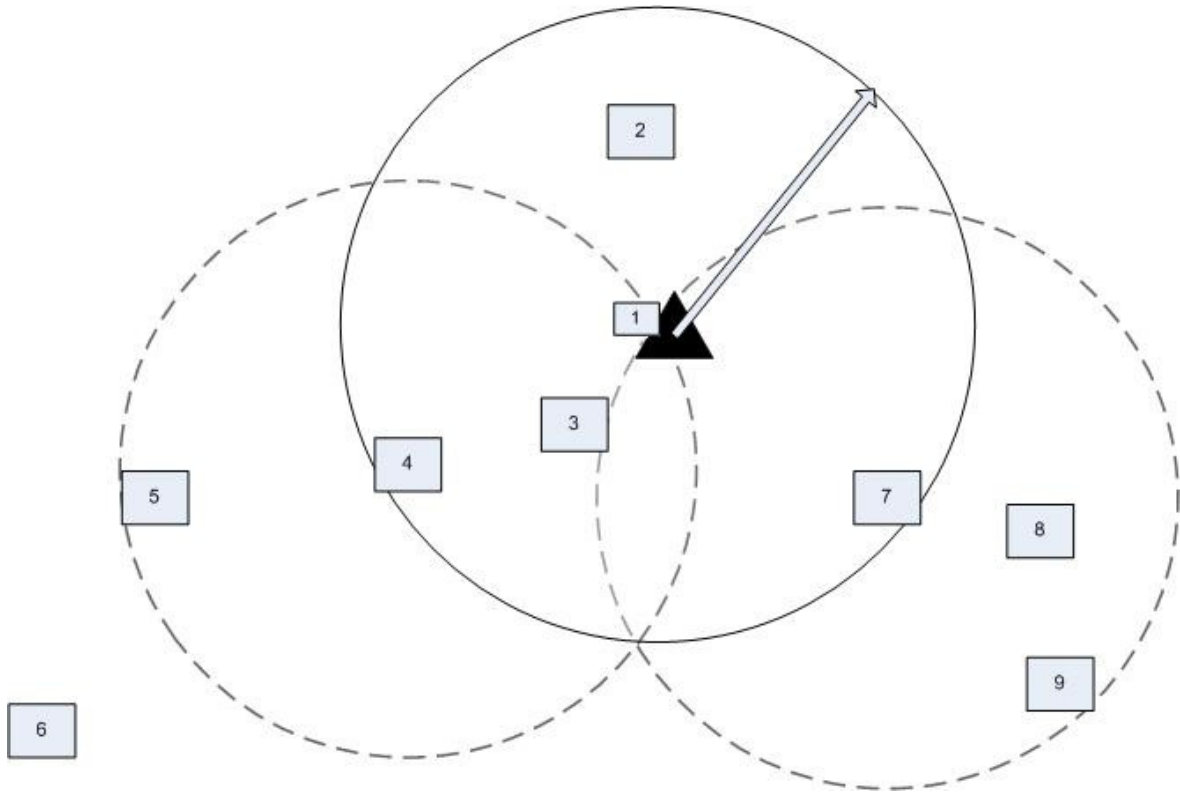


Figure 6.1: Overview of the Perimeter Echo Algorithm

After an event occurs, the sensing node(s) will transmit a message to the sink informing the sink about the event. This message will include the three nearest beacons' IDs and location, as well as the number of nodes the event is from each beacon.

The receiving server uses the data about each beacon's latitude and longitude to determine each beacon's location. An algorithm on the server uses the location coordinates of the three beacons as well as the number of hops from a beacon, to determine the location of the event.

Since only the nodes on the perimeter of the transmitting node's range, re-transmit the message, the number of hops provide a reasonable estimate of the distance from the beacon

in terms of $R \cdot hops$, where R is the transmitting range of a node. Therefore, from Figure 6.2, node 'A' is five times the radius range from beacon 1, eight times the radius range from beacon 2, and 28 times the radius range from beacon 3. The intersection of the three circles provides the approximate location of node 'A' as described in Figure 6.2. The sensor network application area is assumed to be small enough to ignore curvature issues that arise in large distances.

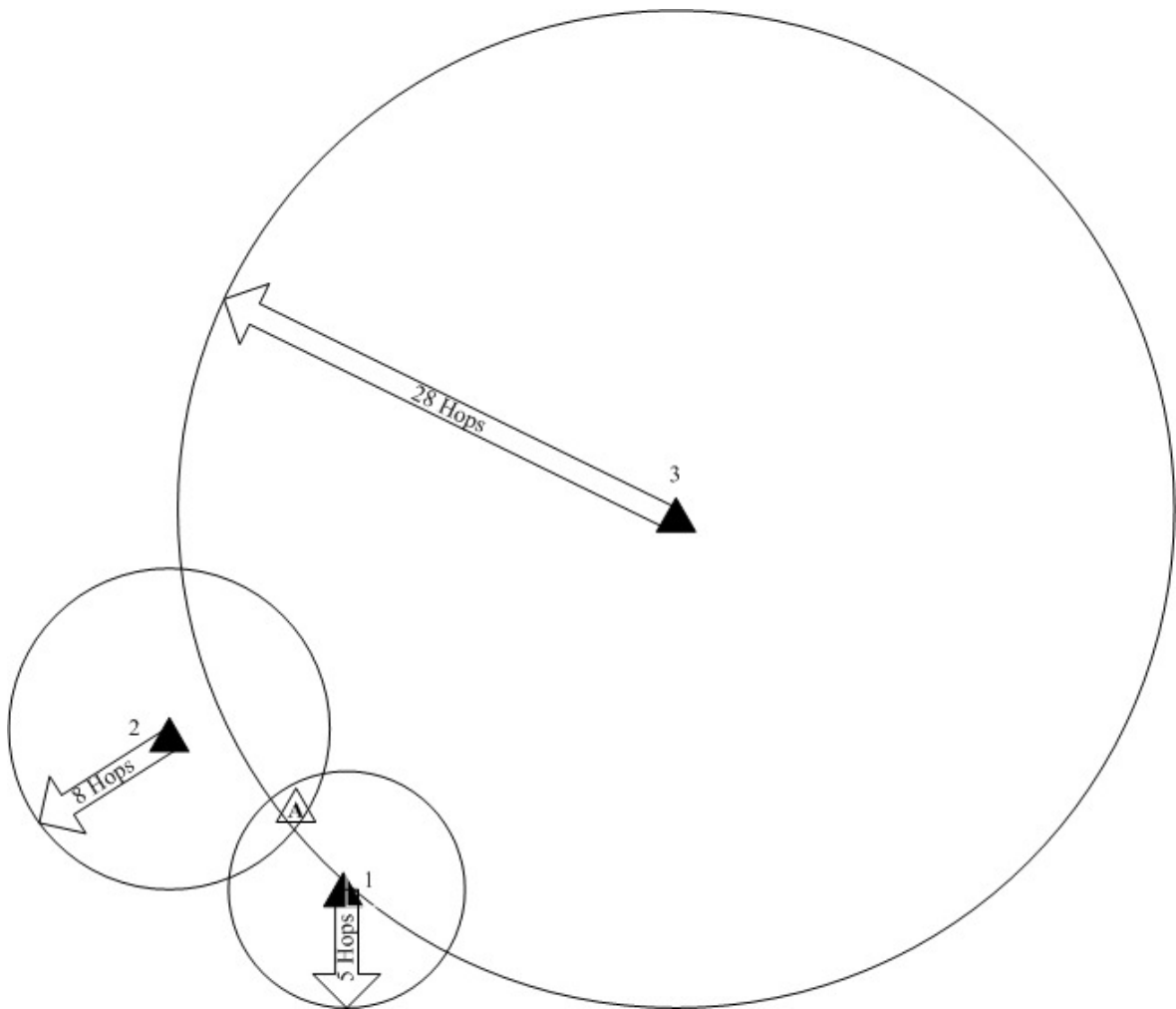


Figure 6.2: Location of node using trilateration

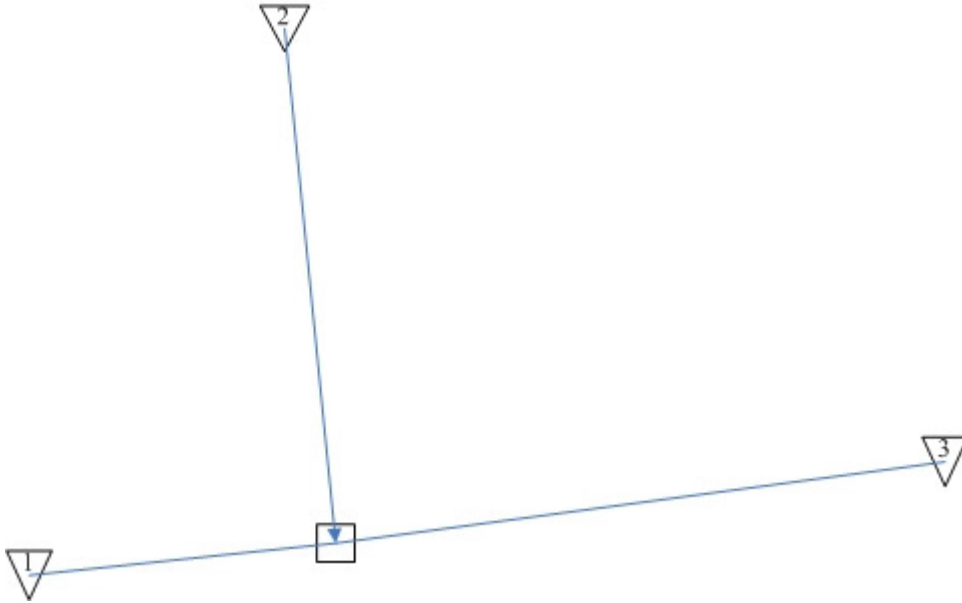


Figure 6.3: Calculating node location

The receiving sensor uses the three beacons' location to determine the distance each of the beacons is from the others. The algorithm then calculates the point of intersection of the radius (calculated as the node range times the number of hops from the node) from each beacon to determine the location of the node, as shown in Figure 6.3.

6.3 RELATED WORK

The iterative multilateration approach is used by Savvides et al. to determine the location of all unknown nodes in a network. On network initialisation a small number of nodes know their location. The unknown nodes determine their distance from their neighbours, using a node location technique such as time of arrival, and then use this information together with data about the known neighbourhood beacons to estimate their position (Savvides et al., 2001). This approach requires that the calculation of node position takes place at an unknown node and that each node broadcasts its position to all nodes within range, resulting in flooding and depletion of the power supply.

The solution presented in this chapter differs from this approach in that each node keeps a location table containing the virtual IDs of the beacon nodes and their distance from the beacon (i.e. number of hops). The node does not calculate its position but attaches the information in the location table to all messages it transmits. The receiving server

calculates the node position using trilateration techniques. Also, a node will only re-transmit beacon data if it lies on the perimeter of the transmitting nodes range. This approach should reduce the number of messages present in the system and thus the amount of energy consumed in processing received messages.

Biswas et al. describe a set of semidefinite programming (SDP) algorithms to determine the location of nodes in a WSN (Biswas et al., 2006). The primary disadvantage of this approach is that SDP programs do not scale to larger densities very well.

Patwari et al. suggest a form of cooperative localisation whereby sensor nodes cooperate in a peer-to-peer manner to form a map of the network. They use statistical models to calculate localisation performance bounds on location estimation precision based on time of arrival, angle-of-arrival and received-signal-strength measurements (Patwari et al., 2005).

In describing solutions to the broadcast storm problem in ad-hoc networks, (Chen et al., 2002) and a comparison of broadcast techniques in mobile ad-hoc networks (Williams & Camp, 2002), the authors mention a distance-based scheme, where the distance between the transmitting and receiving nodes is used to determine if a message should be re-transmitted. These schemes were not tested in a WSN environment, and do not separate the radio signal into accurately decodable data and radio signal with corrupt and non-intelligent data.

6.4 EXPERIMENTAL SIMULATION

A simulation of WSN deployment was developed to test the algorithm. The simulation program was coded in Java on an Intel Pentium computer using the Windows XP operating system. As this is a simulation, the Euclidean distance between two nodes was calculated to determine if a node was on the perimeter. All nodes with a Euclidean distance greater than $0.8R$, (where R is the range or radius of the transmitting node) and less than R are said to be on the perimeter of the transmitting nodes range and are flagged to re-transmit the message. In a practical application, the radio signal strength can be used to determine if the

receiving node is on the perimeter of the transmitting nodes area. Two experimental scenarios were considered.

6.4.1 Scenario 1

Beacon or sink nodes (i.e. nodes with known location) were placed approximately equidistant from one another in a 500mx500m application area. Sensor nodes were randomly placed on the panel. The simulation was run for a range of nodes from 100 sensors to 5000 sensors. The number of beacon nodes is calculated as 10% of the number of sensor nodes. The purpose of scenario 1 is to evaluate the PEA on its own strengths without ascribing any additional advantages or disadvantages from the small world model on the algorithm. It is for this reason that the number of beacon nodes was calculated at 10% of the number of sensor nodes.

6.4.2 Scenario 2

To further expand on the work from Chapter 5, the number of hops required to reach a predefined number of sinks for an application area size varying from 100m to 5000m was evaluated. The number of hops to a given number of beacons (sinks) for a given application area size was calculated based on the previously described small world model. The beacons were placed within the application area at specific points that was calculated from the number of hops of a node on the perimeter of the application area. From chapter 5 (Figure 5.1), the number of messages decreases as the number of hops is greater than 4. Therefore, experiments were run with the number of hops ranging from 4 to 8.

6.5 RESULTS AND ANALYSIS

6.5.1 Scenario 1

The number of times a beacon node's location message was re-transmitted until each sensor node has received three beacon location messages is shown in Figure 6.4. The graph provides a comparison with flooding, (similar to iterative and collaborative

multilateration). The PEA algorithm significantly reduces the number of messages required for all nodes to receive three beacon initialisation messages.

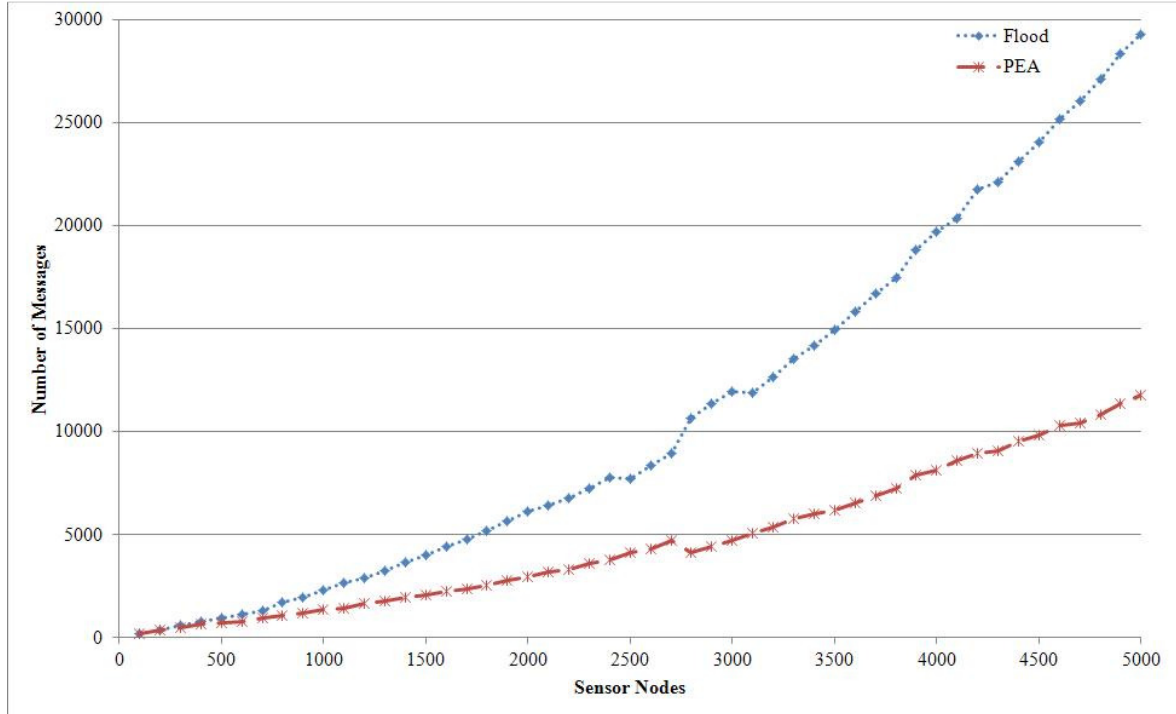


Figure 6.4: Number of beacon initialisation messages re-transmitted in WSN

There is a direct relation between the number of beacon nodes and the number of location messages re-transmitted. The number of hops to obtain the first beacon decreases to 0 as the number of beacons deployed within the specified area increases.

In addition, there is a significant advantage with PEA in the time required to obtain the IDs of three beacon nodes, as shown in Figure 6.5. While the time taken to obtain three messages is dependent on the simulation and supporting computer hardware, the difference between PEA and flooding indicate that initialisation of the WSN will be achieved faster using PEA for localisation.

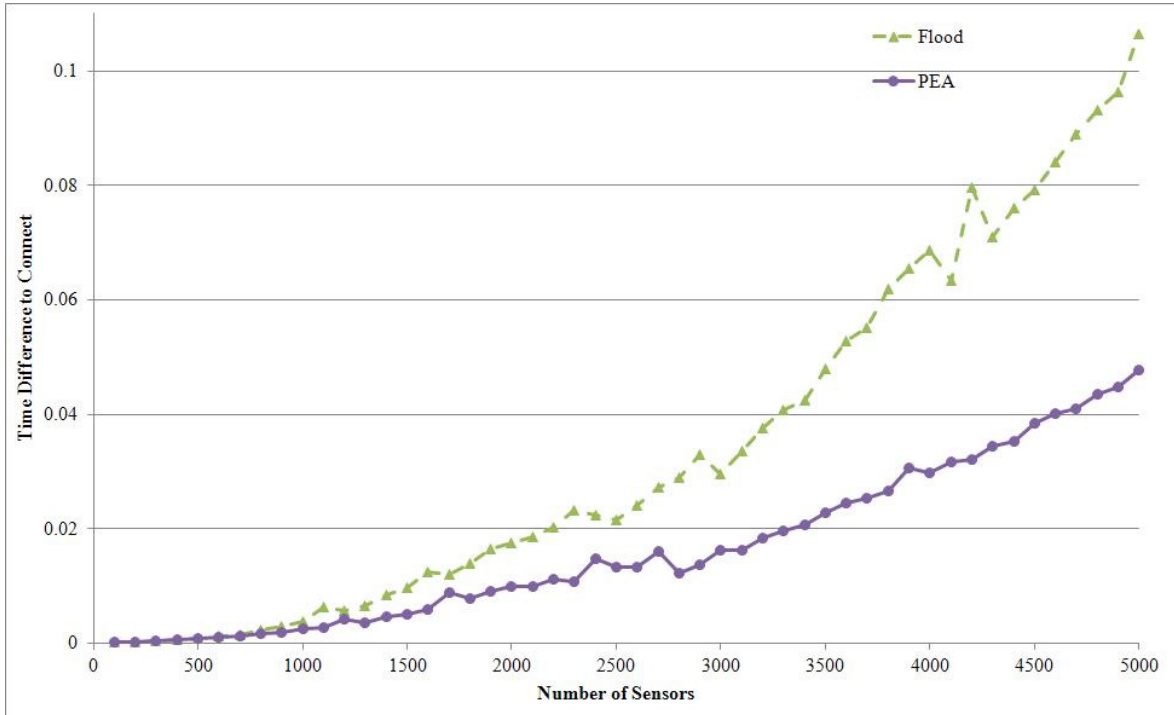


Figure 6.5: Time difference from start to end for each node to receive three beacon messages

6.5.2 Scenario 2

The number of hops required to reach a predefined number of beacons (i.e., 4, 5, 6, 7 and 8 beacons) for an application area size varying from 100mx100m to 5000mx5000m was calculated based on the small world model and is shown in Figure 6.6. Note; this is the number of calculated hops required to reach any **one** beacon from any furthest point within the application area. However for localisation calculations, the location of at least three beacons has to be known. In Figure 6.7 the number of hops from the three closest beacons to a node is shown as the application area size varies from 100m to 5000m. As expected the number of hops will be larger.

What is interesting is that for the three closest beacons the maximum number of hops is almost double the calculated number of hops from any node to a single beacon. This means that placing beacons at specific points according to the small world model guarantees that messages from at least three beacons can reach any node in a WSN from approximately twice the number of hops required to reach at least one beacon. As shown previously in

chapter 5, the number of initialisation messages increases as the number of hops declines. Thus, a WSN application designer can make a decision based on calculated number of hops for the minimum number of required beacons to evaluate the impact of the initialisation message on the application.

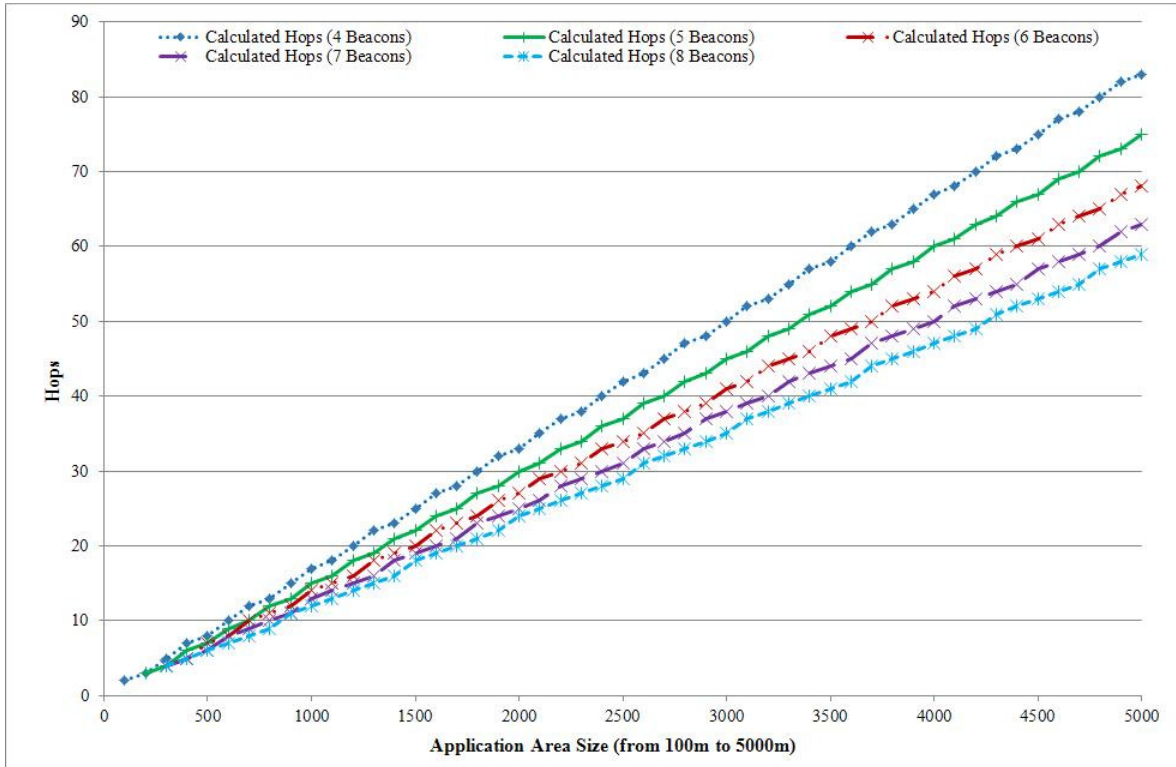


Figure 6.6: Number of hops for a specific number of beacons as application area size varies

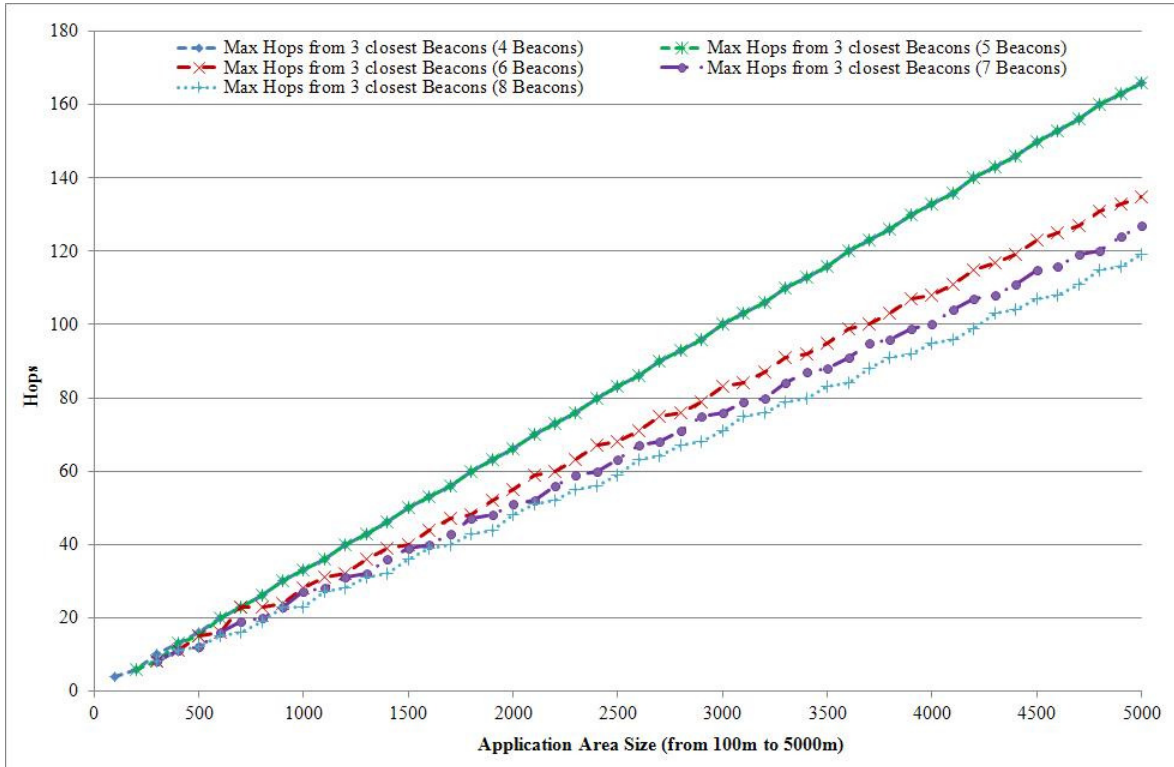


Figure 6.7: Number of hops from 3 closest beacons as application area size varies

As can be seen in Figure 6.7 the maximum hops to any three beacons when there are only for 4 and 5 beacons in a WSN are the same.

6.6 CONCLUSION

It has been shown that the PEA results in a significant reduction in the number of messages required to obtain three beacon localisation messages, as well as a reduction in the time taken for this initialisation. This means that there will be power savings as the number of times a node receives and/or re-transmits a message decreases. The control centre will be able to calculate the location of an event reasonably accurately using the three beacons' IDs supplied in the event-sensing message.

When the WSN is modelled as a small world network and a specific number of beacons were calculated, the number of hops from any three beacons is at most twice the minimum number of hops required by a node to reach the nearest beacon. Thus, the cost of sending

the additional three beacons' data in an initialisation message is at most twice that of sending an initialisation message from only one beacon.

Using the PEA, a WSN irrespective of whether it has been modelled as a small world network or not, should be able to localise all nodes in the network within a short time while prolonging node lifetime.

6.7 DECLARATION

The work in this chapter has been published at the following Conference:
IEEE Africon 2009.