Hop-Count based Virtual Coordinate Assignment Scheme through Four Beacons for Wireless Sensor Network

Alok Kumar
Indian Institute of Information Technology, Allahabad, India

Shirshu Varma
Indian Institute of Information Technology, Allahabad, India

Rajeev Sonkar
IBM India,
Pune, India

ABSTRACT

In this paper, we propose a simple distributed protocol to construct a virtual coordinate scheme (VCA) based on hop count to four beacon (or reference) nodes. In this proposed protocol each node only maintains the hop count to these known location nodes (beacons) and do not require the physical or real location information. We put the assumption that all nodes are deployed in a rectangular shape area and the sink node (or base station) is located at one of the vertex of this rectangle. The sink node is also used as one of the milestone (beacon) nodes. Three other sensor nodes are chosen as beacon nodes placed close to the vertex of the rectangular area like sensor network in our designed protocol. A hop count to four beacon nodes is contained by the virtual coordinate vector, and by using this virtual coordinate vector; a node can make a decision for greedy routing. The simulation results for proposed protocol shows that the virtual coordinate system can support the standard geographic routing more efficiently than the real coordinate system.

General Terms

Virtual Coordinate System

Keywords

Wireless Sensors Networks, Virtual Coordinates, Geographic Routing.

1. INTRODUCTION

Several applications of wireless sensor networks require the accurate location information of the individual nodes. An open problem of wireless sensor network is the robustness of sensor node localization. Here exist many approaches but they all have some significant drawbacks that limit their accuracy and feasibility to real world problems.

In recent years, the advancement in microprocessors, low power and small size sensing units, are making multifunctional and low cost sensor nodes more available. Several applications require large scale sensor networks such as military operations, biological observations, environment monitoring, and other useful critical applications. Since the majority of applications will generate lots of sensing data, the most important functions of sensor nodes is to collect the large amount of sensing data from the densely deployed distributed sensor networks and send it to the base stations(sink node) in a multi-hop way. For highly distributed large scale sensor network, the challenge for us is the effective and efficient deployment of the sensor network.

A sensor node consists of a processor, a radio transceiver and one or more transducers, and it is powered by a low power embedded battery. With this limited amount of energy and memory of the sensor nodes, the use of caches to store paths is not effective/ realistic. Due to this reason the geographic coordinates of the sensor nodes are used to perform geographic routing through greedy heuristics to forward packets. But the greedy forwarding fails due to the problem of void or dead end. To overcome this dead end problem, the sensor network exploit face traversal protocols. The drawback in exploiting geographic coordinate is the awareness of the physical or real position of every node. The physical or real position information of the sensor nodes could be realized in two ways:

1. Manual configuration of locations which is a difficult task for large scale sensor networks.
2. By using Global Positioning System (GPS) for every node.

But, due to massive scale and application context of sensor nodes, use of GPS is unrealistic. In a huge scale sensor network it is very expensive to equip GPS devices for every sensor node .The sensor nodes will also die quickly because of the heavy consumption of power by the GPS devices.

Some location estimation algorithms that assumes the radio model is ideal, uses the received signal strength indicator technique. But this assumption is false in the real environment. Thus, to assign an accurate location to every sensor nodes is really a big challenging issue in wireless sensor networks. However, the efficient and realistic solutions is the exploitation of few sensor nodes which are equipped with GPS device and with these few sensor nodes, the position of the other sensor nodes could find out.

In this paper, we propose a simple distributed protocol to build a virtual coordinate system (VCS) based on hop count to four beacon (or landmark) nodes. Every node only maintains hop count to these beacon nodes and do not need the real or physical location information. We assume that the nodes are deployed in a rectangular type area and the sink node (or base station) is positioned at one of the vertex of the rectangle. The sink node is also used as one of the beacon nodes. Our designed protocol chooses three other sensor nodes as beacon nodes placed near the vertex of the rectangular area like sensor network. The virtual coordinate vector contains hop count to four beacon nodes, and by using virtual coordinate vector; a node can make greedy routing decision. The simulation results for our protocol shows that the virtual coordinate system can more efficiently support the standard geographic routing than the real coordinate system.

2. RELATED WORK

We now review earlier work in geographic routing and explain the basics of the workings of geographic routing with real as well as virtual coordinate that provides the background for our work.

There have been various geographic routing algorithms, including GFG [1], GPSR [2], and the GOAFR family of algorithms [3,4] and many more, such as [5]-[12] [27]. Algorithms such as GFG/GPSR [1, 2] and GOAFR [13] appear to be extremely scalable and resource-frugal and enable routing without routing table, only at the cost that nodes have to know their coordinates. We are not aiming to improve these algorithms, but the purpose of this work is only to provide a set of virtual coordinate vectors over which they can work efficiently then the real coordinates. For the purposes of performance evaluation, a very simple routing algorithm is used.

To build a coordinate system for wireless sensor and ad hoc network several algorithms are proposed .We can classify those protocols into two categories: first is to find absolute coordinate [14]-[20] [26].The objective of an absolute coordinate system is to find the real or physical location of all the sensor nodes. The second is to find the virtual coordinate [21, 22, 23]. The objective of a virtual coordinate system is the embedding of the sensor nodes into multi-dimensional space so that the neighbor relationship among nodes is same as the underlying sensor network.

The authors in [22] proposed connectivity-based approaches. Extending connectivity-based approaches to location problems has been discussed in [20] .The inter-node distance measurements have been integrated into the computation of virtual coordinates [20, 24]. These are centralized approaches which are not feasible for large scale wireless sensor network.

Hop counts with connectivity based approaches are proposed by several researchers. The distributed approach in [25] can build a coordinate system to support geographic routing efficiently, but the memory cost and communication overhead are quite high. In [23], the author proposed a distributed protocol called VCap to determine three landmark nodes. A considerable flooding overhead can be reduced by this technique, but it requires the entire network to be synchronized, which is a problem for real sensor networks. In this paper we proposes a simple distributed algorithm to locate beacons which are placed at the corner boundaries of a wireless sensor network and to allot each node a virtual coordinate vector in the network. The VCap protocol [23] is also based on hop count metric but chooses three beacon nodes to define a virtual coordinate system.

3. BASICS OF VIRTUAL COORDINATE SYSTEM

The concept of virtual coordinate system is that every node maintains hop counts to beacon (or landmark) nodes. These hop counts form a vector, called virtual coordinate vector. For example, as shown in figure 1, four beacon nodes, A, B, C, D and 20 nodes are deployed in the sensor network. Each beacon node produces a control packet containing a hop counter and its own ID. Every node determines a hop count to all beacon nodes and acquires the virtual coordinate vector (a, b, c, d), by flooding this control packet to the whole network. The virtual coordinate vector of node X is (6, 2, 5, 3). It simply means that node X is six hops away from beacon node A, two hops away from beacon node B, five hops away from beacon node C, and three hops away from beacon node D.

The features of virtual coordinate system are: First, in the sensor network graph, virtual coordinate can reflect the true connectivity among sensor nodes, rather than real distance. Second, for virtual coordinate vector, the dimensionality is decided by the number of beacon nodes. Increasing the number of beacon nodes can increases the flooding overhead but it would also improves robustness of the sensor network. Third, when the network density is high, virtual coordinates propagate as circular coronas which are centered on the initiator beacon node. As shown in Figure 2, sensor nodes with first hop centered on landmark B looks like a circle and its radius is equal to the communication range. Sensor nodes with second hop also looks like a circular corona centered on beacon node B, and the radius of this circular corona is equal to the sum of first hop plus the communication range. Fourth, it is necessary that the number of beacon nodes must be greater than two.

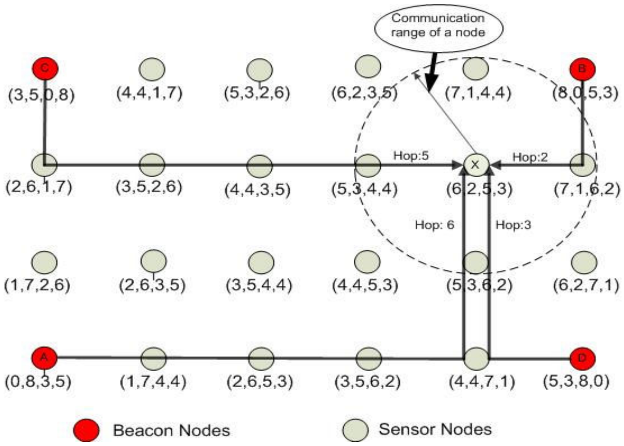

Figure 1: A Logical Coordinate System

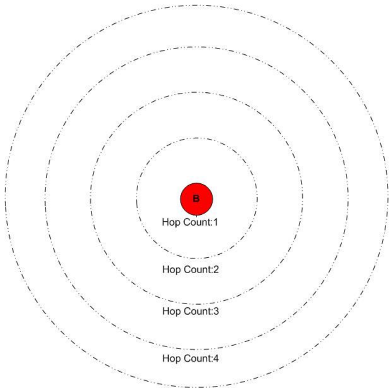

Figure 2: Propagation of hop counts as circular coronas centered on beacon node B

If only two beacon nodes are deployed in the sensor network, there would exist a situation that the two beacon nodes may have the same virtual coordinate vector. The reason is that, there may exist zones which are symmetric to the directrix connecting the two beacon nodes. In Figure 3, sensor nodes in X and Y zones share the same virtual coordinate. If the destination is in zone X, the packet may be routed to zone Y. As shown in Figure 4, one more beacon node B is added to the network. Now zones X and Y can be identified separately.

Fifth, every sensor node does not have a unique virtual coordinate vector. Same virtual coordinate may be shared by a node with its neighbors in the same zone. For example, as shown in Figure 4, sensor nodes in zone X share the same virtual coordinate (2, 4, 4).

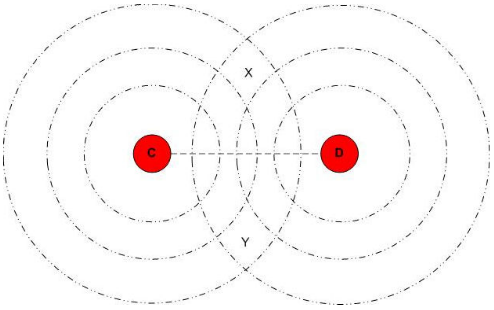

Figure 3: Same virtual coordinate is shared by sensor nodes in zones X and Y

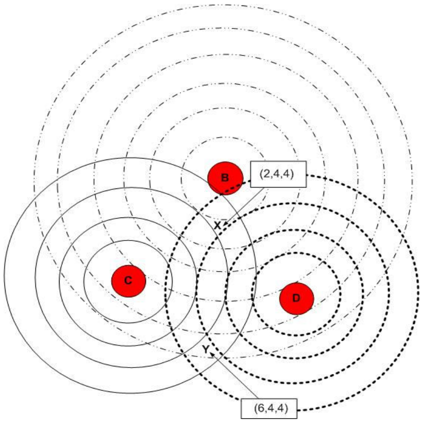

Figure 4: By using a third beacon node B, Zones X any Y can be identified separately

For applying geographic routing over the virtual coordinate system, one node selects the neighbor whose virtual distance from destination is least, as the relay node. The virtual distance (VD) between two virtual coordinate vectors X and Y is defined as;

$$VD = \sqrt{\sum_{i=1}^{n} \left(X_i - Y_i\right)^2}$$

where X_i and Y_i are elements in vectors X and Y, respectively. Every sensor node needs to maintain its neighbors' virtual coordinate for selecting the best relay node greedily.

4. VIRTUAL COORDINATE ASSIGNMENT PROTOCOL

A VCA protocol is used to select four beacon nodes which are positioned near the corners of the sensor network. Using more than four beacon nodes only brings limited improvement and will incur more communication overhead. Therefore, VCA protocol is designed to find four beacon nodes which are located as near the corners of the network as possible. Each sensor node in the network will then be assigned a four-dimensional virtual coordinate vector without any real location information. So, without the help of GPS devices, a virtual coordinate system based on hop counts to the beacon nodes can be established and it can support geographic routing efficiently

VCA protocol consists of four phases: $A-Phase$, $B-Phase$, $C-Phase$, and $D-Phase$.

In the first phase, $A-Phase$, we treat the sink or base node as beacon node A, and deploy it at one of the corner boundary of the sensor network. So once the network is deployed, we can decide the placement of the base node by our will. Initially, Beacon node A will generate a A_msg packet and then broadcast this message to its every neighbor; the A_msg packet consists of a hop counter (initially set to zero), its ID, and $A\_threshold$. The $A\_threshold$ is used to select beacon node A in the next phase of this scheme. When a node receives the A_msg packet, then it increments the hop count by one hop and after that it rebroadcasts the packet to its neighbors. Every sensor node will keep the information of the least hop count packet while receiving multiple A_msg packets. At the end of $A-Phase$, each node will be assigned a hop count to beacon node A, called A coordinate.

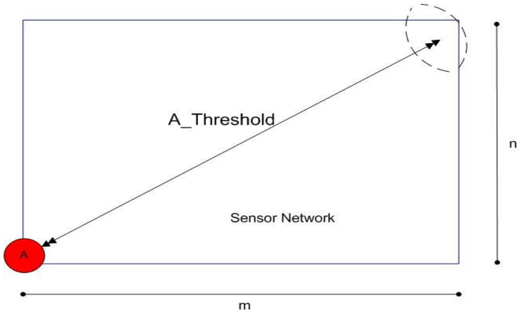

Figure 5: The definition of $A\_Threshold$

In the second phase, $B-Phase$, a node will be selected as beacon node B which is the farthest node to beacon node A in the sensor network. Now let us assume that a sensor network is bounded in a rectangle type area and the side lengths of the rectangle are n and m for short and long side, respectively. In Figure 5, the parameter, $A\_threshold$, represents the minimum hop counts from beacon node A to its diagonal of the sensor network. That is,

$$A\_Threshold = \left(m^2 + n^2\right)^2 / Tx\_Range$$

where transmission range of the radio is represented by Tx_Range.

In $B-Phase$, each node will first broadcast its A coordinate to one-hop neighbors if its A coordinate is equal to or larger than $A\_threshold$. Then every node can find out whether it is a candidate of beacon node B or not. A node will become a candidate if its A coordinate is maximum within one-hop neighbors. Note that, if two sensor nodes have the same A coordinate value, we select the node with smaller ID as a candidate. Since the candidate is decided locally by every sensor node, there may exist two or more candidate of beacon node B. Thus, every candidate node will flood a A_local_msg control packet to the network and find out one of the candidates as beacon node B. The control packet includes candidate node's ID, A coordinate, and a time to live (TTL). Since the A coordinate of any candidate is equal to or larger than $A\_Threshold$, the control packets only need to forward to the sensor nodes whose $coordinate \geq A\_Threshold$

Taking figure 6 as an example, we assumed that the $A\_Threshold$ is 14 and nodes 65 and 55 are the candidates of beacon node B. Both nodes 55 and 56 will flood A_local_msg packet to the network. Any node with A coordinate smaller than $A\_Threshold$ will drop the received control packet. So, if the packet's A value is smaller than one of the previously received value then each node will also drop the received control packet. These local flooding results in a great reduction of a huge number of control packets overhead in $B-phase$. After a predetermined time period T_X, the sensor node with maximum A value will find that it is the beacon node B, where T_X is equal to $TTL \times t$ (t is the time needed to broadcast a packet from a node to its neighbors).

$TTL = (A\_Threshold/2)$ is enough to obtain good results. The selected beacon node B will flood an B_msg control packet including a hop counter (initial set to zero), its ID and w value to the whole sensor network. Every sensor node will obtain its B coordinate from the control packet. For example, In Figure 6, after ending the local flooding, sensor node 55 will consider itself as the beacon node B. After executing the $B-Phase$, each node can obtain its A and B coordinates. The value of $A+B$ of sensor nodes near the center of the sensor network are smaller than those near the corners of the sensor network as shown in figure 7.

In the third phase, $D-Phase$, we would like to select the beacon node C located in the lower- right corner or upper-left corner of the sensor network. Hence, the possible candidates of the next beacon node are positioned in a banding zone of the sensor network from upper-left corner to lower-right corners. We can define this banding zone as "a set of sensor nodes in which their coordinates A and B full fill the following conditions: $A = B+1$ or $A = B$ or $A = B-1$. For example, we randomly deploy 500 sensor nodes in a $1000 \times 1000m$ sensor network; A banding zone is shown in Figure 8. Each node belonging to the banding zone will broadcast its coordinate to one-hop neighbors. A node will become a candidate of beacon node C if its value of $B \times A$ is maximum among its one-hop neighbors. Like in $B-Phase$, there are more than one C candidates in the banding zone. For example, in Figure 9, sensor nodes 45, 152, 331, or 483 are candidates of the beacon node C. To select one of them as the beacon node, each candidate floods a C_local_msg control packet containing its ID, A, and B coordinates to the network. Sensor nodes located in the banding zone will rebroadcast the control packets to the sensor network. When every sensor node receives a C_local_msg packet, the sensor node will drop the control packet if the packet's $A+B$ value is smaller than the previously received one. If two sensor nodes have the same $A+B$ value, then we can break the tie by the use of node ID.

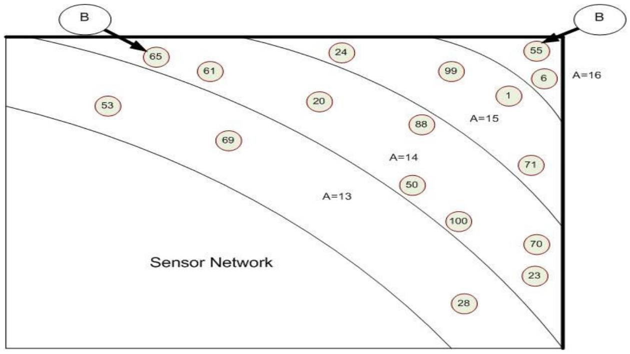

Figure 6: Example: A local flooding for $A\_Threshold = 14$

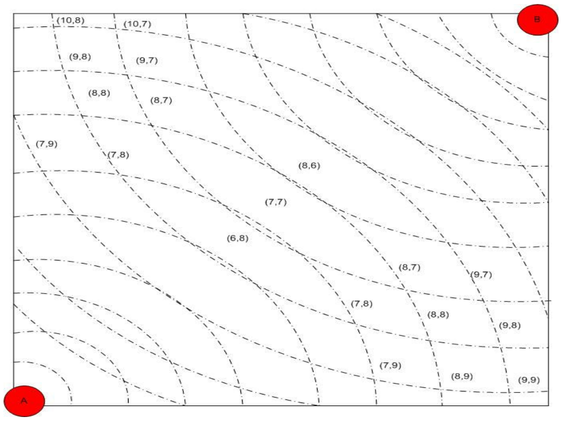

Figure 7: Result after A and B phases is ended

After a predetermined time T_C, the node with the maximum $A+B$ will claim that it is the beacon node C. Note that, $Tc >= C\max * t$, where $C\max$ is the A value of beacon node B in the sensor network. Then the beacon node will flood a control packet C_msg containing a hop counter (initially set to zero) and its ID to notify all the sensor nodes. In Figure 9, node 483 will become the beacon node Y since its sum of A and B is the maximum in the banding zone.

The last phase, $D-Phase$, the beacon node D is the farthest node to the beacon node C. Thus, the candidates of the beacon node D are located in the same banding zone with beacon node C. When a sensor node receives the C_msg packet, then that sensor node has a hop count to the beacon node C. Each node in the banding zone broadcasts its C coordinate to one-hop neighbors. The node that has the maximum C value among its one-hop neighbors becomes the candidate of the beacon node. Note that, if two sensor nodes have the same C coordinate value, then we can break the tie by the use of node ID. Likewise, every candidate node will flood a control packet D_local_msg containing its ID, C coordinate, and a TTL. $TTL = 2$ is enough.

Each candidate node waits $2t$ time periods to determine whether it is the beacon node D or not. Then the beacon node D floods a control packet D_msg containing a hop counter (initially set to zero) and its ID to notify all the sensor nodes and each sensor node can get a D coordinate.

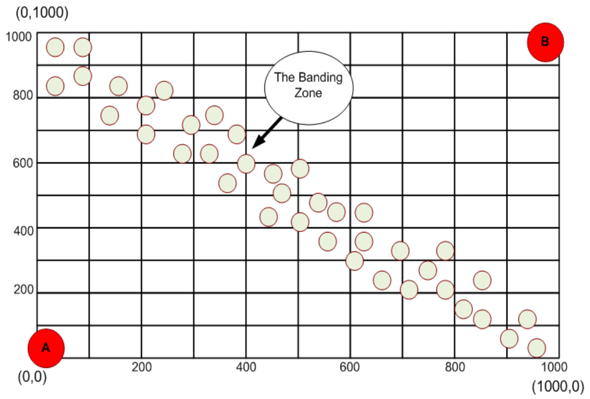

Figure 8: Example: Distribution of sensor nodes which satisfies $A = B + 1$, $A = B$ or $A = B - 1$

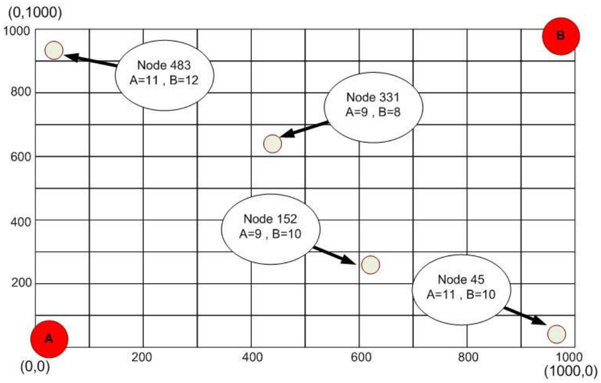

Figure 9: Candidates of C are sensor nodes 45, 152, 331 and 483

Finally, the beacon nodes A, B, C, and D every sensor node can acquire a virtual coordinate vector which includes A, B, C, and D coordinates. Although few sensor nodes might not be able to receive the control packets from some beacon nodes, they can fill their virtual coordinate vector by exchanging some coordinate data with their one-hop neighbors. Then, the virtual coordinate system is constructed completely and geographic routing can be applied to the system. Now by simulation we will show the performance of this system.

5. SIMULATION STUDY

In this portion, the performance of this proposed protocol (VCA) is evaluated through simulations. The implementation of our protocol is on a tool named as NCTUns, which is a discrete event simulator, named as NCTUns. It is our assumption that the sensor nodes are static and are dispersed uniformly in a $1000 \times 500m$ rectangle area. The transmission range of every node is same and it is 100 m for each node. The propagation delay is 1 second. The total number of sensor nodes is in the range of 150 to 900. The simulation results have been calculated as the average of 100 propagations. For our simulations we do three experiments which are as follows:

1st, the packet delivery ratio of greedy forwarding was measured in the virtual coordinate system built by our VCA protocol, (without using time trading technique).While using accurate real coordinate system for sensor nodes we also evaluated the routing performance. In our simulation, we only want to show the superiority of the virtual coordinate system and hence we do not use any backtracking to improve the routing performance.

2nd, the average path length was measured with the virtual coordinate system constructed by VCA and evaluate the real coordinate system.

3rd, to accomplish the purpose of studying the communication overhead we compare the completion time of VCap and VCA with and without the time trading technique and flooding overhead (in number of packets).

5.1 Packet Delivery Ratio

Based on virtual and real coordinate systems, we apply simple geographic routing in our simulations. We choose 100 pairs of sources and destinations randomly. In order to avoid so many routing pairs at the same time and to decrease the probability of collisions, we control the time which every source node generates a routing packet. It helps us to determine the packet delivery ratio more accurately. Figure 10 shows the simulation result, while the average number of neighbors of every node (network density) increases, the reachability of the four real and virtual coordinate systems also increases and it almost reaches to 100%. Our virtual coordinate system (VCA) has more than 90% reachability. We found that when the network density is low, the virtual coordinate system performs better than the real coordinate system. This is due to reason that the real connectivity of nodes can be reflected by virtual coordinate.

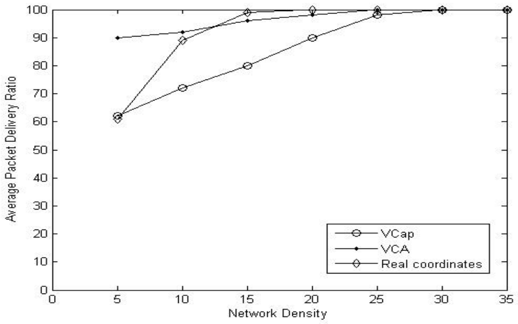

Figure 10: Average packet delivery ratio with network size $1000 \times 500m$.

5.2 Average Path Length

In this portion, we evaluate the average path length of both the coordinate systems. Our simulation result is shown in figure 11, shows that the average path length of four virtual coordinate systems is 10% more than the real coordinate systems. It happens because the sensor nodes can get more accurate information of destination node while routing for real coordinate. With the increase in network density, the average path lengths of VCA are shorter than that of real coordinate system.

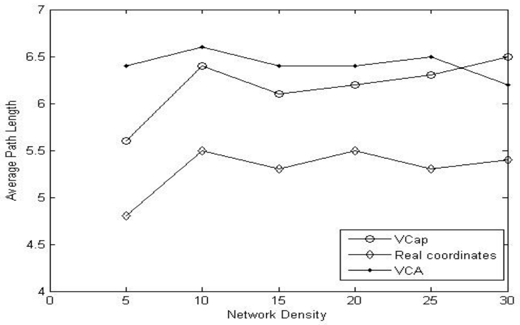

Figure 11: Average Path Length with network size $1000 \times 500m$

5.3 Flooding Overhead

In this portion, we evaluate the flooding overhead of VCap and VCA with and without time trading technique. Flooding overhead is calculated by the total number of control packets sent during simulation period. From figure 12, with time trading technique VCap needs less flooding packets than our protocol VCA. So, VCA needs some more control packets than VCap with time trading. This is due to the reason that last three phases, our protocol VCA has additional local flooding. In our observation, the cost of banding zone and local flooding is acceptable. As we expected that when the number of nodes increases, VCA without time trading technique needs much less flooding overhead than VCap. The only reason is that, in the last three phases of VCap protocol several global flooding occurred without time trading technique.

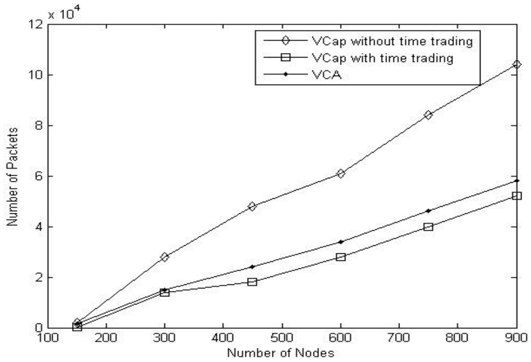

Figure 12: The number of control packets with network size $1000 \times 500m$

5.4 Execution Time

Finally, we calculate the execution time of VCap and VCA. Specifically, we calculate the time of VCap with and without time trading technique. From figure 13, the execution time of VCap is slightly lower than our VCA protocol without time trading technique. Figure 14, shows the execution time of VCA and VCap with time trading technique. We observe that as we increases the number of nodes in the sensor network, the execution time of VCap will also increases with time trading technique. Our VCA protocol takes much lesser time than VCap with time trading technique. The execution time of our protocol VCA is only affected by the network size of the sensor network.

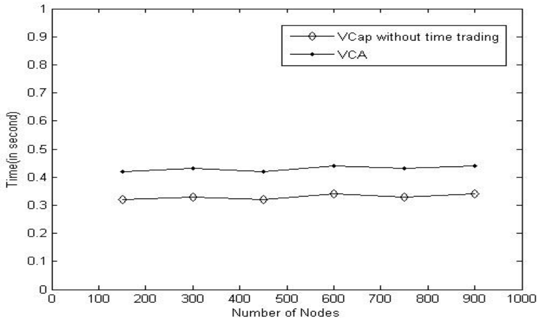

Figure 13: The execution time with network size of $1000 \times 500m$

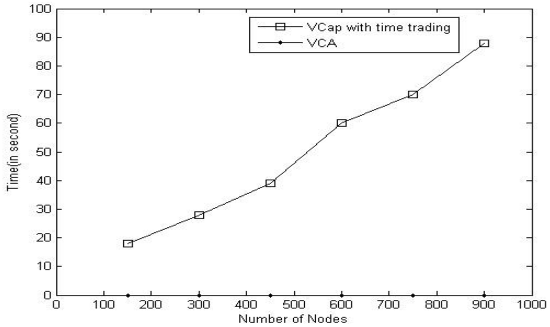

Figure 14: The execution time with network size of $1000 \times 500m$

6. CONCLUSION

In this work, we present a simple scheme to construct a virtual coordinate system which is based on hop counts to beacon nodes. Routing decision is based on the virtual coordinates of the sensor nodes, so no real coordinates is used in making routing decision. The main idea of VCA protocol is to discover four beacon nodes and each of these nodes is at the corner of the network .With these four sensor nodes, a virtual coordinate vector is assigned to every sensor node. With our virtual coordinate system the packet delivery ratio is approx. 100 percent and it is much closer to real coordinate system. The virtual coordinate system has longer average path length than the real coordinate system. When network density is low, the routing performance of VCA protocol is better than that of VCap. In addition, our VCA protocol has lower communication overhead than VCap without time trading technique. But with time trading technique VCap has a lower communication overhead than VCA. Also, with time trading technique, VCap has higher execution time than VCA.

7. REFERENCES

[1] Bose, P., Morin, P., Stojmenovic, I., and Urrutia, J., ‘‘Routing with Guaranteed Delivery in Ad hoc Wireless Networks”. In Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIAL-M), Seattle, WA, August, 1999.

[2] Karp, B., and Kung, H. T., “GPSR: greedy perimeter Stateless Routing for wireless networks”. The Sixth Annual ACM/IEEE MobiCom, Boston, MA, August, 2000.

[3] Kuhn, F., Wattenhofer, R., Zhang, Y., and Zollinger, A., “Geometric ad-hoc routing: Of theory and practice”. 22nd

ACM Symposium on Principles of Distributed Computing (PODC), Pp. 63-72, Boston, Massachusetts, USA, July 2003.

[4] Kuhn, F., Wattenhofer, R., and Zollinger, A. "Worst-Case Optimal and Average-Case Efficient Geometric Ad-Hoc Routing". The 4th ACM Interational Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc), Pp. 267-278, Annapolis, MD, June 2003.

[5] Basagni, S., Chlamtac, I., Syrotiuk, V., and Woodward, B., "A distance routing effect algorithm for mobility (DREAM)". The Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom), Dallas, Texas, August 1998.

[6] Beijar, N., "Zone Routing Protocol (ZRP)". Licentiate course on Telecommunications Technology, Ad Hoc Networking, April 2002.

[7] Bose, P., Morin, P., Stojmenovic, I., and Urrutia, J., "Routing with Guaranteed Delivery in Ad Hoc Wireless Networks", The 3rd international workshop on Discrete algorithms and methods for mobile computing and communications Wireless Networks, Vol. 7, Pp 609-616, 2001.

[8] Gregory, G., Finn. "Routing and addressing problems in large metropolitan-scale intemetworks". Technical Report, ISI/RR-87-180, ISI, March 1987.

[9] Gao, J., Guibas, L. J., Hershburger, J., Zhang, L., and Zhu, A., "Geometric Spanner for Routing in Mobile Networks", The 2nd ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc), pp. 45–55, October 2001.

[10] Imielinski, T., and Navas, J., "GPS-Based Addressing and Routing", RFC-2009, IETF, November 1996.

[11] Li, J., Jannotti, J., Douglas S. J. De Couto, Karger, D.R., and Morris R., "A Scalable Location Service for Geographic Ad Hoc Routing", The 6th MobiCom, Boston, MA, August, 2000.

[12] Liao, W., and Sheu, J., and Tseng, Y., "GRID: A Fully Location-Aware Routing Protocol for Mobile Ad Hoc Networks", In Telecommunication Systems, Volume 18, Pp 37–60, 2001.

[13] Kuhn, F., Wattenhofer, R., Zhang, Y., and Zollinger, A. "Geometric Ad-Hoc Routing: Of Theory and Practice". The 22nd ACM Symposium on Principles of Distributed Computing (PODC 2003), Boston, MA, July, 2003.

[14] Priyantha, N. B., Chakraborty, A., and Balakrishnan, H., "The cricket location support system'', The 6th ACM/IEEE MobiCom, pp. 32-43. Boston, MA, August, 2000.

[15] Niculescu, D., and Nath, B., ''Ad hoc positioning system (APS)'', The IEEE Global Telecommunication Conference (GLOBECOM), San Antonio, TX, pp. 2926-31, November, 2001.

[16] Savvides, A., Han C., and Strivastave, M.B., ''Dynamic fine-grained localization in ad-hoc networks of sensors'', The 7th Annual ACM/IEEE MobiCom, pp. 166-79, Rome, July, 2001.

[17] Nasipuri, A., and Li K., ''A directionality based location discovery scheme for wireless sensor networks'', The First ACM International Workshop on Wireless Sensor Networks and Applications (WSNA), pp. 105-11, Atlanta, GA, September 2002.

[18] Bahl, P., and Padmanabhan, V.N., ''RADAR: an in-building RF-based user location and tracking system'', The 19th INFOCOMM, Vol. 2, pp. 775-84, Tel-Aviv, Israel, March, 2000.

[19] He, T., Huang, C., Blum, B., Stankovic, J.A., and Abdelzaher, T., ''Range-free location schemes in large sensor networks'', The Ninth Annual ACM/IEEE MobiCom, pp. 81-95, San Diego, CA, September 2003.

[20] Shang, Y., Rum, W., Zhang, Y. and Fromherz, M., ''Localization from mere connectivity'', The Fourth International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2003), pp. 201-12, Annapolis, MD, June, 2003.

[21] Filardi, N., Caruso, N., and Chessa, S., ''Virtual Naming and Geographic Routing on Wireless Sensor Networks'', The IEEE Symposium on Computers and Communications (ISCC'07), pp. 609-614, Aveiro, Portugal, July, 2007.

[22] Moscibroda, T., O'Dell, R., Wattenhofer, M., and Attenhofer, R., ''Virtual coordinates for ad hoc and sensor networks'', The ACM Joint Workshop on Foundations of Mobile Computing (DLALM-POMC), pp. 8-16, Philadelphia, PA, October 2004.

[23] Caruso, A., Chessa S., De S., and Urpi A., ''GPS free coordinate assignment and routing in wireless sensor networks'', The IEEE International Conference on Computer Communication (INFOCOM), pp. 150-160, Miami, FL, March, 2005.

[24] Doherty L., Ghaoui L. E., and Pister K., ''Convex Position Estimation in Wireless Sensor Networks". The 20th INFOCOM, pp. 1655-1663, Anchorage, AL, April, 2001.

[25] Rao A., Papadimitriou C., Shenker S., and Stoica I., ''Geographic routing without location information'', The Ninth Annual ACM/IEEE MobiCom, 96-108, San Diego, CA, September 2003.

[26] Madhulika, Kumar, A. and Varma, S., "Iterative and Distributed Range-free Localization Algorithm for Wireless Sensor Networks" In proceedings of IMPACT-2009, Aligarh, March 2009.

[27] Kumar, A. and Varma, S., "Geographic Node-Disjoint Path Routing for Wireless Sensor networks", IEEE Sensors Journal, Vol. 10, no. 6, pp 1135-36, June 2010.