



**INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH  
TECHNOLOGY**

**WIRELESS SENSORS NETWORKS OVERCOMING SHORTEST PATHS  
PROBLEMS USING DIJKSTRA'S ALGORITHM**

**NITIKA<sup>\*1</sup> & Mr. Parveen Khanchi<sup>2</sup>**

(H.O.D. and Assistant Professor)

Department of Electronics & Communication Engineering, Bahra Institute of Management and  
Technology, V.P.O. Chidana, Gohana

DOI: 10.5281/zenodo.1135466

Wireless sensor networks consist of nodes among which beaconing of data occurs through routing. There are a number of Multicast routing protocols and algorithms existing based on different criteria. There are a number of shortest path algorithms available, some of which are applicable in case of shortest path routing in wireless sensor networks. Shortest path Multicast routing algorithms aim at consumption of minimum amount of energy in a WSN. Generally, Dijkstra's Algorithm is followed for routing through shortest path in a WSN. The Floyd-Warshall's Algorithm is again used for computing shortest paths between different nodes in an ordinary graph but this algorithm is not exactly applicable for routing in Wireless sensor network comprises of a set of sensor nodes that communicate among each other using wireless links and work in an open and distributed manner due to which wireless sensor networks are highly prone to attacks. This is difficult to determine the position of the sensor nodes; therefore the sensor network protocols must inculcate self organizing competence. location awareness is one of the important concern in WSN because for a network mostly data collection is grounded on location, so this is imperative for all the nodes to know their position whenever it is required and it is also helpful in calculating the distance between two particular nodes to deal with energy consumption issues. wireless networks because of the absence of handshaking mode. In this work, the Floyd- Warshall's Shortest Path Algorithm has been modified and a new algorithm has been proposed for routing in the wireless sensor networks. The proposed algorithm computes the shortest path available taking into consideration a directed graph and presence of acknowledgement path of every traversed path.

A sensor network comprised of sensing (measuring), computing, and communication elements that gives an administrator the ability to instrument, observe, and react to events and phenomena in a specified environment. The administrator typically is a civil, governmental, commercial, or industrial entity. The environment can be the physical world, a biological system, or an information technology framework. Network sensor systems are seen by observers as an important technology that will experience major deployment in the next few years for a plethora of applications, not the least being national security. Typical applications include, but are not limited to, data collection, monitoring, surveillance, and medical telemetry. In addition to sensing, one is often also interested in control and activation. There are four basic components in a sensor network:

- (1) an assembly of distributed or localized sensors;
- (2) an interconnecting network (usually, but not always, wireless-based);
- (3) a central point of information clustering; and
- (4) a set of computing resources at the central point (or beyond) to handle data correlation, event trending, status querying, and data mining. In this paper, the sensing and computation nodes are considered part of the sensor network; in fact, some of the computing may be done in the network itself. In the MATLAB simulation, a random network of 50 nodes was created and Dijkstra's algorithm was used to find the routes between anchors . This paper proposes a Dijkstra's algorithm which uses the connectivity of information, the estimated distance information



among the sensor nodes and find out the Shortest Path Position Estimation between Source and Destination nodes in Wireless Sensor Networks with Low Cost.

**KEYWORDS:** Dijkstra's algorithm, Wireless Sensors Networks, Shortest path, Low cost

## I. INTRODUCTION

Dijkstra's algorithm is invented by Dutch computer scientist Edsger Dijkstra in 1956 and published in 1959, is a graph based searching algorithm that solves the single source shortest path problem. It is applied only on positive weights graphs. This algorithm is often used in routing. Dijkstra's algorithm is used for finding the shortest path with minimum cost.

Wireless Sensor Networks (WSNs) are a research topic of growing interest over the recent years due to its wide applications. WSNs are a set of wireless embedded devices that have the capability of processing and communicating video and audio streams collected from the environment in a distributed fashion [1]. WSNs find applications in surveillance systems against crime and terrorist attacks. They can also be used for traffic monitoring in cities and highways. They are also very useful in military applications to locate the targets of interest (such as enemy soldiers, tanks) in the battlefield. For most WSN applications, it is important to have the knowledge of the location of the nodes in order to understand the data received. Therefore, there is a great need to develop a sensor node multidimensional scaling algorithm which can be computational resources [2]. This paper proposes a Dijkstra's algorithm using the network connectivity information and the estimated distance information among the sensor nodes and find out the shortest path between the source node and destination node with low cost. Based on the multidimensional scaling (MDS) technique [3, 6] we derive node locations to fit the roughly estimated distances between pairs of nodes. It uses the Shortest Path Position Estimation between Source and Destination nodes in Wireless Sensor Networks with Low Cost.

In recent years, researchers have been developing different sensor localization algorithms for wireless sensor networks. One kind of technique is based on inter-node distance ranging. To measure the distances, there are two basic techniques: received signal strength [5] and signal propagation time [9].

Dijkstra's multi routing algorithm is invented by Dutch computer scientist Edsger Dijkstra in 1956 and published in 1959, is a graph based searching algorithm that solves the single source shortest path problem. It is applied only on positive weights graphs. This algorithm is often used in routing. Dijkstra's algorithm is used for finding the shortest path with minimum cost.

Wireless Sensor Networks (WSNs) are a research topic of growing interest over the recent years due to its wide applications. WSNs are a set of wireless embedded devices that have the capability of processing and communicating video and audio streams collected from the environment in a distributed fashion. WSNs find applications in surveillance systems against crime and terrorist attacks. They can also be used for traffic monitoring in cities and highways. They are also very useful in military applications to locate the targets of interest (such as enemy soldiers, tanks) in the battlefield. For most WSN applications, it is important to have the knowledge of the location of the nodes in order to understand the data received. Therefore, there is a great need to develop a sensor node multidimensional scaling algorithm which can be computational resources. This paper proposes a Dijkstra's algorithm using the network connectivity information and the estimated distance information among the sensor nodes and find out the shortest path between the source node and destination node with low cost. Based on the multidimensional scaling (MDS) technique we derive node locations to fit the roughly estimated distances between pairs of nodes. It uses the Shortest Path Position Estimation between Source and Destination nodes in Wireless Sensor Networks with low Cost.

**Objectives**

- Improve the performance of distance vector algorithm.
- Making distance vector algorithm dynamic and randomly it reduced the increased complexity of the algorithm.
- Avoid loop formulation and malicious attacks.
- Improve the space and time complexity of dijkstra's and bellman-ford algorithm.
- Improve the space complexity for random generated graph in dijkstra's algorithm.
- Improve the run time complexity of dijkstra's, bellman-ford and distance vector algorithm.
- It give accuracy to the wireless sensor network.

**II. PREVIOUSs WORK**

A Square method is used to estimate all sensors' relative locations by applying MDS to compute the relative positions of sensors with high error tolerance. In order to collect some of pair wise distances among sensors, we select a number of source sensors, and they initialize the whole network to estimate some of the pair. The multidimensional scaling (MDS), a technique widely used for the analysis of dissimilarity of data on a set of objects, can discover the spatial structures in the data. It is used it as a data-analytic approach to discover the dimensions that underlie the judgments of distance and model data in a geometric space. The main advantage in using the MDS for position estimation is that it can always generates relatively high accurate position estimation even based on limited and error-prone distance information. There are several varieties of MDS. On classical MDS and the iterative optimization of MDS, the basic idea of which is to assume that the dissimilarity of data are distances and then deduce their coordinates. More details about comprehensive and intuitive explanation of MDS are available in [1-4].

The multidimensional scaling technique, which is a technique that has been successfully used to capture the intercorrelation of high dimensional data at low dimension in social science, is used [1]. A Square method is used to estimate all sensors' relative locations by applying MDS to compute the relative positions of sensors with high error tolerance. In order to collect some of pair wise distances among sensors, we select a number of source sensors, and they initialize the whole network to estimate some of the pair. The multidimensional scaling (MDS), a technique widely used for the analysis of dissimilarity of data on a set of objects, can discover the spatial structures in the data. It is used it as a data-analytic approach to discover the dimensions that underlie the judgments of distance and model data in a geometric space. The main advantage in using the MDS for position estimation is that it can always generates relatively high accurate position estimation even based on limited and error-prone distance information. There are several varieties of MDS. On classical MDS and the iterative optimization of MDS, the basic idea of which is to assume that the dissimilarity of data are distances and then deduce their coordinates.

**III. PROPOSEDS METHOD**

The Shortest path method is used to estimate all sensors' relative locations by applying Dijkstra's algorithm to compute the relative positions of sensors with low cost and high error tolerance. The routing algorithm is stored in the router's memory. The routing algorithm is a major factor in the performance of position estimation and distance measurement in the sensor field. The purpose of the routing algorithm is to make decisions for the router concerning the best paths for estimating the positions of anchors. The router uses the routing algorithm to compute the path that would best serve to find out the shortest path between the source and destination.

Let the hub at which we are beginning be known as the underlying hub. Dijkstra's calculation will allot some initial distance values and will attempt to enhance them well ordered. For the principal cycle, the present crossing point will be the beginning stage and the separation to it will be zero. For consequent cycles (after the primary), the present crossing point will be the nearest unvisited convergence to the beginning stage—this will be anything but difficult to discover. From the present convergence, overhaul the separation to each unvisited crossing point that is straightforwardly associated with it. This is finished by deciding the whole of the separation between an unvisited crossing point and the estimation of the present convergence and relabeling the unvisited convergence with this esteem on the off chance that it is not as much

as its present esteem. In actuality, the crossing point is relabelled if the way to it through the present convergence is shorter than the already known ways.

### Dijkstra's algorithm

The router builds a graph of the network. Then it identifies source and destination nodes, for example R1 and R2. The router builds then a matrix called the "adjacency matrix." In the adjacent matrix, a coordinate indicates weight. [i, j], for example, is the weight of a link between nodes Ri and Rj. If there is no direct link between Ri and Rj, this weight is identified as "infinity."

### Procedure for finding the shortest path using Dijkstra's algorithm is as follows:

1. The router then builds a status record for each node on the network. The record contains the following fields:
  - Predecessor field - shows the previous node.
  - Length field - shows the sum of the weights from the source to that node.
  - Label field - shows the status of node; each node have one status mode: "permanent" or "tentative."
2. In the next step, the router initializes the parameters of the status record (for all nodes) and sets their label to "tentative" and their length to "infinity".
3. During this step, the router sets a T-node. If R1 is to be the source T-node, for example, the router changes R1's label to "permanent." Once a label is changed to "permanent," it never changes again.
4. The router updates the status record for all tentative nodes that are directly linked to the source T-node.
5. The router goes over all of the tentative nodes and chooses the one whose weight to R1 is lowest. That node is then the destination T-node.
6. If the new T-node is not R2 (the intended destination), the router goes back to step 5.
7. If this node is R2, the router extracts its previous node from the status record and does this until it arrives at R1. This list of nodes shows the best route from R1 to R2 as shown in Fig.1

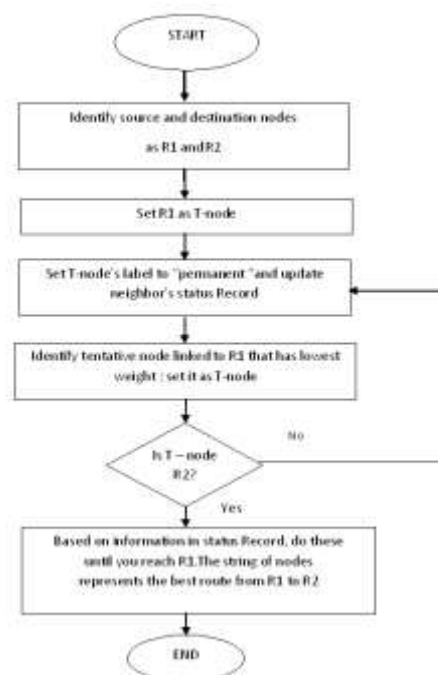


Fig.1 Flow Chart to find shortest path using Dijkstra's algorithm

#### IV. PERFORMANCEs MEASUREMENTs OFs WSNs USINGs DIJKSTRAs ALGORITHM

Wireless sensor network is a collection of ad-hoc networks and many mobility sensor nodes. Wireless sensor network has many sensing nodes which provide facility like, communication, computation. Wireless sensor network provide facility of finding shortest path. This paper provides the shortest path between source to destination for improving performance. In this, bellman ford distributed approach is used that is distance vector algorithm. It will help in finding shortest path with dynamically assigned nodes and it avoids malicious nodes and intrusion. We use dijkstra's algorithm to find malicious nodes and we generate random graphs and it will give calculate the distance between the nodes.

##### A. Introduction

Wireless sensor network is a collection of many ad-hoc networks which provide facility like communication, computation and sensing capability. Wireless sensor node built up from few to thousand of nodes where each node is connected to one or several sensors. Wireless sensor has to produced many tiny nodes and low cost. In this network cooperatively pass their data through the network to main location. Wireless sensor connect the physical world with digital world and provide understanding of our environment. Wireless sensor network must be reliable and scalable to support large applications. They must be accurate in providing required information while performing in network to avoid data load. Wireless sensor nodes provide many applications like. Military, health monitoring, entertainment, smart building, industrial, planning etc. In Wireless sensor network there are many number of small companies that effect the hardware and software design. There are many number of algorithms that find the shortest path between the nodes. Dijkstra's is an all pair shortest path algorithm that finds the shortest path between source to destination and every other node. It will find the route with positive edge weight only. It cannot find route with negative edge. To find route with negative edge weight bellman-ford algorithm is used. That find the route with negative edges. Bellman-ford has two distributed approaches that are: distance vector algorithm and link state algorithm. In distance vector algorithm weight are update periodically with the help of route metric and hop count. And a table is sent to all others neighbours . In link state algorithm only information passed is between node is connectivity related. There are many other algorithms that provides the shortest path for improving performance.

##### B. Problem description

In performance measurement of dijkstra's using wireless sensor network the problem formulation is based on the time complexity of the different algorithms that's are used in estimating the shortest path in various routing algorithm. The dijkstra's algorithm found shortest path with positive edge weight cycles. In bellman ford algorithm and there are negative edge weight cycles that are not able to give the least time complexity in wireless sensor network. Both algorithms have the worst time complexity in nature. And bellman-ford is slower than dijkstra's for some problem, but more versatile, as it is capable of handling graph in which some of the edge weight are negative number. In such a case, the bellman ford can detect negative cycle and report there existence. A distributed variant of bellman-ford is distance vector routing. Distance vector algorithm is iterative and asynchronous and each local iteration is caused by, local link cost change and distance vector update message from neighbour. A distance vector routing protocol require that a router inform its neighbour of topology changes periodically as compared to link state protocols, which require a router to inform all the nodes in a network of topology changes. The distance vector have less computational complexity. In link state routing this contrast with distance-vector routing protocol, which works by having each node share its routing table with its neighbours. In link state protocol the only information passed between node is connectivity require. The distance vector avoid loop formation but suffer from increased complexity. To avoid the increased complexity in distance vector, make distance vector algorithm dynamic. By making algorithm dynamic graphs are dynamically created and avoid loop formation, malicious nodes can not be occurred and there is no intrusion attack. That reduced the time complexity and improve the performance of the algorithm. This gives least time complexity to the wireless sensor network. With the help of dijkstra's it is easy to find the malicious nodes and choose the best

one. The random selection of nodes can take more time and increased the time complexity. The random selection of nodes are developed with dijkstra algorithm.

### C. Experimental Results

In this section, we conduct an experiment to compare the paths between the nodes. In that paths, the multi routing shortest path was done by using dijkstra's algorithm. Here the shortest path it means low cost was found by the shortest path algorithm. Here java code uses the applets so that applet viewer shows how the shortest path was done. With the use of java software the result was shown in the report. We have got successful result. The experiment was successfully complete. The result of this project finding the shortest path for a single source to to all pairs of vertices by using the Dijkstra's Algorithm. It gives the cheapest cost and its implementation is easy.

## DIJKSTRA ROUTING

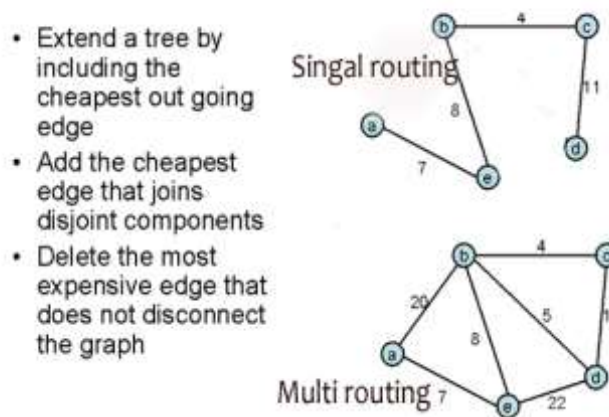


Fig 2: Dijkstra Routing

### D. Multicast

#### Algorithm used during Implementation

#### Dijkstra's and its Implementation

Dijkstra's algorithm is discovered by Edsger W.dijkstra in 1956 and published three year later. The algorithm solves the single shortest path problem on a weighted directed graph. It can find shortest path between many nodes and a single node. Dijkstra's algorithm is used in many fields like artificial intelligence and its variant is known as uniform cost search and its idea is breath first search. Dijkstra's help in finding the route in robot motion planning problem. Dijkstra's is also known as shortest path problem.

#### Steps for dijkstra's algorithm:

1. Initialize S with the start vertex, s, and V-S with the remaining vertices.
2. For all v in V-S
3. Set p[v] to s.
4. If there is an edge(s,v)
5. Set d[v] to w(s,v).

#### Else

6. Set d[v] to  $\infty$ .
7. While V-S is not empty
8. For all u in V-S, find the smallest d[u].
9. Remove u from V-S and add u to S.
10. For all v adjacent to u in V-S
11. If  $d[u]+w(u,v)$  is less than d[v].

12. Set  $d[v]$  to  $d[u]+w(u,v)$ .
13. Set  $p[v]$  to  $u$ .

In dijkstra algorithm assign to every node a tentative distance value: set initial node to zero and to infinity to all other nodes. Mark all node unvisited. Set initial node as current node. Create a set of the unvisited nodes called the unvisited set consisting of all the nodes except the initial node. For current node, consider all of its unvisited neighbours and calculate their tentative distance.

#### Efficiency

- Distance vector algorithm is faster then dijkstra's while performing in wireless network. And network traffic load is less for distance vector algorithm.
- Speed for dijkstra's is faster for same number of process run.
- Dijkstra's perform very faster in random selection of nodes.
- With increasing number of process dijkstra's algorithm eventually becomes faster because no communication occurs.
- Distance vector algorithm has better average delay in network and others.

#### Time Complexity:

- Time complexity of the shortest path algorithms is depend on the number of vertices, number of edges and edge length.
- It is observed that time complexity of the dijkstra's algorithm depends on the number of vertices and is inversely proportional to the number of vertices.
- Time complexity is least for distance vector algorithm but time complexity was higher for bellman-ford than dijkstra's.
- Time complexity in distance vector algorithm is depend on the route metric and hope count.

#### Performance and Efficiency:

- Finding the distance with lower number of nodes the bellman-ford is better and for higher number of nodes the dijkstra's is better and efficient.
- The distance vector is more efficient because it need some calculation for every node in the network and each node has local information and no chances of failure of nodes.
- Distance vector algorithm is better in average delay and traffic load.
- Distance vector has best convergence result

#### E. Multicast Routing In Sensor Networks

Multi-path routing, a routing technique that enables data transmission over multiple paths, is an effective strategy in achieving reliability in wireless sensor networks. However, multi-path routing does not guarantee deterministic transmission. This is because more than one path is available for transferring data from the source node to the destination node. A hybrid multi-path routing algorithm is proposed for industrial wireless mesh networks for improving reliability and determinacy of data transmission, as well as to effectively deal with link failures. The proposed algorithm adopts the enhanced Dijkstra's algorithm for searching the shortest route from the gateway to each end node for first route setup. A virtual pheromone distinct from the regular pheromone is introduced to realize pheromone diffusion and updating. In this way, multiple routes are searched based on the ant colony optimization algorithm. The routes used for data transmission are selected based on their regular pheromone values, facilitating the delivery of data through better routes. Link failures are then handled using route maintenance mechanism. Simulation results demonstrate that the proposed algorithm outperforms traditional algorithms in terms of average end-to-end delay, packet delivery ratio, and routing overhead; moreover, it has a strong capacity to cope with topological changes, thereby making it more suitable for industrial

DIJKSTRA'S ALGORITHM

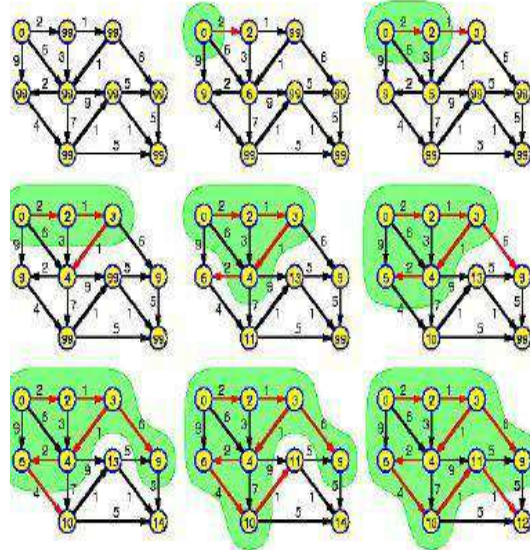


Fig 3 : Dijkstra Algorithm

V. SIMULATIONS RESULTS

In the MATLAB simulation, a random network of 50 nodes was created and Dijkstra's algorithm was used to find the routes between Source node and destination node. In Figure 4 the blue dots represent the true position of unknowns; anchors are indicted explicitly. A line between two nodes indicates a radio link. In the figure the Red dotted line represent the shortest path between the anchors that means from source node to destination node. In this figure the source node is 12 and destination node is 21. The shortest path between source node and destination node is: Path = 12 9 40 46 16 2 21; Total Cost is = 6. In this method 4% of ranging error occurs where as in the previous method [2], the ranging error is 5%.

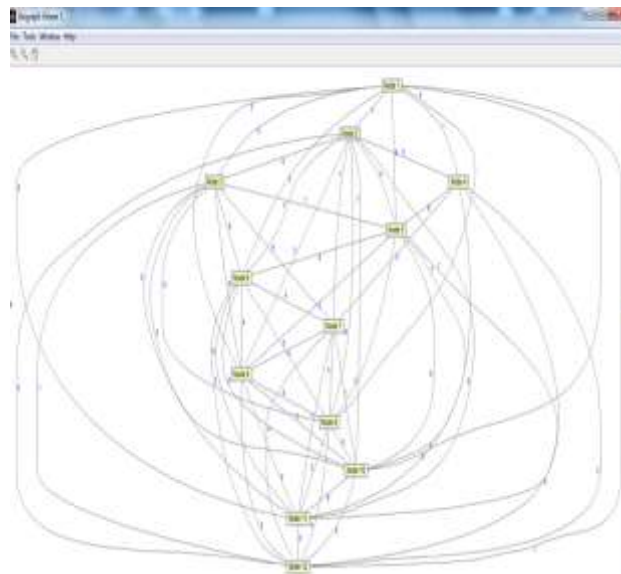


Fig.4. Shortest path between Source node and destination node





Fig 5: Home Page



Fig.6.Create random network

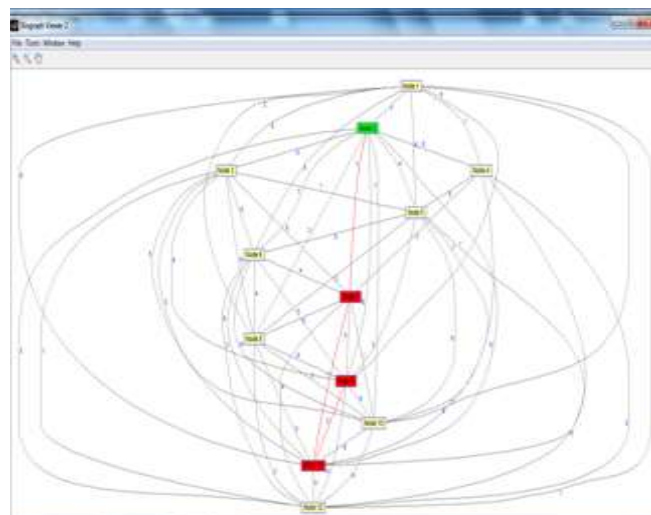


Fig 7. Shortest path between Source node and destination node

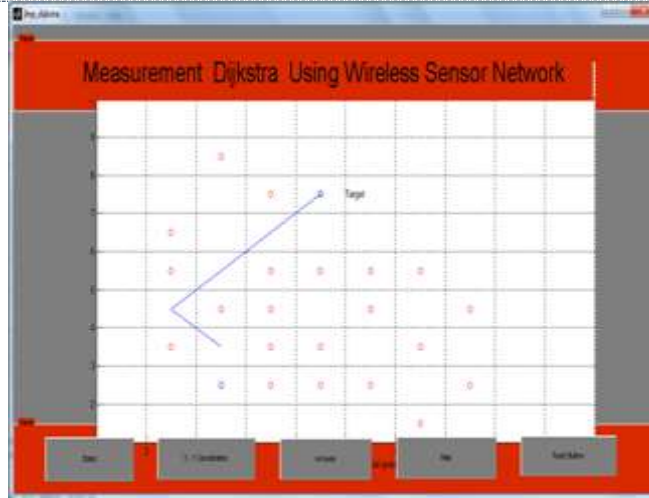


Fig 8. Shortest path x y co-ordinate Source node and destination node

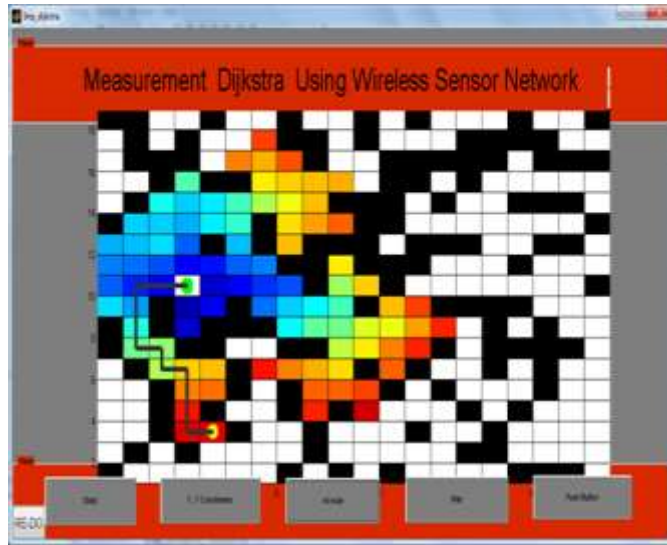


Fig 9. Malicious Node Detection

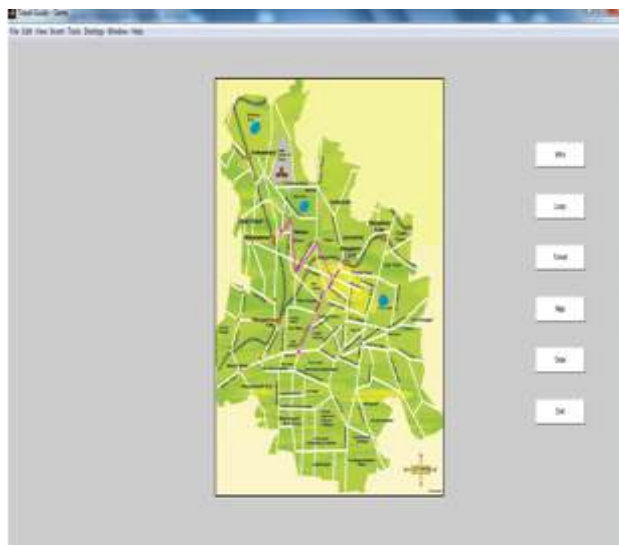
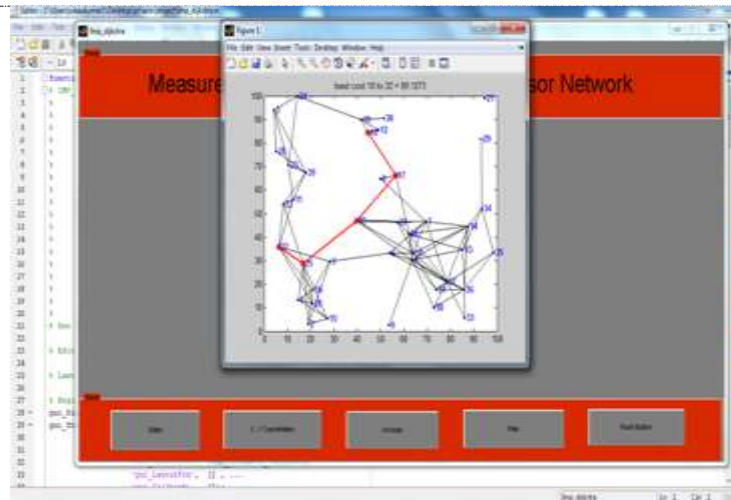


Fig 10. Real Time Shortest path between two cities



*Fig 11. Dynamic shortest path found with distance calculations*

## VI. CONCLUSIONSS ANDS FUTURES WORK

It is shown that the proposed algorithm works well for near uniform radio propagation. However, in the real world, radio propagation indoors and in cluttered circumstances is far from uniform. Local distance estimation may also be poor. Further simulations will be needed to determine reducing the range errors by using MDS algorithms can be to such errors. As Dijkstra's algorithm builds the local positions and routings of the estimated sensors for applications that require absolute coordinates of nodes, waiting until large number sensor nodes has formed before transforming to absolute coordinates may be a poor choice. Using the method described here, Position estimation using the shortest path method between source node and destination node with low cost in wireless sensor networks that compute absolute coordinates of individual nodes or sub networks independently can be developed.

Long distance shortest-path information is used only for rough layout decisions while two-hop information is used to determine precise node positions. It would be interesting to develop a framework that precisely characterizes the contribution of each datum to the position estimation. The main question is whether an approach based on unified statistical inference could be as efficient as the special purpose algorithms explored here. Shortest path method can be extended by applying more advanced MDS techniques. Instead of Dijkstra's algorithm, Interior routing algorithm, Exterior routing algorithm and Hierarchical routing algorithms can be applied. We have done some limited experiments with shortest path method using Dijkstra's algorithm. Our results show that Dijkstra's algorithm is better than the MDS-Square method using the connectivity level of the network is low, and is comparable with Dijkstra's algorithm when the connectivity level is high and ranging error is low as shown in Table.2.



Table 1

Comparative analysis of Shortest path method over Square method						
Author name	Description	No. of nodes	Cost	Radio Range	Accuracy	Range error
Previous Work[1]	A Square method is used to estimate all sensors' relative locations by applying MDS to compute the relative positions of sensors with high error-tolerance.	>50	Less	50	Low	5%
Present Work	Position estimation using the shortest path between the source node and destination node with low cost in wireless sensor networks	>50	Less	200	High	4%

## VII. REFERENCES

- [1] **N. Pushpalatha and Dr.B.Anuradha**, "Distribution of Nodes on Square Method for Wireless Sensor Networks", in International Journal of Computer Science and Telecommunications [Volume 3, Issue 1, January 2012].
- [2] **N.Pushpalatha and Dr.B.Anuradha**, " Study of Various Methods of Wireless Ad-hoc Sensor Networks using Multidimensional Scaling for Position Estimation" Global Journal Engineering and AppliedSciences-ISSN2249-2631(online): 2249-2623(Print) GJEAS Vol.1(3), 2013
- [3] **Rohit Kadam, Sijian Zhang, Qizhi wang, Weihua Sheng**, Itdimensional Scaling Based Location Calibration for Wireless Multimedia Sensor Networks" The 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems October 18-22, 2014, Taipei, Taiwan
- [4] **X. Nguyen, M.I. Jordan, and B. Sinopli**. "A kernel-based learning approach to ad- hoc Sensor network localization". ACM Transactions on Sensor Networks, 1(1):134–152, 2015.
- [5] **Xiangji, and HongyuanZha**, "Sensor positioning in wireless Ad-hoc sensor Networks using multidimensional Scaling", in 23rd annual joint conference of the IEEE computer and communication society, pp: 2652-2661, 2014
- [6] **F.Zhao and L.Guibas**. "Wireless Sensor Networks: An Information Processing Approach". Elsevier and Morgan Kaufmann Publishers, 2014.
- [7] **R. L. Moses, D. Krishnamurthy, and R. Patterson**. "A self-localization method for wireless sensor networks". Paper on Applied Signal Processing, 2003(4):148-158, March 2013.
- [8] **Shang,Y.; Ruml,W; Zhang,Y.; Fromherz, M.P.J.**"Localization from mere connectivity". Fourth International ACM Symposium on Mobile Ad Hoc Networking and Computing ; 2013 June 1-3; Annapolis; MD. NY: ACM; 2003; 201-212.
- [9] **K. Akkaya, M. Younis**, A survey on routing protocols for wireless sensor networks, Ad Hoc Networks, Vol. 3(3), 2015, 325-349
- [10] **F. Zabin, S. Misra, I. Woungang**, REEP: Data-centric, energy-efficient and reliable routing protocol for wireless sensor networks, Communications, IET, Vol. 2-8, 2008, 995-1008
- [11] **Y. Xu, J. Heidemann, D. Estrin**, Adaptive Energy-conserving Routing for Multihop AH hoc Network Research Report 527, USC/Information Sciences Institute, 2014
- [12] **B. Yahya, J. Ben-Othman**, An energy efficient and QoS aware multipath routing protocol for wireless sensor networks, IEEE 34th Conference on Local Computer Networks, 2009, 93-100
- [13] **T. H. Cormen, C. E. Leiserson, R. L. Rivest**, Introduction to Algorithms (Second Edition), MIT Press and McGraw-Hill, 2011, 595-601
- [14] **T. Van Dam, K. Langendoen**, An adaptive energy-efficient MAC protocol for wireless sensor networks, Proceedings of the 1st International Conference on Embedded Networked Sensor Systems, 2013, 171-180
- [15] **D. J. Nelson, K. Sayood, H. Chang**, An extended least-hop distributed routing algorithm, IEEE Transactions on Communications, Vol. 38-4, 1990, 520-528

- [17] **E. R. Sanchez, L. M. Murillo, B. Montrucchio**, An adaptive power-aware multi-hop routing algorithm for wireless sensor networks, 2011 8th International Conference on Information Technology, 2011, 112-116
- [18] **S. S. Chiang, C. H. Huang, K. C. Chang**, A minimum hop routing protocol for home security systems using wireless sensor networks, IEEE Transactions on Consumer Electronics, Vol. 53-40, 2012, 1483-1489
- [19] **E. W. Dijkstra**, A note on two problems in connexion with graphs, Numerische Mathematik, 1959, 269-271
- [20] **P. Murali, K. Rakesh, C. Hota, A. Yla-Jaaski**, Energy-aware routing in mobile ad-hoc networks, Wireless Days, 1st IFIP, 2014, 1-5
- [21] **D. Cheng, Y. Xun, T. Zhou, W. Li**, An energy aware ant colony algorithm for the routing of wireless sensor networks, Intelligent Computing and Information Science, 2011, 395-401
- [22] **Pankaj Verma, J.S Bhatia**, "Design And Development Of GPS-GSM Based Tracking System With Google Map Based Monitoring", International Journal of Computer Science, Engineering and Applications (IJCSA) Vol.3, No.3, June 2013.
- [23] **Vishal Bharte, Kaustubh Patil, Lalit Jadhav, Dhaval Joshi**, "Bus Monitoring System Using Polyline Algorithm", International Journal of Scientific and Research Publications, Volume 4, Issue 4, April 2014.
- [24] **Sacha Varone**, "On a many-to-one shortest paths for a taxi Service", Haute école de gestion de Genève CRAG - Centre de Recherche Appliquée en Gestion Cahier de Recherche
- [25] **Carlos Martín García and Gonzalo Martín Ortega**, "Route planning algorithms: Planific@ Project", International Journal of Artificial Intelligence and Interactive Multimedia, Vol. 1, No 2.
- [26] **Abboud, Marwan, LM Abou Jaoude, and Ziad Kerbage**. "Real Time GPS Navigation System." disponible sur <http://webfea-lb.fe.aub.edu.lb/proceedings/2014/SRCECE-27.pdf> (2014).
- [27] **A.Prakash, R.Manickavasagam**, "Elegant Way of Reaching Destination Using GPS and Drivers Ability", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 4, Issue 3, March 2014.
- [28] **Hu Jian-ming; Li Jie; Li Guang-Hui**, "Automobile Antitheft System Based on GSM and GPS Module," Intelligent Networks and Intelligent Systems (ICINIS), 2012 Fifth International Conference
- [29] **Boris V. Cherkassky**, Andrew V. Goldberg, and Tomasz Radzik. Shortest paths algorithms: Theory and experimental evaluation. Math. Program., 73:129–174, 1996.
- [30] **Harold N. Gabow**. Scaling algorithms for network problems. J. Comput. Syst. Sci., 31(2):148–168, 2014.

#### CITE AN ARTICLE

**N., & Khanchi, P. WIRELESS SENSORS NETWORKS OVERCOMING SHORTEST PATHS PROBLEMS USING DIJKSTRA'S ALGORITHM. INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY, 7(1), 97-109.**