# Scalable and Energy Efficient Localisation in Wireless Sensor Networks

Daniel J. E. Erasmus, Gerhard P. Hancke

Department of Electrical, Electronic and Computer Engineering, University of Pretoria, Pretoria, South Africa

## ABSTRACT

A need exists for localisation algorithms in Wireless Sensor Networks (WSNs) that are scalable, energy efficient and able to function in easy to deploy sensor networks. This paper proposes a beaconless Cluster-based Radial Coordinate Establishment (CRCE) positioning algorithm to locate sensor nodes relative to a local coordinate system. The system does not make use of Global Positioning System (GPS) or any other method to provide a-priori positioning information for a set of nodes prior to the CRCE process.
The objective is to reduce energy consumption while providing a scalable coordinate establishment method by focussing on the minimisation of message exchanges in a WSN. This is achieved by implementing a cluster-based network topology and utilising the processing potential of geographically distributed sensor processors together with radial coordinate propagation.
Three other localisation algorithms are investigated and compared to CRCE to identify the one best suited for coordinate establishment in WSNs. The results show a significant decrease in the number of messages that is necessary to establish a network-wide coordinate system successfully, ultimately proving the CRCE method to be more scalable and energy efficient.

KEYWORDS: Beaconless, cluster-based, clusterless, coordinate establishment, energy efficient, iterative convergence, localisation, positioning, radial convergence, Wireless Sensor Networks.

## 1 INTRODUCTION

Wireless Sensor Networks (WSNs) refer to a group of spatially deployed devices which are used to monitor or detect phenomena, and have the ability to relay sensed data and signalling wirelessly. It is becoming an increasingly attractive means to get insight into the behaviour and characteristics of modern day dynamic systems - valuable data are collected, integrated, and utilised to enhance production processes, increase revenue, and decrease security risks, to name but a few.

WSNs do not exist without their challenges though. It can consist of thousands and even millions of wireless sensor nodes. Therefore, *scalability* is a critical factor in the system design. *Energy efficiency* is another major concern in WSNs; sensor nodes are very small, low cost devices and do not provide a means for recharging batteries or installing large power supplies.

Any number of resource constrained sensors can be deployed in an ad-hoc fashion with no a-priori knowledge of their positions. For optimal routing and data to be meaningful, location information is an absolute must. Thus, *localisation* is yet another challenge pertaining to WSNs. In this paper we address all three of these key issues by proposing a Cluster-based Radial Coordinate Establishment (CRCE) method. The objective of CRCE is to reduce energy consumption while providing a highly scalable coordinate establishment method for use in WSNs.

We investigate three other localisation algorithms and compare them with CRCE. Two of these are already presented in the literature and the third is a modification of one of them.

The organisation of the rest of the paper is as follows. Section II covers some background aspects on coordinate establishment in WSNs and sets the scope for the proposed algorithm. CRCE is described in Section III. In Section IV we take a brief look at the other three algorithms that are being compared to CRCE. The analysis and results are presented in Section V and we conclude the paper in Section VI.

**Email:** Daniel J. E. Erasmus `elardus@ieee.org`, Gerhard P. Hancke `g.hancke@ieee.org`

## 2 BACKGOUND AND SCOPE

Node coordinates are used as a means of addressing, in conjunction with node IDs, to facilitate tasks or actions such as routing, information gathering and data aggregation.

Various proposals have been made on how to establish a network wide coordinate system in WSNs. Most of them make use of GPS or any other method (even manual configuration) to establish a set of beacon/anchor nodes with known locations from where the positioning algorithm will be started or referenced to [1, 2, 3, 4, 5, 6].

The main problems associated with GPS are increased power consumption, its large form factor (compared to small, inexpensive sensors), poor indoor reception and high production cost; while manual configuration of sensor coordinates severely affects the rapid deployment of scattered sensor networks.

Other methods to decrease energy consumption by minimising to message cost are also proposed. The authors of [7] make use of local node capabilities to internally position only the groups of nodes involved in particular conversations, whereas [8] suggests a method called Sectoral Sweepers. This entails the use of one centrally located node with an amplified directional antenna. This node delivers task orders directly to the region of interest and receives aggregated data replies on a hop-by-hop basis. A drawback of this method is that the network cannot be deployed freely, e.g. dropped from a plane. The central node needs to be positioned intelligently.

[1], [5], [7], and [9] propose techniques to estimate and/or measure node distances as well as ways to mitigate the errors. This can be considered as a research area in its own right, and falls outside of the scope of this paper. The proposed algorithm focuses on a way to communicate the distances efficiently while establishing a network wide coordinate system, instead of doing distance error correction and refinement.

CRCE is a network layer protocol, thus, medium access, transceive timing, and synchronisation issues are addressed by the lower layers and also does not fall within the scope of this paper. Nevertheless, to enable us to establish the message cost of different localisation algorithms, it is necessary to know the number of resultant messages on the physical layer, for each type of message (broadcast or unicast) on the network layer. We base our simulations on a MAC layer with the following characteristics:

- Each node (node $i$) has knowledge and is synchronised to its direct neighbours only; number of neighbours of node $i = n_i$.
- Broadcast messages: one message on the network layer, labelled as a broadcast message, results in $n_i$ unicast messages on the lower layers - one for each neighbour.
- Unicast messages: a message from the network layer of node $a$, to node $b$, results in one unicast message between the lower layers of these two nodes.

CSMA-MPS is an example of such a MAC protocol [10].

## 3 CLUSTER-BASED RADIAL COORDINATE ESTABLISHMENT

The total energy consumption ($E_T$) in a wireless sensor, as given in [11], is:

$$E_T = E_{TX} + E_P \qquad (1)$$

$E_{TX}$ is the energy used to transmit, receive and amplify data, and $E_P$ is the energy used to process the data. $E_{TX}$ dominates $E_P$ [12], and therefore, by minimising the number of messages transmitted by the nodes, one can effectively reduce the energy consumption of the entire network. This also enhances the scalability of the network.

CRCE employs two methods to achieve this. Clustering is one of these, and we compare our algorithm to that of another beacon-less cluster-based algorithm presented in [13].

The other method is for coordinate updates to traverse the network radially, meaning from a specified node outwards, instead of having all nodes converge simultaneously in an iterative manner, as in [14].

### 3.1 The CRCE Network topology

The network topology of CRCE is one composed of clusters. Each cluster consists of one master node, the cluster head (CH), and some slave and border nodes. The master nodes compute the local coordinate systems of clusters and manage relative convergence in the network. One of the master nodes in the network will assume the role of a sink node. The sink node initiates the coordinate establishment process once the network is deployed. By definition, no master node can be a neighbour of any other master node.

Slave nodes are one-hop neighbours of master nodes. Slaves do not communicate directly (except in the very first stage of coordinate establishment) and provide a cluster-based environment in which message cost reduction is possible.

The border nodes are essentially slave nodes that are part of two or more adjacent clusters. Put differently, a border node is a slave node within communication distance of more than one master node. These nodes are used for coordinate establishment purposes.

Note that the classification of nodes is only logical. Physically, these nodes are identical.

### 3.2 Message types

CRCE makes use of different message types to accomplish network-wide coordinate convergence. Seeing as this is a network layer protocol, these messages are called *packets*. Packets consist of the *header* (all the information required for CRCE) and optional upper layer data.

A *preamble* precedes every CRCE packet header. It consists of the *destination* and *source* addresses, and indicates the packet *type*. Nodes use the type information to distinguish between the packets being used in different stages of the CRCE process. We make use of six types of packets which can be represented by three binary digits (Table 1):

- P1 and P2: The header for these consists of only the preamble. They are used for distance recording and node type establishment in the first stage of the CRCE process.
- P3: The header for packet 3 carries information on all the one-hop neighbours of the nodes and consists of the preamble, neighbour node ID's, types (master, slave or border) and distances to them.
- P4: This header contains the position coordinates of a slave. Its format is: preamble, converged ID (the ID of the node to which its master converged to) and coordinate.
- P5: The header for packet 5 carries the correction angle together with the same information as in packet 4's header. The format is: preamble, converged ID, coordinate and correction angle.
- P6: Preamble, converged ID and correction angle. This is the same correction angle as in P5, that is passed from master to master via a border node using P5 and P6.

### 3.3 CRCE Algorithm

CRCE consists of two stages, each comprising of two phases. They are described next.

*Table 1:* Packet types

| Type | Bin | Sent from | Sent to | Purpose |
|---|---|---|---|---|
| 1 | $001_2$ | Masters | All neighbour nodes | Distance recording and establishment of node types: itself as master and the neighbours as slaves. |
| 2 | $010_2$ | Slaves | All neighbour nodes | Distance recording and master node discovering. Neighbouring nodes get to be masters by setting a random timer upon receipt of this packet; if the timer expires before a master node contacts it, it assumes the master node role. |
| 3 | $011_2$ | Slave/border nodes | Master | Reports first hop neighbours' IDs along with the distances to them. |
| 4 | $100_2$ | Master | Slaves and some borders | Updates position coordinates. |
| 5 | $101_2$ | Master | Borders | Updates position coordinates and forwards the correction angle. |
| 6 | $110_2$ | Borders | Masters | Forwards the information required for the coordinate translation/convergence process (correction angle). |

#### 3.3.1 Stage 1: Neighbour discovery

The objective of this stage is to classify every node in the network as either a master, slave or border node, and then to allow the master nodes to obtain the necessary distance information. At the end of this stage the master node will have recorded all distances to its one-hop nodes, as well as the distances between its one and two-hop nodes. This will enable coordinate establishment in Stage 2. The algorithm for the first stage of CRCE is given (Algorithm 1).

**Phase 1: Distance and node type establishment.** At the beginning of Stage 1, no node, except the sink, knows its type (master, slave, or border node) and no messages have been exchanged between nodes.

The sink node starts the process by broadcasting P1 (packet 1) type messages to all surrounding nodes. Every node receiving a message records the distance to the master (in this case the sink) and labels itself as a slave node. These nodes, in turn, broadcast P2 messages which will trigger the countdown of a random timer at the nodes that are not yet classified. All nodes that receive P2 messages record the distances between themselves and the sender.

If the timer of a node expires before it receives a message from any other master node (P1 type message), it broadcasts a P1 message that stops the countdown at any node in its range/domain, ultimately establishing itself as a master and the other as slaves. Nodes that are shared by two or more masters will then classify themselves as border nodes. All nodes receiving this P1 message record their distances to the master and only those that have not previously broadcasted P2 messages do so now.

This cycle repeats until all nodes are classified and all distances between them are recorded.

**Phase 2: Neighbour distance reporting.** Every node now has a set of known distances to all its neighbouring nodes. The master nodes need to know the distances between their one and two-hop neighbours for coordinate calculation purposes. Once a slave/border node has received all distance updates from its neighbours (in the previous phase) it sends this information to its master node(s) using P3 type messages.

#### 3.3.2 Stage 2: Coordinate establishment

Every node in the network will have position coordinates relative to the sink node at the end of this stage. Algorithm 2 applies to the second stage of CRCE.

**Phase 1: Local coordinate calculation.** No messages are sent in this phase. Every master node calculates its local coordinate system by using triangulation, based on the distances obtained in the previous phase. It does not transmit the calculated positions of the slave/border nodes to them until Phase2.

The mathematical procedure followed for the local coordinate setup is presented in [14]. Stage 1 gathered the information necessary for this procedure.

**Phase 2: Relative coordinate convergence** All master nodes now need to reorient their local coordinate systems only once to converge to that of the sink node - this is the main reason for the energy efficient nature of CRCE. This is possible because coordinate updates traverse the network radially - from the sink node outward, the convergence messages are propagated through the network only once and the nodes converge along this path.

The master nodes calculated the positions of their slaves (including border nodes) in the previous phase. Using P4 type messages, these coordinates are now sent to the slave nodes to update them with their positions accordingly. The master node also sends the coordinates and correction angle to those border nodes that have the same second or third master. This is accomplished using P5 type messages, with P4 messages sent to border nodes that do not share a second or third master.

In turn, each border node then transmits P6 type messages to all their masters that have not yet converged to the coordinate system of the sink node. These messages contain information to perform the necessary rotational and translational calculations. The converged masters then broadcast P4 and P5 messages in their domains and the cycle continues until all nodes in the network have known positions, relative to that of the sink node. The network is now converged.

The mechanism for determining the coordinate translation parameters for a system of three nodes which are

**Algorithm 1** CRCE algorithm: Stage 1

```
if sink node then
    broadcast(P1)
end if
if unknown node then
    wait for P1 OR M2 and record distance
    if M2 received then
        initialise random timer
        decrement random timer
        if (timer = 0) AND (no P1 received) then
            status ← master
            broadcast P1
            wait P2s and record distances
        else if P1 received then
            record distance
            if more than one P1 received then
                status ← border
            else
                status ← slave
            end if
            broadcast P2
        else if P1 received then
            record distance
            if more than one P1 received then
                status ← border
            else
                status ← slave
            end if
            broadcast P2
        end if
    end if
end if
if (slave node) OR (border node) then
    send P3 to master(s)
end if
```

**Algorithm 2** CRCE algorithm: Stage 2

```
if (master node) OR (sink node) then
    wait P3s and establish local coord. system
end if
if (sink) AND (coord. system established) then
    MasterSendCoordinates(sink)
end if
if (slave/border node) AND (P4 received) then
    update position coordinates
end if
if (border node) AND (P5 received) then
    update position coordinates
    send P6 to unconverged masters
end if
if (master node) AND (P6 received) then
    translate coordinate system
    MasterSendCoordinates(master)
end if

function MASTERSENDCOORDINATES(node)
    for all borders (b_i) of node do
        for all borders (b_y) of node do
            if b_y & b_i share 2nd/3rd master then
                send P5 w/ correctionAngle_iy to b_i
                messageSent ← TRUE
            end if
        end for
        if messageSent = FALSE then
            send P4 to b_i
        end if
    end for
    for all slave nodes do
        send P4 to slave
    end for
end function
```

aware of their mutual distances [14], cannot be applied in this phase. The reason for this is because, by definition, two master nodes cannot be within direct range of each other and therefore do not know the distance between them. Thus, using the angles and distances from the previous steps, together with the calculations described in [13], we obtain the translational parameters necessary for network-wide coordinate convergence.

## 4 LOCALISATION METHODS FOR COMPARISON

CRCE is compared to three other localisation methods. Two of them are presented in the literature and the other is a modified version of one of the two. All these methods can be divided into the same two stages as CRCE.

The first method to which we will compare CRCE is called the Self Positioning Algorithm [14]. It does not make use of clusters, neither does it converge radially to the coordinate system of the sink - we will call it *RC-less* for comparison purposes. Instead, each node has to reorient its coordinate system to that of the node with the lowest ID in the network. Effectively this means that the node has to reorient its system to the neighbour with the lowest ID repeatedly, until the neighbour has converged to the node in the network with the lowest ID. This is an iterative process and wastes a lot of energy due to the message overhead.

In Stage 1, every node sends a broadcast message to all their neighbours. These messages are used to log all the one-hop neighbours and record the distances between them in the same way as in CRCE. Once this information has been obtained, it is sent to all the one-hop neighbours in Phase 2 - unlike CRCE, where it is sent to the cluster heads only.

Each node now knows the distances to its one-hop neighbours. Also, it knows its two-hop neighbours as well as the distances between its one-hop and two-hop neighbours. This information is now used in Stage 2 to establish a relative coordinate system.

The convergence process takes place during Stage 2 - the same for all methods. RC-less employs iterative convergence as opposed to CRCE's radial convergence in this stage.

The second method is also not cluster-based but does indeed converge radially - we'll call it *C-less*. It is a modification of RC-less and was designed during this research to study the effect of radial convergence in clusterless network topologies; and to compare it against CRCE.

Stage 1 of C-less is exactly the same as for the RC-less method - both are clusterless. Neighbour nodes are discovered and distances between them recorded. In contrast to the iterative convergence of the RC-less method in Stage 2, coordinate updates in the C-less method traverse the network radially.

Another difference is that the network converges to a designated *sink node*, rather than the node with the lowest ID - the same as CRCE.

The third method, presented in [13], is cluster-based but does not converge radially - we'll call it *R-less*.

The neighbour discovery stage of the R-less method is exactly the same as in CRCE - both are cluster based. The slaves and border nodes send the distance information to their master nodes only. This reduces the number of messages transmitted in Stage 1 when compared to the clusterless methods.

In Stage 2, each master has to reorient its coordinate system to that of the master node with the lowest ID in the network. This effectively reduces the number of nodes at which reorientation is necessary (compared to the clusterless methods), but ultimately shares the unwanted iterative property present in the RC-less method.

## 5 ANALYSIS AND RESULTS

Numerical equations are presented in this section to enable the calculation of message cost for these four localisation methods. Simulation results confirm the validity of the numerical formulas, and were obtained by programming each of the four algorithms in the OMNeT++ simulation environment [15] and making use of the Mobility Framework [16].

The numerical and simulation setup is described: we consider a flat square area with an edge length of $L$ units. We then assume that the area is covered with a uniform distribution of stationary nodes with density $\lambda$ nodes/unit$^2$. Each node has a transmission range of $r$ units.

Results were obtained for each of the two stages of every method and are presented next. After that we combine the results for the two stages to get an overall comparison.

### 5.1 Message cost: Stage 1

We consider the operation of the methods during Phase 1 and 2, with regards to the number of messages being transmitted to derive formulas for message cost in Stage 1. The two clusterless methods operate in exactly the same way in this stage; as do the cluster-based methods.

In Phase 1, the clusterless and cluster-based methods all transmit messages from each node to all of their neighbours. Therefore, the message cost for Phase 1 could simply be the product of the number of nodes in the network ($N_T$) and the number of neighbours for each node; but not all nodes in the network have the same number of neighbours - e.g. nodes located at the corners of the square network will have less neighbours than those in the middle, resulting in less transmissions. The same goes for edge nodes.

To accommodate for this in the message cost formulas, it is necessary to differentiate between nodes based on their geographical locations in the network. They are: edge nodes (nodes located at the edge of the network and can be considered as the network boundary), corner nodes (nodes located at the four corners of the rectangular network), and inside nodes (those nodes contained within the network boundary).

In Phase 2 of the clusterless methods, each node sends replies to all of its neighbours - the same amount of messages as in Phase 1. On the other hand, for the cluster-based methods, only slave and border nodes send replies to their masters (which are established in Phase 1). Equations (2) and (3) present the message cost for Stage 1 accordingly.

$$M1_{Clusterless} = 2\left[N_I{\cdot}n_i + N_E{\cdot}n_E + N_C{\cdot}n_C\right] \tag{2}$$

$$M1_{Cluster-based} = [N_I{\cdot}n_i + N_E{\cdot}n_E + N_C{\cdot}n_C] + N_K{\cdot}n_I \tag{3}$$

Where $M1$ is the total number of transmitted messages (message cost) in Stage 1, $N_I$, $N_E$, $N_C$ and $N_K$ are the number of inside nodes, edge nodes, corner nodes and clusters in the network, respectively. $n_I$, $n_E$ and $n_C$ are the number of neighbours for the respective node types, indicated by the subscripts.

We now describe how to calculate the values for the different variables in (2) and (3). Given: transmission range ($r$), and the number of neighbours for an inside node ($n_I$) - i.e. the number of neighbours for a node with no connection limitation due to its geographical location (e.g. a corner has connections only in one quadrant of its coverage area).

Calculate the node density ($\lambda$) according to 4.

$$\lambda = \frac{n_I + 1}{\pi r^2} \tag{4}$$

Choose the number of nodes in the network. Keep in mind that the simulation setup states a square network; an intelligent number of nodes has to be chosen to keep the node count for $N_E$ integer (in the next step), i.e. $N_T = i^2$, where $i = 1, 2, 3, \ldots$ The area of the network will then be $A = \lambda N_T$.

Now calculate the number of nodes for each of the three types of geographically classified nodes. According to the simulation setup, we have $N_C = 4$, $N_E = 4\left(\sqrt{N_T} - 2\right)$, $N_I = N_T - (N_E + N_C)$, and $N_K = A/\left(\pi r^2\right)$.

Then set the neighbour counts, $n_I$, $n_E$ and $n_C$. $n_I$ was given (or calculated in the first step if $\lambda$ was given, by rearranging (4) to solve for $n_I$). $n_E$ and $n_C$ were obtained by setting up a number of networks with varying densities; $\lambda = 0.4$ to 1.8. By counting the number of neighbours for the corner and edge nodes in these networks and plotting them on a graph as a function of $\lambda$, it was possible to establish $n_E = 7.02\,(\lambda)$, and $n_C = 4.31\,(\lambda)$. Figure 1 compares the message cost of the clusterless and cluster-based methods in Stage 1, by plotting the number of transmitted messages as a function of $\lambda$; (2) and (3). The simulation results are also presented on this graph.

It can be seen that the non-cluster based methods are almost twice as expensive as their cluster-based counter-

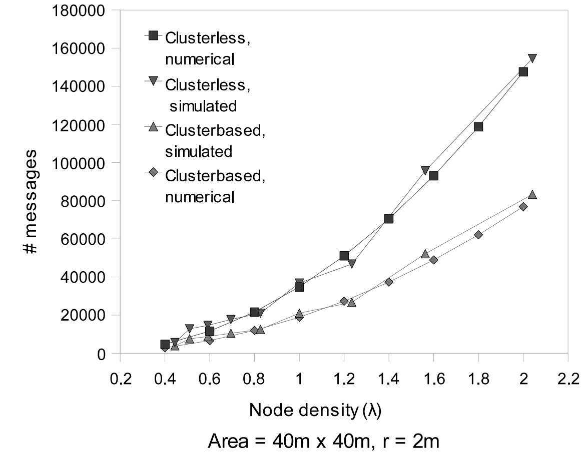


*Figure 1:* Stage 1 message cost - clusterless and clusterbased

parts. We also observe the close match between the numerical and simulation results, validating (2) and (3) as numerical approximations to the message cost in the neighbour discovery stage.

## 5.2 Message cost: Stage 2

Phase 1 of Stage 2 is concerned only with coordinate calculation and adds nothing to the message cost of the various localisation methods. All transmissions in Stage 2 take place during global coordinate convergence (Phase 2).

We describe the principle used to derive the mathematical formulas for the message cost in Stage 2 (without doing the actual derivation here):

1. Set up relatively small networks, $5^2 \leq N_T \leq 15^2$, with a node density that will enable simple neighbour connections ($\lambda \approx 0.716$ results in $n_I = 8$; easy to draw network).
2. Identify the nodes with different message transmission numbers. This can be based on geographical location (as in Stage 1) and/or network role (slave, border and master). For clusterless networks we still consider only corner, edge and inside nodes, but for cluster-based methods we classify corner nodes, two types of edge nodes (slave nodes and border nodes with two masters), and three types of inside nodes (master nodes, border nodes with only two masters, and border nodes with four masters).
3. Physically count and record the number of message transmissions for every type of node (identified in step 2) by following the algorithm for each localisation method in Phase 2 of the coordinate establishment stage.
4. Identify a pattern from this. Simplify and formulate it into equations so that it can be applied to networks of any size.

### 5.2.1 Clusterless methods - radial vs. non-radial

Equations (5) and (6) calculate the number of Stage 2 message transmissions of the RC-less and C-less methods respectively.

$$M2_{RC-less} = 0.952 n_I \sum_{i=0}^{\sqrt{N_T}-1} \left(N_T - i^2\right) \tag{5}$$

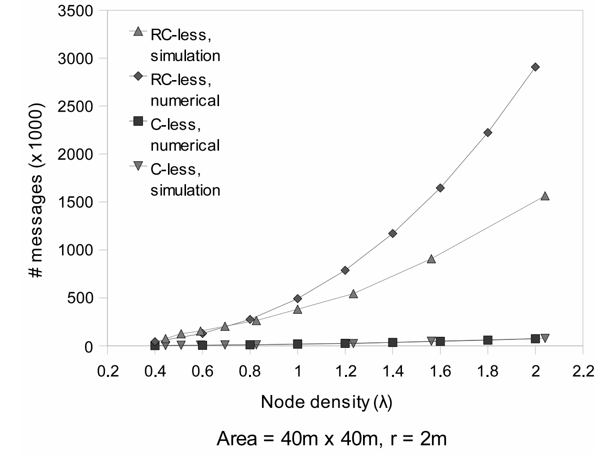


*Figure 2:* Numerical and simulation results for both C-less and RC-less

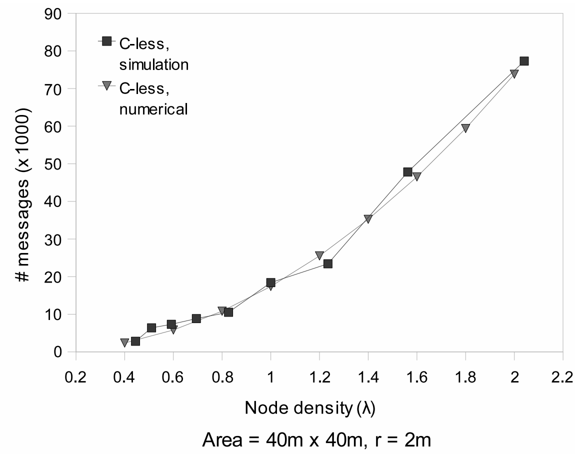


*Figure 3:* Numerical and simulation results for C-less only

$$M2_{C-less} = N_I \cdot n_I + N_E \cdot n_E + N_C \cdot n_C \tag{6}$$

Figure 2 is evidence to the enormous impact that radial convergence has on coordinate establishment in clusterless methods. The poor performance of the RC-less method is attributed to its inherent iterative property - evident in (5).

In Figure 3 we remove RC-less to better observe the match between the numerical and simulation results for C-less, validating (6) for computing the message cost of the C-less method in Stage 2. Figure 2, however, suggests that the difference between the numerical and simulation results increases as the density of the network increases. It is explained next.

As the density of the network increases, more and more nodes close to the edge of the network do not obtain the maximum number of neighbours that are possible for a given transmission range. The *maximum number of neighbours* can be defined as the number of nodes in the domain of node *i*, where node *i* is located such that the number of its neighbours are limited only by the density of the network, and not by the lack of node deployment in any sector of the coverage area - e.g. a corner node will not have a maximum number of neighbours.

The formulas for message cost do not take this into account. It assumes that all nodes within the boundary of the network have maximum neighbour connectivity. The reason why it is only visible (and of interest) in the RC-less method is that the error compounds with every iteration of

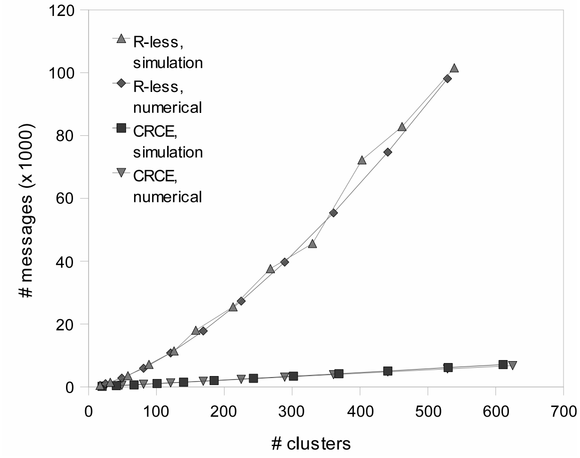


*Figure 4:* Numerical and simulation results for both R-less and CRCE

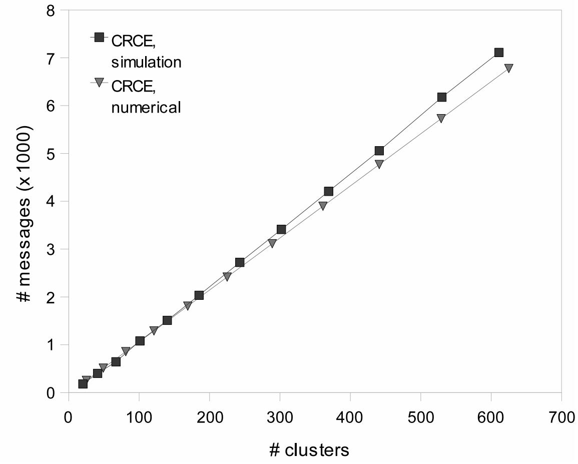


*Figure 5:* Numerical and simulation results for CRCE only

the convergence process. In a square network of 40 x 40 *nodes*, density of 1 $node/m^2$ and transmission range of 2 *m*, RC-less iterates 40 times and C-less only once. This results in a higher numerical message cost than the actual simulation cost.

Further investigation is necessary to consider this in the message cost formula for RC-less, but witnessing the very poor performance of this method, the effort is not considered to be worthwhile.

### 5.2.2 Cluster-based methods - radial vs. non-radial

The two cluster-based methods are compared next. This time the density is kept constant $(\lambda = 0.716\, nodes/m^2)$ and the message cost is calculated as a function of the total number of clusters in the network ($N_K$ is varied). Equations (7) and (8) are presented.

$$M2_{R-less} = \sum_{i=0}^{\sqrt{N_K}-2}\left(N_I - 2i\sqrt{N_C}\right) + \sum_{i=0}^{\sqrt{N_K}-2}(N_E - 4i) + 3\sum_{i=1}^{\sqrt{N_K}-1} i^2 + 4\sum_{i=0}^{y} i\left[\sqrt{N_K} - (i+1)\right] \quad (7)$$

$$\text{where } y = \begin{cases} \sqrt{N_K} & \text{if } (\sqrt{N_K}-1) \bmod 2 = 0 \\ \sqrt{N_K}-1 & \text{if } (\sqrt{N_K}-1) \bmod 2 \neq 0 \end{cases}$$

$$M2_{CRCE} = 8N_K + 3\left(\sqrt{N_K}-1\right)^2 + 2\left(\sqrt{N_K}-1\right) \quad (8)$$

The iterative property of RC-less is yet again evident in R-less, (7). Figure 4 shows the effect this has on the number of transmissions in cluster-based coordinate establishment. RC-less is removed in Figure 5 so that the match between numerical and simulation results could be seen more clearly. Here we note that CRCE scales linearly as the number of clusters in the network increases - a highly desirable attribute. We also note that the simulations enforce the validity of the numerical formulas.

### 5.2.3 Clusterless vs. cluster-based

Up to now, obtaining the message cost in Stage 2 involved varying the density of the nodes in a set area of $1600m^2$, for the clusterless methods. For the cluster-based methods the number of clusters has been varied at a constant density - the size of the cluster-based network being directly proportional to the number of clusters in the network.

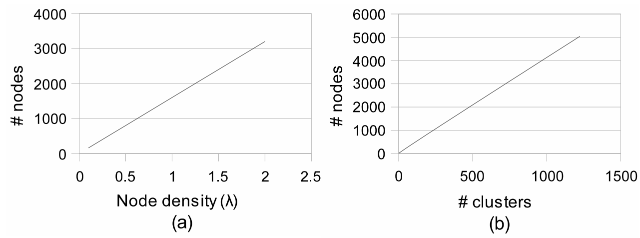


*Figure 6:* Increase in the number of nodes in the network as (a) the node density (clusterless methods) and (b) the number of clusters (cluster-based methods) increase

To compare all four methods of coordinate propagation on one graph, it is necessary to level the plane (establish some common ground). Thus, $\lambda$ of the clusterless methods has to be set to a value that corresponds to the density of the cluster-based methods. The number of clusters is then set to a value that will result in a match in the numbers of nodes between each network at the chosen $\lambda$.

The degree of connectivity in a WSN refers to the average number of neighbours a node (node *i*) has. It has been proven in [17] that a network needs a degree of no less than 6 in order to have complete connectivity with high probability. Consequently we choose a safe degree of connectivity at $D_i = 8$. Therefore the total number of nodes in the domain of node *i* is $N_i = 9$ (including node *i*).

From (4) we can obtain the density of the network by taking $r = 2m$ and $n_I = N_i - 1$. Then $\lambda \approx 0.716\, nodes/m^2$. Figure 6a and 6b show how the number of nodes increases as $\lambda$ and the number of clusters increase respectively. By using $\lambda$ as index on Figure 6a, we find the total number of nodes $N_T = 1146\, nodes$. Indexing with $N_T$ on Figure 6b results in 260 *clusters*.

Now it is possible to compare the number of messages of the clusterless methods at $\lambda \approx 0.716\, nodes/m^2$ to the cluster-based methods with the number of clusters set to 260. Figure 7 presents the results.

The superior performance of the CRCE method is observed. The performance of the C-less method over that of the cluster-based R-less method is also noted. This is due

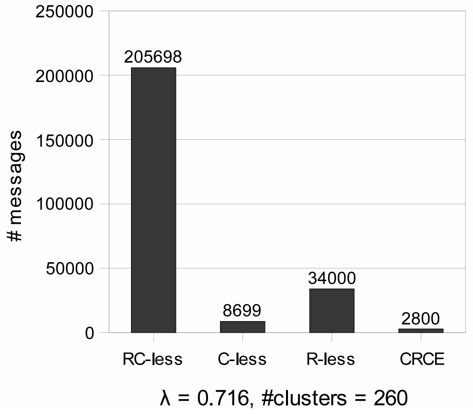


*Figure 7:* Second stage message transmissions

to the inefficient iterative coordinate propagation of the R-less method.

### 5.2.4 Random node placement

The message cost in Stage 2 depends, in part, on the location of the sink node (radial methods) or the location of the node with the lowest ID (non-radial methods). All position coordinates have to be calculated relative to the sink or the node with the lowest ID, leaving that node as the origin of the network-wide coordinate system - we call it the reference node.

Having the origin somewhere in the middle of the network shortens the convergence distance - distance between the reference node and the node furthest from it - as opposed to being located on the edge of the network. This does not affect the number of message transmissions in Stage 1.

Node numbering also influences message cost. It is particularly of interest at the numerical analysis because we need a consistent node numbering scheme in order to derive a formula for message cost calculation.

For this reason, it is important to note that the formulas in Stage 2 were obtained in networks where the nodes were numbered from left to right, top to bottom, with the reference node located at a corner of the network (top left in this case). The same goes for the cluster numbering for the cluster-based methods.

In practice, there will often be no control over the location of the reference node or node numbering conventions.

Simulations were performed to investigate the effect of random node numbering and random reference node placement. They were run 20 times for each method with a different uniformly distributed random placement of nodes for every run. Again, it was necessary to set $\lambda \approx 0.716\ nodes/m^2$ and $N_T = 1146\ nodes$ (260 *clusters*). Transmission radius $r = 2m$.

The mean values for message cost that were obtained from the simulations are displayed in Figure 8.

The variation in message transmissions for each method (except RC-less) is normalised relative to the mean of that method in Figure 9. RC-less is omitted because it scales the graph too small to make any worthwhile comparison between the other - variations of up to 53 000 messages were witnessed for RC-less.

It is clear that node placement in the network influences the message cost of the iterative methods (i.e. the

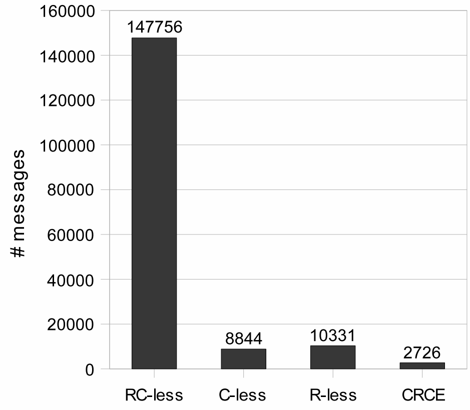


*Figure 8:* Mean values of message cost for random node placement

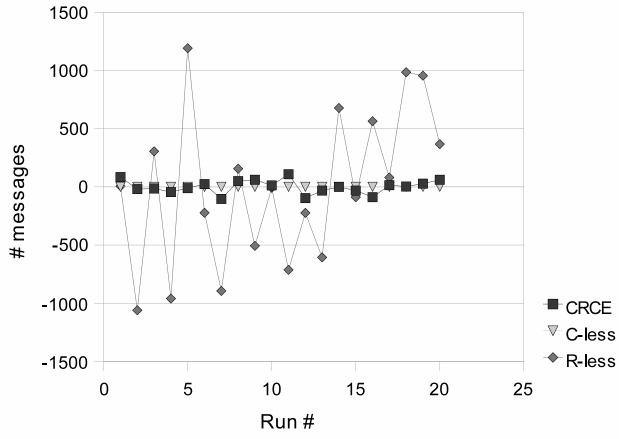


*Figure 9:* Variation of message cost with random placement of nodes in the same network

non-radial methods) much more than their radial counterparts.

There is no variation in the number of transmissions for C-less, proving that the message cost of the radial methods is not influenced by node numbering or sink node location - coordinate updates transverse the entire network only once, independent of the reference node location.

The little variation in CRCE is due to different cluster formations in each run, which is dependant on random timers and not numbering or sink node location.

## 5.3 Message cost: Stage 1 and 2 combined

Finally, an overall comparison (combination of Stage 1 and 2) is made for all four coordinate establishment methods. Yet again we set $\lambda \approx 0.716\ nodes/m^2$, $r = 2m$ and $N_T = 1146$ (260 *clusters*) to obtain a common ground for the clusterless and cluster-based methods regarding node density and number of clusters respectively.

The result is shown in Figure 10. RC-less is omitted from this - it transmits a total of 353 454 messages. CRCE outperforms its closest competitor, C-less, by 68.5%, and R-less by 87.5%.

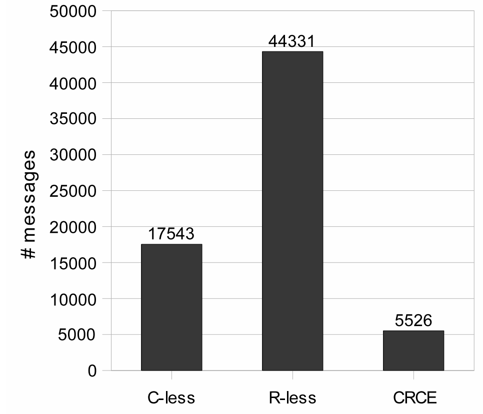


*Figure 10:* Total message cost for the coordinate establishment methods

## 6 CONCLUSION

A cluster-based radial coordinate establishment method (CRCE) was introduced and compared to three other relative localisation methods. It is shown to be the cheapest of the four with regards to the message cost of establishing a network wide coordinate system. It also exhibits a linear scaling ability in Stage 2 and close to linear in Stage 1.

It is interesting to note is that the radial C-less method outperforms the cluster-based R-less method in Stage 2 - a trend that is also reflected in the overall comparison of the localisation methods - and suggests that radial convergence alone is more effective than implementing only clustering in a network. Not only is it more energy efficient but also simpler to implement and less resource intensive.

By implementing radial convergence (proposed in this research) in a cluster-based topology, CRCE is shown to be almost 70% more efficient in terms of message cost reduction than the modified RC-less method, C-less, and close to 90% more efficient than the published R-less method, making it the preferred method for coordinate establishment in energy constrained WSNs.

## REFERENCES


[1] A. Savvides, C.-C. Han and M. B. Strivastava. "Dynamic fine-grained localization in Ad-Hoc networks of sensors". In *MobiCom '01: Proceedings of the 7th annual international conference on Mobile computing and networking*, pp. 166–179. ACM Press, New York, NY, USA, 2001.

[2] T. He, C. Huang, B. M. Blum, J. A. Stankovic and T. Abdelzaher. "Range-free localization schemes for large scale sensor networks". In *MobiCom '03: Proceedings of the 9th annual international conference on Mobile computing and networking*, pp. 81–95. ACM Press, New York, NY, USA, 2003.

[3] A. Nasipuri and K. Li. "A directionality based location discovery scheme for wireless sensor networks". In *WSNA '02: Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, pp. 105–111. ACM Press, New York, NY, USA, 2002.

[4] L. Doherty, K. S. J. Pister and L. E. Ghaoui. "Convex position estimation in wireless sensor networks". In *Proceedings of IEEE INFOCOM '01*, pp. 1655–1663. April 2001.

[5] N. Patwari, A. O. Hero, M. Perkins, N. Correal and R. J. O'Dea. "Relative location estimation in wireless sensor networks". *Submitted to IEEE Trans*. on Sig. Proc., pp. 2137–2148, August 2003.

[6] D. Niculescu and B. Nath. "Position and orientation in ad hoc networks". *Elsevier's Journal of Ad Hoc Networks*, vol. 2, no. 2, pp. 133–51, April 2004.

[7] D. Niculescu and B. Nath. "Localized positioning in ad hoc networks". *Elsevier's Journal of Ad Hoc Networks, Special Issue on Sensor Network Protocols and Applications*, vol. 1, pp. 211–349, September 2003.

[8] A. Erdogan, E. Cayirci and V. Coskun. "Sectoral sweepers for sensor node management and location estimation in ad hoc sensor networks". In *Milcom '03*. October 2003.

[9] J. Riba and A. Urruela. "A non-line-of-sight mitigation technique based on ML-detection". In *Proceedings of ICASSP '04*. May 2004.

[10] S. Mahlknecht and M. Bck. "CSMA-MPS: A minimum preamble sampling MAC protocol for low power wireless sensor networks". In *Proceedings of the International Workshop on Factory Communication Systems*, pp. 73–80. 2004.

[11] S. M. G. M. N. Halgamuge and A. Jennings. "Energy effcient cluster formation in wireless sensor networks". In *ICT '03: Proc. 10th IEEE International Conference on Telecommunications*, vol. 2, pp. 1571–1576. February-March 2003.

[12] A. Wang and A. Chandrakasan. "Energy-Effcient DSPs for Wireless Sensor Networks". *IEEE Signal Processing Magazine*, vol. 19, no. 4, pp. 68–78, July 2004.

[13] R. Iyengar and B. Sikdar. "Scalable and Distributed GPS Free Positioning for Sensor Networks". In *IEEE ICC '03*. Anchorage, AK, May 2003.

[14] S. Capkun, M. Hamdi and J. Hubaux. "GPS-Free Positioning in Mobile ad-hoc Networks". In *HICSS '01: Proceedings of the 34th Annual Hawaii International Conference on System Sciences ( HICSS-34)*, vol. 9, pp. 3481–3490. January 2001.

[15] "Mobility Framework for OMNeT++". http://mobility-fw.sourceforge.net, 9th August 2005.

[16] OMNeT++. "Discreet event simulation system". http://www.omnetpp.org, 9th August 2005.

[17] S. B. W. B. Krishnamachari and R. Bejar. "Phase transition phenomena in wireless ad-hoc networks". In *Proceedings of the Symposium on Ad-Hoc Wireless Networks (GlobeCom2001)*. November 2001.