

THESIS

VIRTUAL AND TOPOLOGICAL COORDINATE BASED ROUTING, MOBILITY
TRACKING AND PREDICTION IN 2D AND 3D WIRELESS SENSOR NETWORKS

Submitted by

Yi Jiang

Department of Electrical and Computer Engineering

In partial fulfillment of the requirements

For the Degree of Master of Science

Colorado State University

Fort Collins, Colorado

Fall 2013

Master's Committee:

Advisor: Anura P. Jayasumana

J. Rockey Luo
Indrakshi Ray

ABSTRACT

VIRTUAL AND TOPOLOGICAL COORDINATE BASED ROUTING, MOBILITY TRACKING AND PREDICTION IN 2D AND 3D WIRELESS SENSOR NETWORKS

A Virtual Coordinate System (VCS) for Wireless Sensor Networks (WSNs) characterizes each sensor node's location using the minimum number of hops to a specific set of sensor nodes called anchors. VCS does not require geographic localization hardware such as Global Positioning System (GPS), or localization algorithms based on Received Signal Strength Indication (RSSI) measurements. Topological Coordinates (TCs) are derived from Virtual Coordinates (VCs) of networks using Singular Value Decomposition (SVD). Topology Preserving Maps (TPMs) based on TCs contain 2D or 3D network topology and directional information that are lost in VCs. This thesis extends the scope of VC and TC based techniques to 3D sensor networks and networks with mobile nodes. Specifically, we apply existing Extreme Node Search (ENS) for anchor placement for 3D WSNs. 3D Geo-Logical Routing (3D-GLR), a routing algorithm for 3D sensor networks that alternates between VC and TC domains is evaluated. VC and TC based methods have hitherto been used only in static networks. We develop methods to use VCs in mobile networks, including the generation of coordinates, for mobile sensors without having to regenerate VCs every time the topology changes. 2D and 3D Topological Coordinate based Tracking and Prediction (2D-TCTP and 3D-TCTP) are novel algorithms developed for mobility tracking and prediction in sensor networks without the need of physical distance measurements.

Most existing 2D sensor networking algorithms fail or perform poorly in 3D networks. Developing VC and TC based algorithms for 3D sensor networks is crucial to benefit from the

scalability, adjustability and flexibility of VCs as well as to overcome the many disadvantages associated with geographic coordinate systems. Existing ENS algorithm for 2D sensor networks plays a key role in providing a good anchor placement and we continue to use ENS algorithm for anchor selection in 3D network. Additionally, we propose a comparison algorithm for ENS algorithm named Double-ENS algorithm which uses two independent pairs of initial anchors and thereby increases the coverage of ENS anchors in 3D networks, in order to further prove if anchor selection from original ENS algorithm is already optimal. Existing Geo-Logical Routing (GLR) algorithm demonstrates very good routing performance by switching between greedy forwarding in virtual and topological domains in 2D sensor networks. Proposed 3D-GLR extends the algorithm to 3D networks by replacing 2D TCs with 3D TCs in TC distance calculation. Simulation results show that the 3D-GLR algorithm with ENS anchor placement can significantly outperform current Geographic Coordinates (GCs) based 3D Greedy Distributed Spanning Tree Routing (3D-GDSTR) algorithm in various network environments. This demonstrates the effectiveness of ENS algorithm and 3D-GLR algorithm in 3D sensor networks.

Tracking and communicating with mobile sensors has so far required the use of localization or geographic information. This thesis presents a novel approach to achieve tracking and communication without geographic information, thus significantly reducing the hardware cost and energy consumption. Mobility of sensors in WSNs is considered under two scenarios: dynamic deployment and continuous movement. An efficient VC generation scheme, which uses the average of neighboring sensors' VCs, is proposed for newly deployed sensors to get coordinates without flooding based VC generation. For the second scenario, a prediction and tracking algorithm called 2D-TCTP for continuously moving sensors is developed for 2D sensor networks. Predicted location of a mobile sensor at a future time is calculated based on current

sampled velocity and direction in topological domain. The set of sensors inside an ellipse-shaped detection area around the predicted future location is alerted for the arrival of mobile sensor for communication or detection purposes. Using TPMs as a 2D guide map, tracking and prediction performances can be achieved similar to those based on GCs. A simple modification for TPMs generation is proposed, which considers radial information contained in the first principle component from SVD. This modification improves the compression or folding at the edges that has been observed in TPMs, and thus the accuracy of tracking. 3D-TCTP uses a detection area in the shape of a 3D sphere. 3D-TCTP simulation results are similar to 2D-TCTP and show competence comparable to the same algorithms based on GCs although without any 3D geographic information.

ACKNOWLEDGEMENTS

I am honored to accomplish my research work for Master of Science degree with the guidance of my advisor Dr. Anura Jayasumana, committee members Dr. Indrakshi Ray and Dr. Rocky Luo, help from friends, and support from my family.

I would like to express my deepest gratitude to my advisor, Dr. Anura Jayasumana, for his guidance, patience and providing me with an excellent atmosphere for doing research. I would like to thank Dr. Dulanjali Dhanapala and Vidarshana Bandara from my lab, for their kindness in broadening my academic view. I also would like to thank my friends Negar Mosharraf and Dr. Dilum Bandara who have accompanied me during my master study in my lab.

I would like to thank my parents and relatives. They were always supporting me and encouraging me with their best wishes. I would not have done my study and research in America without love and support from my family.

TABLE OF CONTENTS

Abstract.....	ii
Acknowledgements.....	v
Chapter 1 – Introduction	1
1.1 Motivation and problem statement	4
1.2 Contributions.....	7
1.3 Outline.....	10
Chapter 2 – Background and Related Work	11
2.1 Virtual Coordinate System (VCS) and related techniques	11
2.2 Routing algorithms in geographic and virtual domain	15
2.2.1 Geographic coordinate based routing algorithms for 2D and 3D WSNs.....	15
2.2.2 Virtual coordinate based routing algorithms for 2D WSNs.....	17
2.3 Mobility tracking and prediction algorithms for WSNs	20
Chapter 3 – Extreme Nodes Search and Geo-Logical Routing for 3D WSNs	23
3.1 Extreme Node Search (ENS) for 3D WSNs	23
3.2 Geo-Logical Routing (GLR) for 3D WSNs.....	26
3.3 Simulation and discussion.....	28
3.3.1 Test networks for simulation	28
3.3.2 Effects of anchor selection on performance of 3D-GLR	35
3.3.3 Routing performance comparison between 3D-GLR and 3D-GDSTR	38
3.3.3.1 Routing performance comparison in TN1, TN2 and TN3	38
3.3.3.2 Routing performance comparison in TN4 and TN5	39
3.3.3.3 Routing performance comparison in scaled TN4	40

3.4 Summary	41
Chapter 4 – Virtual Coordinate Generation Schemes for Mobile Sensor.....	42
4.1 Virtual coordinate generation schemes for mobile sensor.....	44
4.2 Simulation and discussion.....	45
4.2.1 Evaluation metrics	45
4.2.2 Simulation results for 2D networks	46
4.2.2.1 Mobile sensors receive packets from 2D networks	48
4.2.2.2 Mobile sensors pass packets in 2D networks.....	51
4.2.3 Simulation results for 3D networks	54
4.2.3.1 Mobile sensors receive packets from 3D networks	55
4.2.3.2 Mobile sensors pass packets in 3D networks.....	55
4.3 Summary	57
Chapter 5 – Topology Preserving Maps Modification Schemes	58
5.1 Topology preserving maps generation.....	58
5.2 Modification scheme1 and simulation results.....	60
5.3 Modification scheme2 and simulation results.....	66
5.4 Summary	69
Chapter 6 – Topological Coordinate based Mobility Tracking and Prediction	70
6.1 Topological coordinate generation scheme for mobile sensor	70
6.2 2D Topological coordinate based tracking and prediction algorithm.....	71
6.2.1 Simulation results.....	75
6.3 3D Topological coordinate based tracking and prediction algorithm.....	84
6.3.1 Simulation results.....	86

6.4 Summary	89
Chapter 7 – Summary and Future Work	91
7.1 Summary and conclusion.....	91
7.2 Future work.....	93
References.....	95
Appendix A – Simulator of TCTP Algorithm	100
A.1 Main function.....	100
List of Abbreviations	105

CHAPTER 1

INTRODUCTION

Wireless Sensor Networks (WSNs) consist of a large number of sensor nodes, deployed over an area to collaboratively monitor or sample data from physical or environmental phenomenon such as humidity in soil, temperature of water, density of chemicals in air, suspicious event or target in battlefield, etc. Recent developments in Microelectromechanical Systems (MEMS) manufacturing and on-chip technology have brought revolutionary changes to the design and performance of sensors. Sensor nodes have become tiny in size, affordable in cost, robust in performance, imbedded with powerful processor and have satisfactory computation capability and memory [3][49][61]. Large-scale WSNs are able to accomplish complicated sensing tasks in harsh and dangerous environments. Sensors periodically communicate with each other and report sampled data to a Base Station (BS) so that large area sensing process can be managed intelligently. WSNs have so far drawn much attention from researchers, and intensive research work has been done in 2D WSNs in areas such as localization, routing, and tracking [2][3][4][5][6][9][11][23][30][39]. However, many problems remain to be solved especially in 3-dimensional (3D) WSNs and Mobile Wireless Sensor Networks (MWSNs).

Compared with 2D WSNs, 3D WSNs can be placed in buildings [51][56], under-water monitoring [27][48], SkyMedia camera system [41] and even artificial eye vision system for human to restore vision [52]. Although a significant body of research exists on geographic information based 2D WSNs, many such schemes are not effective, do not scale well, or cannot be used with 3D WSNs. The difficulty of applying 2D to 3D stems from two main causes: a) the differences in geometric and inherent network properties of 2D and 3D networks, and b) the constraints related to radio or other signal communication among nodes in open 2D networks vs. in 3D deployments. First, consider the inherent geometric and network properties. The difficulties in extending 2D algorithm to 3D in sensor networking context is addressed in [44], with examples related to node deployment, coverage, and structural restrictions. While solutions to certain 2D problems are directly applicable to 3D, many require a

significant increase in computational complexity; there are also several other problems that do not generalize to 3D at all [44]. The second factor that makes it difficult or in some cases almost impossible to apply solutions designed for 2D networks to 3D is related to the restrictions from communication signals. 3D networks in many cases are deployed in harsh and irregular environments, which contain 3D obstacles, surfaces reflecting or absorbing radio signals. Some of these networks, e.g., those deployed on 3D surfaces or 3D volumes enclosing regions devoid of sensors, introduce complex geographical voids. Such environments render many technologies such as Global Positioning System (GPS) and distance measurements using Received Signal Strength Indication (RSSI), ineffective. Furthermore, in such environments, the communication topology can become significantly decoupled from the Geographic Coordinates (GCs) of the nodes. Two sensor nodes that are on different floors of a building may just be feet apart, but require complex communication paths. Subsurface contaminant tracking, where nodes are placed in wells, is another such example [31]. Thus, even if GCs are obtained by means other than GPS or RSSI, the algorithms based on GCs will perform significantly poorly in 3D networks. Several recent works has focused on GC based 3D WSN related to applications such as those for underwater sensing [48] or indoor building environments [51][56]. In 3D WSNs, sensor nodes need to be equipped with specific sensors to get altitude or depth information and GPS equipment to get horizontal plane position [56]. In 3D Underwater Sensor Network (USNs), normally a positioning algorithm is required for localization because GPS does not work well underwater [27][48]. Basically, 3D geographic position of beacon sensor nodes near water surface will be projected to other sensor nodes to get comparative location for these sensor nodes [48]. GC based 3D WSNs have to afford extra money, energy and computation cost not only from localization equipment to get 3D geographic information but also from redesigning related algorithms to adjust to 3D networks, thus still facing face challenges in areas such as localization, routing, etc. [27][42][48].

An example of 3D USNs can be seen in Figure1.1, which is from Georgia Institute of Technology's Underwater Acoustic Sensor Networks (UW-ASNs) project [59] .

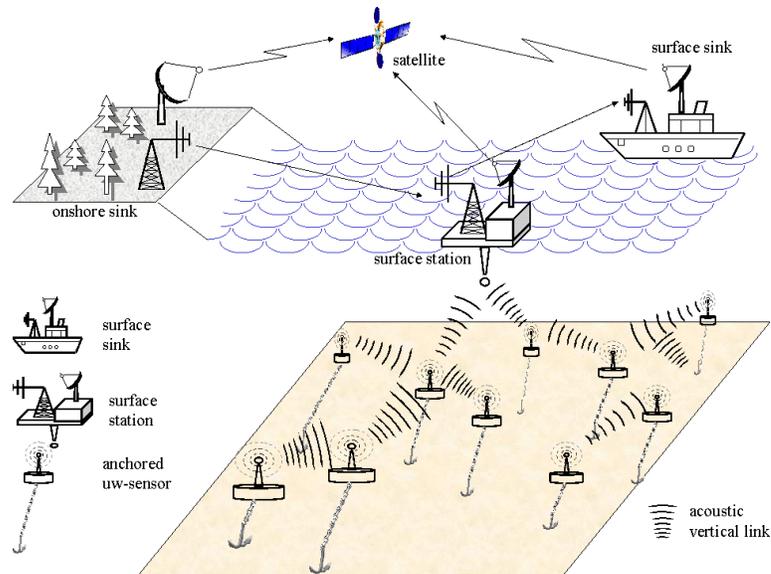


Figure 1.1: Example of 3D USNs [59]

As a class of WSNs, MWSNs have advantages of more efficient energy usage, more channel capacity, better targeting and data fidelity over the traditional static WSNs [2][38][55]. Mobility applications include mobile object tracking in military reconnaissance, animal and habitat protection and traffic control in city area. Figure 1.2 gives an example of MWSNs from Ohio State University's project Wireless Multimedia Sensor Networks [60] in which sensors are carried by mobile objects such as human, vehicle and aircraft. However, challenges brought about by mobility also exist such as mobile sensor's localization and tracking [1][2][30][39]. MWSNs may consume higher power energy and have more dynamic topology [2]. Mobility increases the difficulty in getting sensor nodes accurately localized and localization issue remains a hot topic among researchers. Traditional localization technologies including GPS, RSSI, Time of Arrival (TOA), Direction of Arrival (DOA), Angle of Arrival (AOA), etc., are used to obtain geographic position information for sensors in sensor networks. These techniques however have limitations related to energy consumption, measuring accuracy and applicable environments [2][43][55], which become the main obstacles in developing large-scale MWSNs.

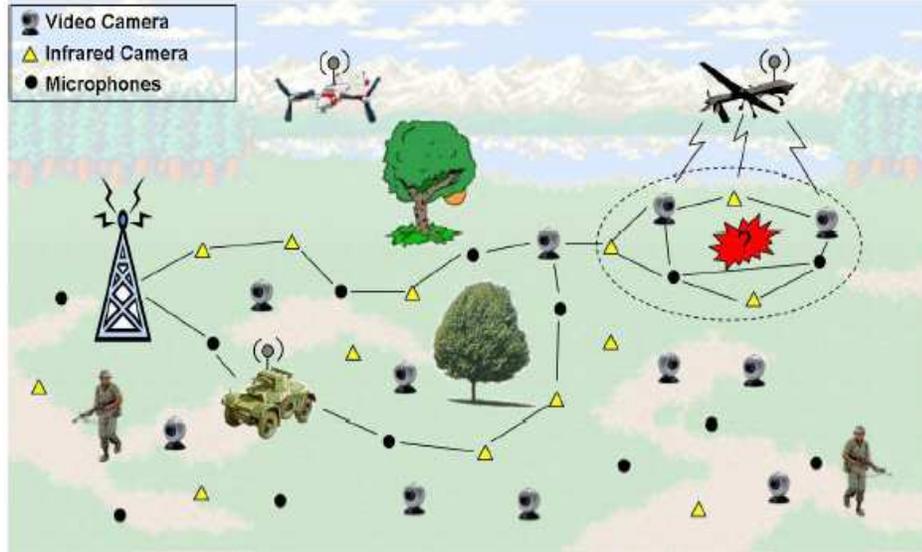


Figure 1.2: Example of MWSNs [60]

1.1 Motivation and problem statement

Further development of both 3D WSNs and MWSNs will suffer from hardware cost, power consumption from geographic localization equipment and usage limitations from physical environments, especially when the scale of WSNs becomes larger and larger. Virtual Coordinate System (VCS) based WSNs are free of geographic information and physical distance measurement. Different from traditional localization system based on GCs, VCS is established based on hop distances from each sensor node to a specific set of sensor nodes called anchors or landmarks [13][17][19][37]. Suppose there are M anchors in network, each node in VCS is characterized by a coordinate vector of size M containing the shortest hop distance to each of a set of M anchors [13][17][19][37]. Virtual Coordinates (VCs) replace GCs for sensors to get location in multi-dimensional virtual domain instead of 2D or 3D geographic domain. Thus, VCS is free of geographic localization equipment such as GPS and localization algorithm such as RSSI, etc. Additionally, VCS is insensitive to physical voids and the VC generation based on hop counts makes it easy to extend from 2D to 3D sensor network. Therefore, VCS can adapt to various environments and conditions in 2D or 3D indoor, outdoor or underwater area sensor networks. Due to the advantages stated above, VCS based sensor networks have received board attention from researchers

[10][13][14][16][17][18][19][20][21][37][45]. However, the main drawback of VCS is that 2D or 3D directional information is lost because VCs propagate radially. To overcome this drawback, Topology Preserving Maps (TPMs) [17][18] is a novel technique which can generate topology maps of 2D and 3D networks and obtain lost direction information from VCS. Singular Value Decomposition (SVD) based dimensionality reduction scheme is used to derive Topological Coordinates (TCs) of sensor nodes from VCs. TPMs are able to preserve the internal and external boundaries and basic shape of both 2D and 3D network, validated to provide a good alternative of 2D physical maps for applications in mapping [17], routing [20] and boundary detection [18]. Extreme Node Search (ENS) algorithm [19] is another key technique for VCS based 2D WSNs. ENS algorithm is aimed at providing a good set of anchors, which is necessary basis for TPMs technique. In VCS based 2D WSNs, Geo-Logical Routing (GLR) algorithm is a state-of-art routing technique [20]. Since GLR algorithm proposed in [20] is used only in 2D WSNs, we call it 2D-GLR in the rest of this thesis. 2D-GLR switches between virtual domain and topological domain to reach the best routing performance. Using greedy forwarding, the sensor node compares its and the neighbors' norms L^2 of VC/TC distance to destination and forward the packet to the neighbor who has the minimum distance [20]. If the packet is stuck in local minima in both VC and TC routing mode, the packet will be routed to the nearest anchor to destination and then routing is switched back to the VC routing mode again. 2D-GLR algorithm shows desirable routing performance for 2D networks compared with Greedy Perimeter Stateless Routing (GPSR) [34] algorithm and some other VC based routing algorithms such as Logical Coordinate Routing (LCR) [13] and Convex Subspace Routing (CSR) [16]. Since VC generation and TC derivation is based on hop count and network connectivity, which is free of geographic dimensional information. 2D-GLR routing algorithm is expected to be extended to 3D sensor networks without complicated modification.

Although VC generation doesn't depend on dimension of the network and TPMs technique apply in both 2D and 3D sensor networks, VCS based 3D WSNs still face challenge from proper anchor selection. Anchor placement plays key role in of VCS based techniques such as TPMs and 2D-GLR and poor anchor selection will result in a large amount of identical coordinates and thus huge Topology

Preserving Error (E_{TP}) [17][19], which will severely decrease the accuracy of TCs and performance of routing algorithm. In this thesis, we are motivated to use existing ENS algorithm in order to provide effective anchor selection for VCS based 3D WSNs. Also in this thesis, routing performance and effectiveness of 3D-GLR with ENS anchor placement in 3D WSNs are explored for the first time.

In MWSNs, mobility of sensors makes it possible to sense and monitor dynamic events in environment or collecting data in a more flexible way. In general, mobility can be understood in the following two aspects: objects of mobility and patterns of mobility. Mobile objects in MWSNs include mobile BS, mobile sensor nodes, mobile relay nodes and mobile cluster heads [55]. For patterns of mobility, basically there are two types of scenarios [2]. The first scenario is that the mobile sensors are deployed dynamically. They may be dropped from aircraft and vehicle or placed as additional new sensors manually in the original network environment. In this scenario, change in network connectivity is due to the dynamic deployment of sensors, and these newly introduced mobile sensors normally stay in stationary status or change location very slowly. Such new sensors are deployed in sensing area when there is a need of increasing sensing granularity or sensing tasks. In the second scenario, mobile sensors such as the mobile sensors equipped in robots in battle field or attached to animals in habitat protection projects move within a WSNs [2]. In this thesis, we mainly focus on communication of mobile sensor nodes in both scenarios of dynamic deployment and continuous movement, including dynamic deployed mobile sensor's routing performance and continuous moving mobile sensors' tracking and detection performance.

In VCS based MWSNs, an important question is how to communicate with the mobile sensors, e.g., how to route packets to the mobile sensors. Mobile sensors are newly inserted into the original static sensor network. These nodes do not have VCs or other location information in virtual domain which is necessary for address based routing algorithms [13][16][20][37]. Obtaining VCs for mobile sensors in virtual domain is the initial, but also an important phase. Re-flooding VC generation messages containing hop distances information from anchor set to entire network is both time and energy consuming. What's worse, when mobile sensors keep changing positions over time, the VCs obtained from flooding are

always out of date and thus of little use due to the dynamic topology change in network. As a result, taking concern of time, energy, computation of generating VCs, a simple VC generation scheme without VC re-flooding from anchors should be applied for mobile sensors. Mobile sensors are equipped with radios to be able to communicate with neighbor sensors within a certain range. In this thesis, we solve VC generation problem for mobile sensors by making use of their neighboring sensors' VCs, which we name as Average Scheme, Mixture Scheme and Minimum Scheme. Thus, mobile sensors' generated VCs come from average value, minimum value from neighboring sensors' VCs. When evaluating the effectiveness of VC generation schemes, two main aspects of routing performance of mobile sensors are considered: mobile sensors receive packets from original static network, and mobile sensors pass packets over the original static network. Ideally, mobile sensors' routing performance under such cases should be at least as good as the routing performance of the original static sensor network. In addition, the insertion of new mobile sensors should not affect the routing performance of the original network. For mobile sensors in continuous movement, prediction based tracking and detection strategy is widely used in GCs based MWSNs. However, this strategy has not yet been applied in topological domain where exact geographic position and physical distance don't exist. In this thesis, we borrow the idea of prediction based tracking and detection from geographic domain and apply this idea in topological domain which is derived from VC set of the network. We propose an algorithm for mobile sensor detection, prediction and tracking algorithm, named as Topological Coordinate based Tracking and Prediction (TCTP) algorithm. The 2D and 3D versions of this algorithm are called 2D-TCTP [32] and 3D-TCTP respectively. TCTP algorithm detects mobile sensor's future location based on its current velocity and direction, estimated in the topological domain. 2D-TCTP and 3D-TCTP are able to accomplish mobile sensor tracking, and thus communication without any geographic information and physical distance measurements.

1.2 Contributions

Since anchor selection plays a crucial role in performance of VCS based algorithm and techniques such as mapping [17] and routing [20], improper anchor placement can greatly degrades

mapping accuracy and routing performance in VCS based WSNs. Before any anchor selection scheme becomes available, anchors are selected either randomly or manually [13][37] for VCS based algorithms. ENS algorithm [19] provides a good set of anchors to improve performances in mapping and routing algorithms [19]. The possibility and performance of extending ENS algorithm in 3D WSNs need to be explored in order to provide necessary basis for VCS based techniques in 3D networks. Among VCS based routing algorithms [10][13][16][20][21][37], 2D-GLR algorithm is a novel routing technique [7]. 2D-GLR switches greedy forwarding routing in three modes: VC mode, TC mode and anchor mode to reach the best routing performance. In VCS whose direction information is lost, TCs are derived from VCs of network using SVD from TPMs technique [20] and TCs act as 2D/3D Cartesian coordinates well to preserve the internal and external boundaries and basic shape of both 2D and 3D networks [17][18]. In greedy forwarding of 2D-GLR, the sensor node compares neighboring nodes' norms L^2 of VC/TC distances to destination with the distance from itself, then, forward the packet to the neighbor who has the minimum distance [20]. If the packet is stuck in local minima in both VC and TC routing mode, the packet will be routed to the nearest anchor to destination and then routing is switched back to the VC mode again [20]. 2D-GLR algorithm shows satisfactory routing performance for 2D networks compared with GC based GPSR algorithm and some other VC based routing algorithms such as LCR [13] and CSR [16]. However, the routing performance of GLR algorithm has not yet been studied in 3D WSNs. In 3D WSNs, the comparison of routing performance of GC based and VC/TC based routing algorithm still remain unknown. Although VCS base WSNs have advantages of saving hardware and energy cost from GPS localization equipment, its effectiveness in routing and communication performance need to be further validated for 3D WSNs.

Contributions of this thesis include the following:

Firstly, the extension of ENS algorithm in 3D WSNs is presented. To prove the effectiveness of ENS anchor selection in 3D networks, another anchor selection algorithm named Double-ENS algorithm based on ENS algorithm is proposed as comparison algorithm. In Double-ENS algorithm, two independent ENS anchor set will be generated instead of one set so that the amount and coverage of ENS

anchors in 3D sensor networks can be increased. The purpose of proposal of Double-ENS algorithm is to figure out if increased amount of ENS anchor will necessarily increase the routing performance for 3D networks, compared with original ENS anchors. Then, 2D-GLR algorithm is extended to 3D-GLR algorithm by simply replacing 2D TCs with 3D TCs in TC distance calculation in TC based greedy forwarding mode. Performance of 3D-GLR algorithm is evaluated via simulation of five 3D network examples. 3D-GLR's performance is compared with GC based 3D Greedy Distributed Spanning Tree Routing (3D-GPSTR), in order to further prove the routing effectiveness and efficiency of ENS and 3D-GLR algorithm in 3D WSNs.

The second main contribution addresses the VC generation when new sensor nodes are introduced or mobile sensors move around in a sensor network. Three VC generation schemes for mobile sensors are designed that require only simple computations and no additional floodings from anchor nodes, namely Average Scheme, Mixture Scheme and Minimum Scheme. The routing performances of mobile sensors' receiving packets and passing packets in the network are evaluated by simulation in both 2D and 3D networks under these three VC generation schemes. The routing performances are compared with that of the original static network without the mobile sensors. Thirdly, considering the mobile sensors are in continuous moving status, a prediction based tracking and detecting algorithms called 2D-TCTP for VCS based 2D WSNs is proposed. The detecting performance of 2D-TCTP algorithm in mobile sensor tracking and perdition is compared with the same approach based on GC information. Detection failure rate will be used as the main evaluation matric. Also, a simple modification of TPMs which considers radial information present in the first principle component from SVD is also developed to improve the folding problem at the edges of TPMs so that the accuracy of TPMs can be increased. TPMs with more accurate topology are used to provide a more helpful guide map for 2D-TCTP algorithm.

The fourth contribution in this thesis is to extend and modify 2D-TCTP algorithm for 3D WSNs. Performance of 3D-TCTP algorithm is compared with same tracking and prediction algorithm based on GCs.

1.3 Outline

The rest of thesis is organized as followed. Background and related work is reviewed in Chapter 2. Chapter 3 presents the extension of the ENS algorithm to 3D networks. Also, simulation results and discussion for 3D-GLR algorithm with ENS anchor placement are presented. In Chapter 4, VC generation schemes for mobile sensor nodes and related routing simulation results are given. TPMs modification schemes and related simulation results are shown in Chapter 5. Following Chapter 5, 2D-TCTP algorithm is developed and tested in Chapter 6. Also in Chapter 6, extension of 2D-TCTP to 3D-TCTP is accomplished. Summary and future work is given in Chapter 7.

CHAPTER 2

BACKGROUND AND RELATED WORK

The reviews of background and related work are organized as followed. VCS based techniques such as TPMs, ENS and boundary detection are presented in Section 2.1. Section 2.2 mainly focuses on existing GC and VC based routing algorithms in 2D and 3D WSNs. In Section 2.3, recent tracking and prediction techniques in MWSNs are reviewed.

2.1 Virtual Coordinate System (VCS) and related techniques

A VCS is based on a set of anchors, which correspond to a subset of nodes, selected randomly or by an anchor selection strategy [17][18]. Each node in the network, including anchors, is characterized by a VC vector, consisting of shortest hop distances to each of the anchors [13][17][19][37]. Figure 2.1 below shows an example of VCS and corresponding VCs to each of the nodes. In general, for sensor network of N sensor nodes, if number of M anchors is selected, there will be a $N \times M$ matrix for VC of the network. Figure 2.2 shows VC matrix for a network consisting of 100 nodes and 10 anchors. Compared with traditional WSN in which sensors obtain geographic location from GPS, sensor network based on VCS does not have either directional information or topology map of the network.

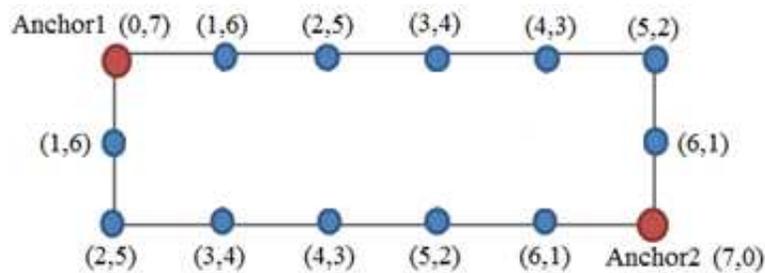


Figure 2.1: Example of VCS based sensor network with two anchors

	1	2	3	4	5	6	7	8	9	10
1	10	41	39	23	27	6	28	37	29	40
2	9	40	40	24	28	7	27	36	30	41
3	8	39	41	25	29	8	26	35	31	42
4	7	38	40	26	30	9	25	34	32	41
5	6	37	39	27	31	10	24	33	33	40
...
100	3	32	34	24	26	7	21	28	30	35

Figure 2.2: VC matrix for network of 100 nodes and 10 anchors

Without any geographic information, one important feature of VCS is that physical void becomes invisible in virtual domain [16]. VCs based on hop distance are able to provide topological connectivity information of the network better than GCs, which helps in VC based routing algorithms. In GC based routing, distance between two sensor nodes is calculated as the exact straight-line distance, which does not contain any route information. With this distance, there may be no actual path between these two nodes at all. However, in VC based routing, VCs are generated from hop distances and are related to actual available paths [16]. Thus, VC based routing algorithms have demonstrated satisfactory routing performance [10][13][16][20][21][37]. However, the information of minimum number of hops to a certain set of anchors is far from enough to obtain any 2D and 3D topology information of the network. Since geographic topology is the key and indispensable information for many important applications like boundary detection and tracking, a topology extracting method using SVD is presented in [17] named as TPMs. VCS based WSNs only have higher dimensional information of the network based on connectivity. For example, the network of M anchors has M dimensional information. SVD is applied to extract 2D or 3D topology information from this higher dimensional information and therefore retain the lost directional information.

Consider a network with M anchors and N sensor nodes (Normally $M \ll N$). Thus, each node is characterized by a VC vector of length M , the i^{th} element of which corresponds to the number of hops from the node to the i^{th} anchor. Let P be the $N \times M$ matrix containing VCs of all sensor nodes in the network. The method to generate the TPM of network using SVD method as follows [17]:

$$P = U.S.V^T \quad (2.1)$$

$$P_{SVD} = P \times V \quad (2.2)$$

$$[X_T, Y_T] = [P_{SVD}^{(2)}, P_{SVD}^{(3)}] \quad (2.3)$$

P_{SVD} is a $N \times M$ matrix containing sensor nodes' principal components and can be seen as the projection of network's VCs set on unitary matrix V . The first column $P_{SVD}^{(1)}$ is the most significant component but contains 1D radial information which is not sufficient for 2D or 3D TPMs [17][18]. Meanwhile second column $P_{SVD}^{(2)}$ and third column $P_{SVD}^{(3)}$ contain the topological information that can be translated to angular information and these two columns can be seen as 2D Cartesian coordinate set for sensor nodes in topological domain [17]. X_T and Y_T , are each $N \times 1$ column vectors, and $[X_T, Y_T]$ in eqn. (2.3) is the TC set for the 2D network. i.e., its i^{th} row corresponds to the TCs of the i^{th} node. Then the generated maps containing lost topology information are called TPMs in [17]. The direction information and network topology are well obtained without any help from GPS localization equipment. In [17], other methods to generate TPMs of VCS based WSN and the complexities of each method are also described and analyzed. The key idea is to find which set of nodes to do SVD with lowest computation complexity and then generate its unitary matrix V . Besides using VC set of entire nodes in network, only partial nodes in network such as anchor node set or randomly selected node set can also be used to generate unitary matrix V . Consider A is the matrix containing VCs of anchor nodes. A is a $M \times M$ matrix. SVD components of P are obtained by replacing V with V_A from matrix A 's SVD in eqn. (2.4) and (2.5). 2D TCs are still extracted from second and third column of P_{SVD} in eqn. (2.6). When random nodes are selected as node set for SVD, consider the number of random nodes is r . R is $r \times r$ matrix containing VCs of selected random nodes. Similarly, SVD components of P are obtained by replacing V with V_R in eqn. (2.7) and (2.8). 2D TCs are extracted in the same way. When P is used for SVD calculation, the complexity is approximately $(4N^2M + 8NM^2 + 9M^3)$ operations [17]. When only subset of nodes is used for SVD calculation, the complexity is less than $(4M^2N + 8M^3)$ [17]. Moreover, TPMs extraction from

SVD not only works for 2D network, but also is proved to be an effective method for 3D sensor network topology extraction [18].

$$A = U_A \cdot S_A \cdot V_A^T \quad (2.4)$$

$$P_{SVD} = P \times V_A \quad (2.5)$$

$$[X_T, Y_T] = [P_{SVD}^{(2)}, P_{SVD}^{(3)}] \quad (2.6)$$

$$R = U_R \cdot S_R \cdot V_R^T \quad (2.7)$$

$$P_{SVD} = P \times V_R \quad (2.8)$$

Generating TPMs using SVD for 3D networks is similar to 2D networks. SVD components can be generated based on entire node set in network or partial node set [18]. Differently, the fourth column $P_{SVD}^{(4)}$ of generated P_{SVD} is seen as the coordinates in Z axis, which can be seen from eqn. (2.9). Thus, X_T , Y_T and Z_T are each $N \times 1$ column vectors and $[X_T, Y_T, Z_T]$ in eqn. (2.9) [18] is the TC set for the 3D network. i.e., its i^{th} row corresponds to the TCs of the i^{th} node. Although the topology information of network can be retained with help of SVD, TPM is just approximate map and distortion of map, flipping order of nodes, folding effects and etc. do exist [17][18]. Modifying TPMs for more accurate direction information plays crucial role if TPMs act as guidance map replacing geographic map in practical applications.

$$[X_T, Y_T, Z_T] = [P_{SVD}^{(2)}, P_{SVD}^{(3)}, P_{SVD}^{(4)}] \quad (2.9)$$

Anchor placement has significant effects on performance of VCS based WSNs. Poor anchor placement will result in huge identical VC problem and E_{TP} which means a pair of sensor nodes is out of order compared to original geographic topology [17][19]. TPMs and VCS related techniques will suffer from poor anchor placement. ENS algorithm [19] is a state-of-art technique for VCS based 2D sensor networks. ENS algorithm is aimed at providing a good set of anchors, which is the important basis for TPMs technique. Basically, one pair of random anchors is selected to initiate the VC generation messages flooding [19]. Then Directional Virtual Coordinate (DVC) is generated from a pair of VCs [19]. Consider sensor node n_i 's VCs from anchor A_j and A_k are $[h_{ij}, h_{ik}]$. The DVC of n_i is given in eqn. (2.10) [19].

$h_{A_j A_k}$ is the shortest hop count from anchor A_j to A_k . After DVC calculation is finished, local sensor nodes having with the minimum value or the maximum value of DVC among its neighboring nodes will be selected as anchors. Simulation results show that VCS based sensor networks with ENS anchor placement have less topology error (E_{TP}) and better routing performances [19] compared with random anchor placement.

$$f(h_{ij}, h_{ik}) = (h_{ij}^2 - h_{ik}^2) / 2 h_{A_j A_k} \quad (2.10)$$

Boundary detection for 2D and 3D VCS based WSNs is another novel technique based on TPMs technique [18]. By replacing GCs with TCs in Heron's formula [18], boundary nodes will be detected in topological domain instead of geographic domain. Boundary detection based on TPMs works in both 2D networks and 3D surface networks [18].

In all, TPMs are able to preserve the internal and external boundaries and basic shape of both 2D and 3D network, validated to provide a good substitute of 2D physical maps for applications in mapping [17], routing [20] and boundary detection [18] in WSNs.

2.2 Routing algorithms in geographic and virtual domain

2.2.1 Geographic coordinate based routing algorithms for 2D and 3D WSNs

In large-scale WSNs, a large number of sensor nodes are densely scattered in large area. The communication is done either among each sensor node and its neighbor nodes or between sensor node and BS. Routing performance is a crucial criterion when evaluating WSN's communication system because routing can affect many areas such as power management, data dissemination, etc. [4]. Capability of routing packets among sensor nodes effectively is required in today's WSNs development. When a new model of network is designed, routing performance is the first thing to be evaluated. The big difference between traditional wired networks routing and WSNs routing is that IP address based routing is not feasible in WSNs. Since a large amount of sensors are deployed in unplanned physical environment, it is not realistic to set up certain number of sensors as "routers". Additionally, sensors' computation

capability and memory are limited to maintain such “routing table”. Therefore, a global addressing scheme is not suitable in WSNs [4]. WSNs have its own unique features and these features need to be considered before any routing algorithm is designed. In many situations and cases, the purposes of communication in WSNs are reporting sensing data back to BS or BS sends control or request commands to sensors in a particular sensing area. In WSNs, each sensor can play the role of data packet sender, receiver and passer. Besides, unlike the wired internet, the topology of WSNs is hard to stay the same over long time. The changes due to the unpredictable physical environment and sensing task can lead to changes in the networks topology even though the mobility of certain sensor nodes can be ignored. Therefore more adaptive routing algorithms are required in WSNs. GPSR [34] is commonly used responsive routing algorithm designed for fast changing mobile and wireless networks. In GPSR algorithm, packets and data are routed based on the connectivity information and physical position. A sensor node evaluates its neighbors’ norm distances in geographic domain to the destination and forwards the packet to the neighbor sensor which has the minimum distance to the destination. GPSR is based on greedy forwarding and requires the global geographic location information of the network. However, in today’s WSNs, sensing tasks in the unpredictable and even dangerous environment such as volcano sensing, under water sensing or indoor sensing won’t allow the use of GPS.

Additionally, GPSR can hardly be practical for 3D WSNs because of local minima problem. Several works have been published to find solutions to GC based routing algorithm for 3D sensor networks. Greedy Random Greedy routing (GRG) is randomized algorithm and tries to route packets based on random walk only for network with Unit Ball Graph (UBG) topology [22]. Greedy Hull Greedy (GHG) routing algorithm uses planarization to construct network hulls [36]. GRG algorithm suffers from limitation of sensor networks types and GHG algorithm suffers from complexity from planarization computation. J. Zhou et al. proposed 3D Greedy Distributed Spanning Tree Routing (3D-GDSTR) [56] algorithm which is extended from B. Leong’s 2D-GDSTR algorithm [35]. 3D-GDSTR is free of planarization algorithm and tested on both real sensor networks test-bed and TOSSIM simulation. 3D-GDSTR uses GC based greedy forwarding to find closer neighbor to destination in 1-hop communication

range. When packet is stuck in local minima, 3D-GDSTR increases communication range to 2-hop neighborhood to find closer neighbor. If local minima still exists in 2-hop range, routing will switch to hull trees (which are generated from minimal spanning tree) routing mode to find sensor node who has destination as its child node. 3D-GDSTR is stated practical and to be able to achieve near 100% delivery rate in designed network with satisfactory node degree [56]. However, the cost from localization equipment and hull tree generation are the main drawbacks of 3D-GDSTR algorithm.

Due to the disadvantages of GC based routing algorithms stated above, routing algorithms with limited use of GPS or free of GPS are highly recommended nowadays to adapt to the varied sensing tasks and application. Techniques like RSSI, TOA, DOA, AOA, etc. are also used by today's WSN [8] so that only a small number of sensors are equipped with GPS to get geographic location and the other sensors obtain their own locations by estimating the signal strength with neighboring nodes. However, these techniques are error-prone [43] thus the accuracy of locations is far from enough for application like tracking. Solutions to find appropriate technique replacing GPS or other localization algorithm are being researched these decades and one of the most popular solutions is VC based routing algorithm.

2.2.2 Virtual coordinate based routing algorithms for 2D WSNs

Since 2000, researchers have started to find alternative solutions to replace GPS based localization technique in WSNs and some results and paper related to routing performances from GPS-free routing algorithms are published. Different from geographic domain, the domain without GPS based techniques is called virtual domain. Like GC based routing algorithms, greedy forwarding is commonly used in VC based routing algorithms [10][13][16][20][21][37]. Packet is routed to the neighbor who has the shortest distance to destination node in virtual domain. The distance between two nodes in virtual domain is the L^2 norm. Consider a network with M anchors and thus node X is identified by VC vector of size M : (x_1, x_2, \dots, x_m) . Similarly, node Y is identified by a VC vector of (y_1, y_2, \dots, y_m) . The distance between node X and Y node in virtual domain is defined in eqn. (2.11) [13][16].

$$D_{XY} = \sqrt{\sum_{i=1}^M (x_i - y_i)^2} \quad (2.11)$$

Local minima caused by inappropriate anchor selection or identical VCs are the main problem deteriorating routing performance of greedy forwarding in virtual domain. The following will give a brief review of VC based routing algorithms in recent 10 years.

LCR algorithm [13] was developed by Q. Cao et al. in 2004. In LCR algorithm, packet is forwarded to the nearest node to destination and backtracking function is called when it comes to the local minima. In 2005, A. Caruso et al. established a scalable coordinate-based routing algorithm for routing in GPS-free domain [10]. Greedy forwarding is used for routing data [10]. When the data is stuck in local minima, a ring search is conducted until a closer node is found. Thus routing algorithm is improved by delivering data to a zone of nodes without identical coordinates. The improved method is able to avoid identical coordinate problem although only a small amount of anchors are selected as anchors. In 2006, K. Liu et al. analyzed local minima in VC based routing as quantization noise for the first time. A new algorithm called Aligned Virtual Coordinates for Greedy Routing [37] in WSNs is proposed to remove the quantization noise by taking average VCs of n-hop neighboring sensor nodes as the node's VCs. The larger the range of neighborhood range is, the more accuracy will be obtained for sensor node's virtual location thus packet delivery rate performance can be improved. However, enlarging communication range will also increase communicate cost and power consumption. In 2007, Axis-Based Virtual Coordinate Assignment Protocol (ABVCap) [50] is proposed by M. J. Tsai et al. In this protocol, VCs of nodes in network are generated in such way that each node is assigned a five tuple VC corresponding to longitude, latitude, ripple, up and down. In 2011, D. C. Dhanapala et al. developed Convex Subspace Routing (CSR) [16] algorithm for VCS based WSNs. In this algorithm, not all of the VCs corresponding to each anchor are used for distance calculation in routing. Only a triplet of anchors which can compose a convex sub space for the node is selected as anchors for routing. When the packet is stuck in local minima, another triplet of anchors will be selected to ensure packet is routed in convex space to let packet escape from local minima. CSR is effective to avoid using anchors which may cause local minima issue. Simulations show that CSR outperforms GPSR and LCR in routing delivery rate [16]. In the same year, 2D-GLR algorithm was designed by the same authors [20]. All the routing algorithms which have been

mentioned so far take place only in virtual domain. Thus all related distance calculation and comparison happen only in virtual domain. 2D-GLR algorithm for the first time combines greedy forwarding in virtual and topological domain together [20]. The TPMs generation technique provides TC set for the network. TC set is very similar to GC set but in a somehow distorted topology domain. There are three modes in 2D-GLR algorithm [20]. The first one is greedy forwarding in virtual domain using VCs. The second one is routing in topological domain using generated TCs from SVD. The last one is anchor mode, which means the packet is routed to the nearest anchor to the destination and then routing is switched back to greedy forwarding in virtual domain. The routing is accomplished in the switching among these three modes. The packet is routing in virtual domain at the beginning. If it gets stuck in local minima, the mode is switched to topological domain. If the packet is stuck in local minima in both virtual and topological domain, the packet will be routed to nearest anchor. In 2D-GLR algorithm, high packet delivery rate can be guaranteed because algorithm takes advantage of three domains. From the simulation results, 2D-GLR algorithm can achieve the best packet routing performance so far compared with LCR and CSR algorithms in virtual domain and GPSR in geographic domain. Another routing algorithm called Direction Virtual Coordinate Routing (DVCR) [21] algorithm is using generated coordinates named DVCs for greedy forwarding and good routing performance is also reported [21].

The routing algorithms mentioned above are designed in VCS based 2D WSNs. Routing performance plays the most crucial role in sensor network's communication system. High packet delivery rate and low hop distance of routing packets are highly desired in today's large-scale WSN. The assumptions of all above algorithms are: 1. The network type is 2D WSNs. 2. The sensor nodes are all static thus the sensor network is static without dynamic sensor node deployment. However, performance of the VC based routing algorithms remains still unknown in 3D WSNs. Also, when tacking mobility into consideration, the routing algorithm for mobile sensors needs to be proposed to avoid regeneration of VCs and guarantee acceptable routing performance.

2.3 Mobility tracking and prediction algorithms for WSNs

WSNs are no longer static as the sensing demands grow. MWSNs make it possible to implement sensing in more dynamic environment. Robots technology can execute sensing task in environments which are dangerous to human being such as battle field or forest fire. In habitat protection and monitoring, equipping animals with sensors helps researcher better track, monitor and study their living habits and thus better protect them and the nature. No matter in which application, the collaboration among mobile sensors, static sensors and BS is done by communication between mobile sensors and static sensors. So far, many tracking and monitoring mobile object algorithm are proposed. In early times, this used to be called Object Tracking Sensor Network (OTSN) [39]. In OTSN, the placement of mobile sensors can be pre-defined or random. The number of mobile sensor can be single or multiple. The mobile sensor(s) can enter the WSN at random time or a specific time. In all, the activities of mobile sensor(s) can be both in a flexible or a fixed way depending on the specific requirements of applications.

Any tracking or monitoring algorithm should achieve this general goal: no matter in which way the mobile sensor moves or is operated, there should be some methods helping BS or human being to track, detect and then communicate with the mobile object. The exact or comparative location of mobile object should be known. The detection success rate should be high as much as possible. Also, communication between mobile sensor and BS should be effective and at the same time energy efficient. Many aspects need to be reconsidered or redesigned like routing protocols, data aggregation strategies, mobility of the target, localization algorithms and scheduling algorithms.

Generally, tracking algorithms can also be classified in the following three aspects: network structure, number of mobile objects being tracked and the type of mobile object. The common structures for sensor networks are: leader based structure, tree based structure and cluster based structure. Based on the number of mobile objects, tracking algorithms can be designed for single mobile object or multiple mobile objects. For the type of objects, algorithms can be classified into continuous object tracking and discrete object tracking. Consider the following two scenarios. First is a mobile node that moves in a sensor field, to be detected or tracked by the sensors. It could be a friendly node that cooperates with

sensors or a passive target. Either way its geographic position is found using the GCs of the sensors. This lets the node navigate further (using GCs) or allows sensors to track its position using GCs. Secondly, as a mobile node moves in a sensor field, it may be necessary to predict its position at some time in future. The GCs are used to estimate the velocity and direction, and then using some mobility model, to predict the target's position at a future time. Such predictions may be used to alert nodes downstream, for example. Another situation occurs when the mobile node cooperates and communicates with other sensor nodes. Suppose there is a BS that wishes to send a message or program updates to rendezvous with the mobile node at a future time. Rather than routing the message along the path that the mobile node followed, a prediction of its future position will allow the BS to send the messages directly to the rendezvous point.

So far, prediction scheme is widely used in tracking and detection algorithm in GCs based WSNs and can be applied in different structures of networks such as tree based structure and cluster based structure. The following of this section will give a brief review of existing prediction based tracking and detection algorithms in GC based WSNs. When mobile sensor keeps changing position at comparatively high velocity during sensing operation time, localization method cannot provide sufficient location information if BS wants to route to or reach the mobile sensors. In this case, prediction based tracking algorithms are designed to reach the mobile object at the future time by calculating future position linearly from sampling current velocity and direction. In [54], Distributive Predictive Tracking (DPT) is proposed for cluster-based WSNs. Cluster heads predict the mobile target's future position and wake up a triplet of sensors to wait for mobile target's arrival. Prediction-based Optimistic Object Tracking (POOT) scheme in [28] combines collecting and maintaining tracking information to minimize routing distance for predictive tracking. In [53], dual predictions take place at both BS and sensor nodes to improve the detection accuracy. Such predictions may, for example, be used to alert nodes downstream. In these prediction based tracking algorithms, it's assumed that mobile sensors get localized by either GPS or Radio Frequency (RF) techniques such as RSSI, etc. Considering the large cost and error from GPS and

RF based techniques, track algorithms without geographic information are of great interest and promising potential.

However, the protocols or algorithms cannot be applied in VCS based MWSNs directly and need to be changed or modified due to the difference between virtual domain and geographic domain. Considering applications related to VCS based MWSNs. We may be interested in the following scenarios:

1. Mobile sensor contacts other sensors or BS. Consider the mobile object is a friendly sensor. It moves in a certain area to collect data from environment such as humidity and temperature. Or animals are equipped with sensor so human being can monitor this animal. Sensors in this kind of objects need to send data to other sensors or BS in for the purpose of reporting.
2. Mobile objects are reached by other sensors or BS. In this case, BS may want to send commands to mobile objects for specific tasks.

For the first scenario, since mobile sensor act as data sender, sending data to somewhere in the network so it is not hard to find a neighbor who can pass the packet to the destination in VCS based MWSNs. However in the second scenario, traditional prediction based tracking algorithms in geographic main for tracking and detecting mobile objects cannot be applied in VCS based MWSNs directly since geographic information is not available. Tracking and prediction algorithms are needed to be reconsidered in topological domain instead of virtual domain, in order to adjust to VCS based WSNs.

CHAPTER 3

EXTREME NODE SEARCH AND GEO-LOGICAL ROUTING FOR 3D WSNs

The performance of mapping [17] and routing [16][20] greatly rely on anchor placement in VCS based WSNs. Identical coordinates problem resulted from improper anchor selection can lead to identical TC problem because SVD transition from VCs to TCs is a linear operation. Thus, nodes which are located in different geographic position will have the same position in generated topological domain, which greatly decreases the accuracy of generated TPMs. What's worse, identical coordinates problem in both virtual and topological domain can result in severe local minima problem and poor performance of greedy forwarding based routing would be expected. Before any anchor selection algorithm becomes available, anchors are selected either randomly or manually [13][37] for VCS based algorithms. However, performance of VCS based techniques cannot be guaranteed because the effectiveness of random anchor selection is unknown.

In this chapter, we firstly introduce existing ENS algorithm and related evaluation in Section 3.1. In Section 3.2, 3D-GLR algorithm is proposed. To prove the effectiveness of 3D-GLR with ENS anchor placement, routing performance simulations of 3D-GLR are given in Section 3.3, compared with GC based routing algorithm 3D-GDSTR. Section 3.4 gives a simple summary.

3.1 Extreme Node Search (ENS) for 3D WSNs

Selecting the number of anchors and the nodes to become anchors is critical for many VCS based algorithms. Due to the difficulty of solving these two problems together, random anchors are often used. With random anchor selection, a relatively large fraction of nodes have to serve as anchors to achieve good performance, but the number of anchors directly contributes to overhead and complexity. The ENS algorithm in [19] identifies a small set of extreme nodes as anchors. Results in [19] show that it can achieve better topology maps and very good routing performance in 2D networks. In ENS algorithm, one

DVC is generated using a pair of random anchors to impose a directional relationship among the nodes, and to identify extreme nodes based on this directional coordinate value [19][21].

In this thesis, we continue to use ENS algorithm for anchor selection for 3D WSNs. In order to prove the effectiveness of ENS algorithm in 3D networks, ENS algorithm is compared with random anchor selection in routing performance of 3D-GLR algorithm. Additionally, another anchor selection algorithm called Double-ENS algorithm for 3D networks is also designed as comparison algorithm for ENS. Basically in Double-ENS algorithm, two separate sets of ENS anchors are generated instead of one set of ENS anchors. The purpose is to figure out if the increased amount and coverage of extreme node anchors in network will also increase the routing performance, thus to prove if the original ENS algorithm already provides an optimal selection of anchors with smaller amount for 3D networks. The detailed description of Double-ENS algorithm is given below.

In original ENS algorithm, DVC for each node in network is generated using one pair of random anchors in VCS and lost directional information is restored in DVC [19][21]. Suppose sensor node n_i 's VC vector from two random anchors A_1 and A_2 are $[h_{iA_1}, h_{iA_2}]$, containing minimum hop distances to two anchors. Generation of n_i 's DVC is given in eqn. (3.1) and $h_{A_1A_2}$ is the shortest hop distance from anchor A_1 to A_2 . Then each node in the network checks whether its DVC value is the local minimum/maximum in h-hop neighborhood. If so, node will be selected as ENS anchor.

$$f_{DVC}(h_{iA_1}, h_{iA_2}) = (h_{iA_1}^2 - h_{iA_2}^2) / 2 h_{A_1A_2} \quad (3.1)$$

In Double-ENS algorithm, the anchor selection algorithm starts by selecting four random nodes A_1, A_2, A_3 and A_4 as initial anchors. After VC generation, each node now is in the possession of its initial VC vector containing minimum hop distances to the four anchors. This initial coordinates at node i is denoted by $[h_{iA_1}, h_{iA_2}, h_{iA_3}, h_{iA_4}]$, where h_{iA_j} correspond to the shortest hop distance between A_j and node i . Each node now calculates a pair of DVC [19][21]. The first, $f_{DVC}(h_{iA_1}, h_{iA_2})$ in eqn. (3.1), is based on A_1 and A_2 , and the second $f_{DVC}(h_{iA_3}, h_{iA_4})$ is based on A_3 and A_4 , which can be obtained from eqn. (3.2).

$$f_{DVC}(h_{iA_3}, h_{iA_4}) = (h_{iA_3}^2 - h_{iA_4}^2) / 2 h_{A_3A_4} \quad (3.2)$$

In a VCS, each anchor establishes a radially propagating distance profile from itself. For example, all the nodes that have a value $h_{iA_1} = 30$ are at a hop distance 30 from A_1 . This is similar to r coordinate in a (r, θ) coordinate system. As described in [19][21], a DVC combines two such VCs, to impose a directional relationship, similar to the x coordinate in a directional coordinate system such as (x, y) Cartesian system. Thus each node now is in possession of its DVC pair $[f_{DVC}(h_{iA_1}, h_{iA_2}), f_{DVC}(h_{iA_3}, h_{iA_4})]$ that imposes a two-dimensional coordinate system, albeit not orthogonal to each other.

Each node now checks whether it is a local minima or a maxima in its h -hop neighborhood with respect to $f_{DVC}(h_{iA_1}, h_{iA_2})$ or $f_{DVC}(h_{iA_3}, h_{iA_4})$ or both. If it is, the node selects itself as an anchor for the final VC generation. We call such anchors ENS anchors to avoid confusion with the initial anchors. To avoid too many anchors close to each other, that may happen because the initial four anchors cannot discriminate among those adequately, a localized pruning process is carried out. Each of these selected anchors checks whether any of its direct neighbors, i.e., those within 1-hop communication range, are also ENS anchors. If so, only one of those anchors, selected randomly, remain an ENS anchor, while others cease being ENS anchors. The purpose of this step is to remove multiple anchors that do not provide additional connectivity information, and to keep number to be the minimal to reduce unnecessary cost for VC generation. The Double-ENS anchor selection procedure is shown in Figure 3.1. $K_h(n_i)$ is the set of nodes in node n_i 's h -hop neighborhood in Figure 3.1.

```

1.  {A1, A2, A3, A4} = pick four random nodes
2.  Initiate floodings on {A1, A2, A3, A4} and generate a VCS
3.  For each node
4.      locally generates two DVCs
5.  End
6.  For each node
7.      //checks whether current node is a local minimum/ maximum h-hop neighborhood
8.  If  $f_{DVC}(h_{iA_1}, h_{iA_2}) < f_{DVC}(h_{jA_1}, h_{jA_2}); \forall n_j \in K_h(n_i)$ 
9.      Or  $f_{DVC}(h_{iA_3}, h_{iA_4}) < f_{DVC}(h_{jA_3}, h_{jA_4}); \forall n_j \in K_h(n_i)$ 
10.          $n_i$  is an anchor
11.     End
12.  If  $f_{DVC}(h_{iA_1}, h_{iA_2}) > f_{DVC}(h_{jA_1}, h_{jA_2}); \forall n_j \in K_h(n_i)$ 
13.     Or  $f_{DVC}(h_{iA_3}, h_{iA_4}) > f_{DVC}(h_{jA_3}, h_{jA_4}); \forall n_j \in K_h(n_i)$ 
14.          $n_i$  is an anchor
15.     End
16.  End
17.  For each node in anchor set
18.     For each neighbor in 1-hop neighborhood of current anchor node
19.         If neighbor is in anchor set
20.             record current anchor node once and neighbor node in redundant node set
21.         End
22.     End
23.     If redundant node set is not empty
24.         Pick and mark one node randomly in redundant node set
25.         Remove other nodes except for the marked node in redundant node set from the anchor set
26.         Update anchor set
27.         Clear redundant node set
28.     End
29.  End
30.  Selected anchor nodes generate the VCS

```

Figure 3.1: Double-ENS anchor selection algorithm

3.2 Geo-Logical Routing (GLR) for 3D WSNs

As outlined in Chapter 2, the TPMs of a network based on the VCs of nodes preserves are a distorted version of the physical map of a network. It has been shown in [20] that the VCs are in fact better than GCs for identifying the next node for forwarding the packet in 2D networks due to the fact the distortion accounts for the connectivity.

GLR algorithm uses greedy forwarding to route packet to neighbor node which has the minimum distance to the destination and switches greedy forwarding between virtual and topological domain. The distance is calculated as norm 2 distance L^2 in both virtual and topological domain. Suppose there are M anchors in VCS, two nodes N_a and N_b are each characterized by a VC vector, (a_1, a_2, \dots, a_M) and (b_1, b_2, \dots, b_M) correspondingly. The norm 2 distance L^2 between node N_a and N_b in virtual domain is given in eqn. (3.3) [20].

$$D_{AB, virtual} = \sqrt{\sum_{i=1}^M (a_i - b_i)^2} \quad (3.3)$$

Suppose generated TCs using TPMs technique of node N_a and N_b are $[X_{TC,a}, Y_{TC,a}]$ and $[X_{TC,b}, Y_{TC,b}]$ correspondingly. The norm 2 distance L^2 between node N_a and N_b in 2D topological domain is given in eqn. (3.4) [20].

$$D_{AB, topological} = \sqrt{(X_{TC,i} - X_{TC,d})^2 + (Y_{TC,i} - Y_{TC,d})^2} \quad (3.4)$$

Local minima problem is the main obstacle deteriorating the performance of greedy forwarding, in which the node sending the packet has the minima distance to the destination and cannot find a closer neighbor node to route packet to. GLR algorithm uses combination of three routing mode (VC mode, TC mode and anchor mode) to let packet escape from local minima and thus achieve better routing performance [20]. In GLR algorithm, the packet is routed using greedy forwarding in virtual domain (VC routing mode) at first place. If the packet is stuck in virtual domain, the routing is switched to greedy forwarding in topological domain (TC routing mode). If the packet is stuck in local minima in both virtual and topological domain, the packet is routed to the nearest anchor node to the destination and then routing mode is switched back to VC mode (anchor mode). GLR algorithm is easy to extend to 3D sensor network since generation of VC and TC only rely on network connectivity and SVD computation, which is independent from geographic dimensional information. In this thesis, we keep using the original algorithm proposed in [20] as 3D-GLR. We only replace 2D TCs with 3D TCs when calculating norm distance D_{AB} in topological domain between two nodes N_a (whose TCs are $[X_{TC,a}, Y_{TC,a}, Z_{TC,a}]$) and N_b (whose TCs are $[X_{TC,b}, Y_{TC,b}, Z_{TC,b}]$) in TC routing mode, which is given in eqn. (3.5).

$$D_{AB, topological} = \sqrt{(X_{TC,a} - X_{TC,b})^2 + (Y_{TC,a} - Y_{TC,b})^2 + (Z_{TC,a} - Z_{TC,b})^2} \quad (3.5)$$

The matric for evaluating 3D-GLR algorithm in this thesis is average routability, which is defined below:

$$\text{Average Routability [16][20][21]} = \frac{\text{Total number of packet reached destination}}{\text{Total number of packet generated}} \times 100\% \quad (3.6)$$

3.3 Simulation and discussion

In this section, 3D-GLR algorithm is simulated in five test networks of different shapes, connectivity information and scales, which are introduced in Section 3.3.1. To further prove the anchor selection effectiveness of original ENS algorithm for 3D networks, routing performances of 3D-GLR algorithm using anchor selection from original ENS, Double-ENS and random anchor placement are compared in Section 3.3.2. 3D-GLR algorithm is compared with GCs based 3D-GDSTR algorithm, in order to validate the routing performance of 3D-GLR algorithm, which is given in detail in Section 3.3.3. The performance of routing algorithms is evaluated by metric average routability. A simulator was developed using MATLAB® 2013a.

3.3.1 Test networks for simulation

In this section we introduce five types of 3D networks designed for simulation. These five 3D test networks include networks with a fixed or random topology, in concave or non-concave shape, with full network connectivity or low connectivity and of different scales. ENS algorithm is used to select anchor node for each test network.

Figure 3.2(a) and Figure 3.3(a) show the geographic maps of Test Network1 (TN1) and Test Network2 (TN2). TN1 and TN2 are both networks with a fixed topology in 1000unit×1000unit×1000unit cubic area and have physical sphere voids inside the networks. To help in viewing the big sphere void in the center of TN1, void is drawn as grey sphere and there is no sensor placement inside sphere void in Figure 3.2(a). Similarly, three small sphere voids of TN2 are drawn in grey color in Figure 3.3(a). TN1 consists of 792 sensor nodes and 6 of them are selected as anchors using ENS algorithm. TN3 consists of 791 sensor nodes and 9 nodes are selected as ENS anchors. ENS anchors are marked as black triangles in both Figure 3.2 and Figure 3.3. The communication range of each sensor is 100unit and the average node degree is 5 for TN1 and TN2. Figure 3.2(b) and Figure 3.3(b) show TN2 and TN3's generated TPMs with ENS anchor placement.

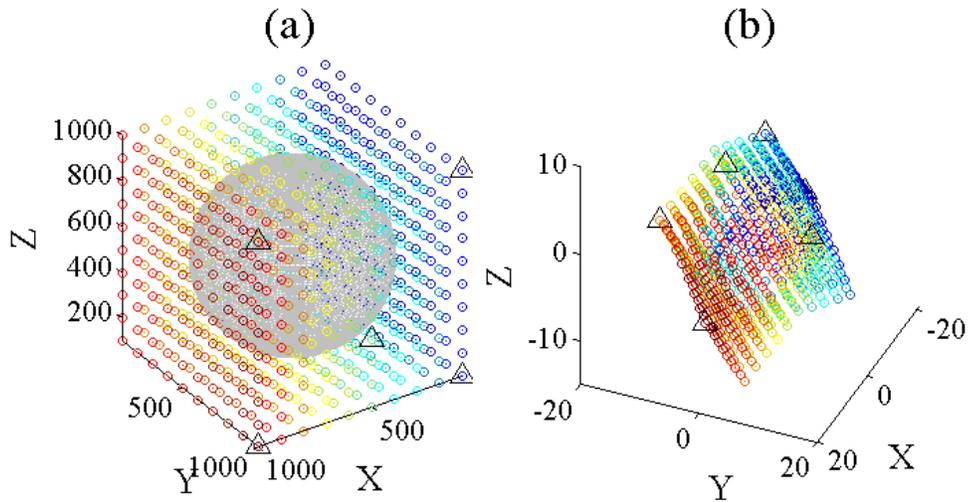


Figure 3.2: a) Geographic map of TN1 with one big visible sphere void inside and b) generated TPM of TN1 with 6 ENS anchors

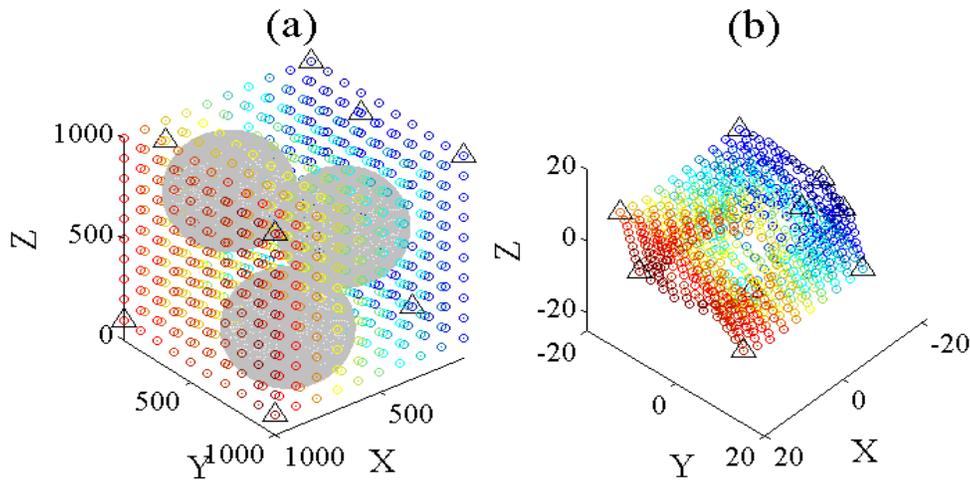


Figure 3.3: a) Geographic map of TN2 with three small visible sphere voids inside and b) generated TPM of TN2 with 9 ENS anchors

Test Network3 (TN3) used for simulation is shown in Figure 3.4(a). TN4 has a concave shape in $1000\text{unit} \times 1000\text{unit} \times 1000\text{unit}$ cubic area and 760 sensor nodes are placed inside the concave shape. Figure 3.5 shows the top view of TN3. Each node's communication range is 100 units and average node degree of TN3 is also 5. 6 ENS anchors are selected and generated TPM of TN3 is shown in Figure 3.4(b). ENS anchors are shown as black triangles in Figure 3.4.

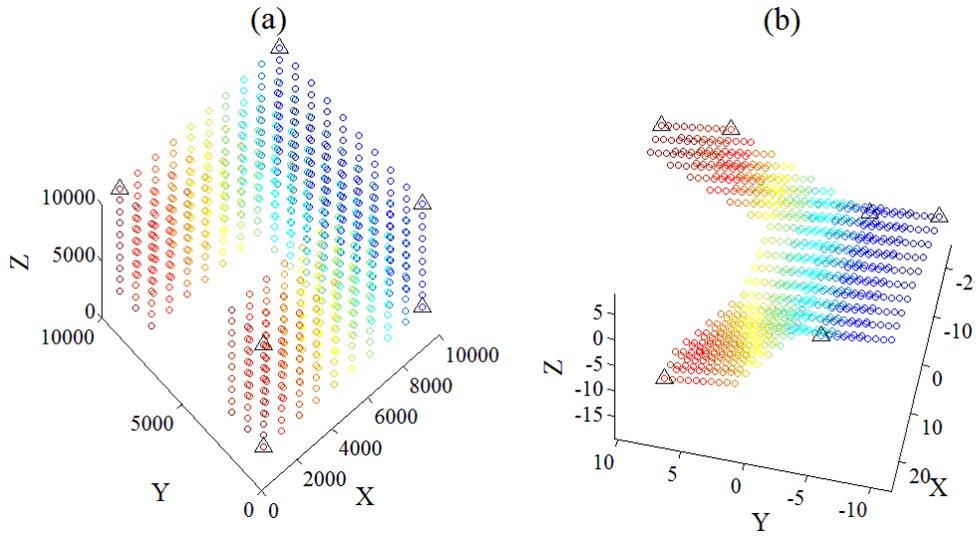


Figure 3.4: a) Geographic map of TN3 and b) generated TPM of TN3 with 6 ENS anchors

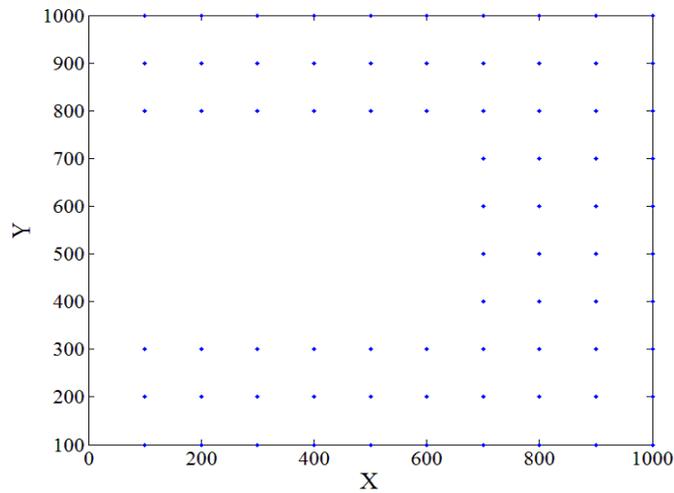


Figure 3.5: Top view of TN3

TN1, TN2 and TN3 are designed to test 3D-GLR algorithm in different network shape with imperfectness inside network. The following to be introduced Test Network4 (TN4) and Test Network5 (TN5) are designed with different network connectivity information.

TN4 are a group of networks based on a full grid network in 1000unit×1000unit×1000unit cubic area which consists of 1000 nodes. Each node has 100unit as communication range. A certain number of random chosen nodes are removed from the full grid network so that the average node degree will

decrease. For TN4, number of 0, 100, 200, 300, 400 of random nodes are removed from full grid network and they are named as TN4(a), TN4(b), TN4(c), TN4(d) and TN4(e) respectively. For each of TN4(b), TN4(c), TN4(d) and TN4(e), 5 random topologies are generated for simulation. The number of nodes, average number of ENS anchors and average node degree of five TN4 networks are given in Table 3.1. The average node degree gradually decreases as the number of removal nodes grows. 3D geographic maps and generated TPMs of TN4(a), TN4(b), TN4(c), TN4(d) and TN4(e) (each is selected from five random topologies) are shown in Figure 3.6 to Figure 3.10 respectively, with ENS anchors marked as black triangles.

Table 3.1: Number of nodes, average number of ENS anchors and average node degree of TN4(a), TN4(b), TN4(c), TN4(d) and TN4(e)

	Network Topology				
	<i>TN4(a)</i>	<i>TN4(b)</i>	<i>TN4(c)</i>	<i>TN4(d)</i>	<i>TN4(e)</i>
Number of nodes	1000	900	800	700	600
Average number of ENS anchors	5	6	7	7	6
Average node degree	5.4	4.9	4.3	3.8	3.4

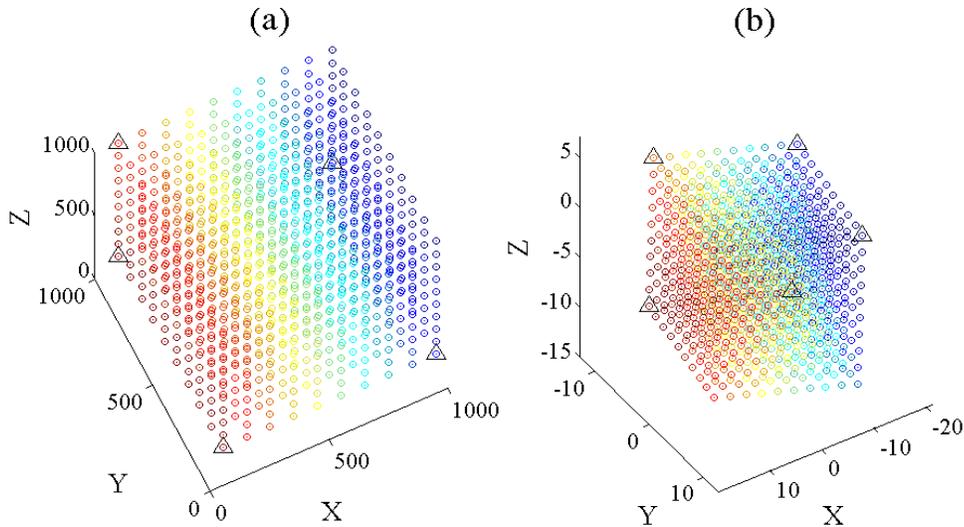


Figure 3.6: a) Geographic map of TN4(a) and b) generated TPM of TN4(a) with 5 ENS anchors

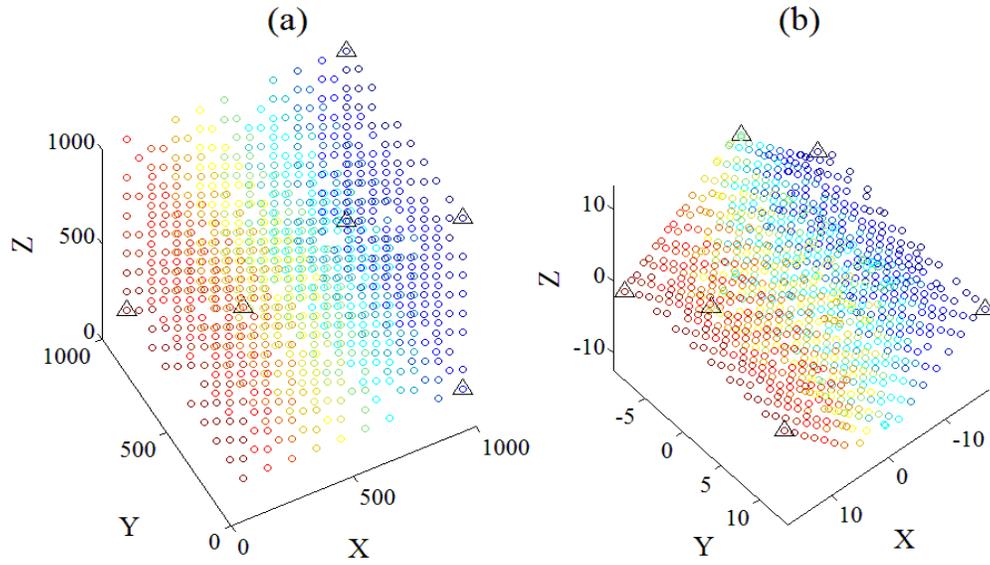


Figure 3.7: a) Geographic map of TN4(b) and b) generated TPM of TN4(b) with 6 ENS anchors

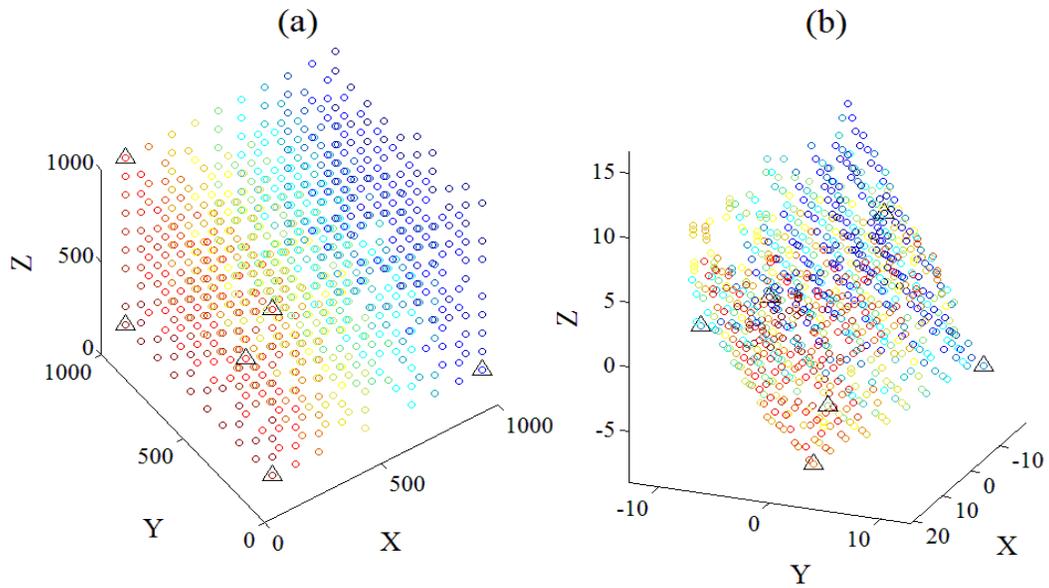


Figure 3.8: a) Geographic map of TN4(c) and b) generated TPM of TN4(c) with 6 ENS anchors

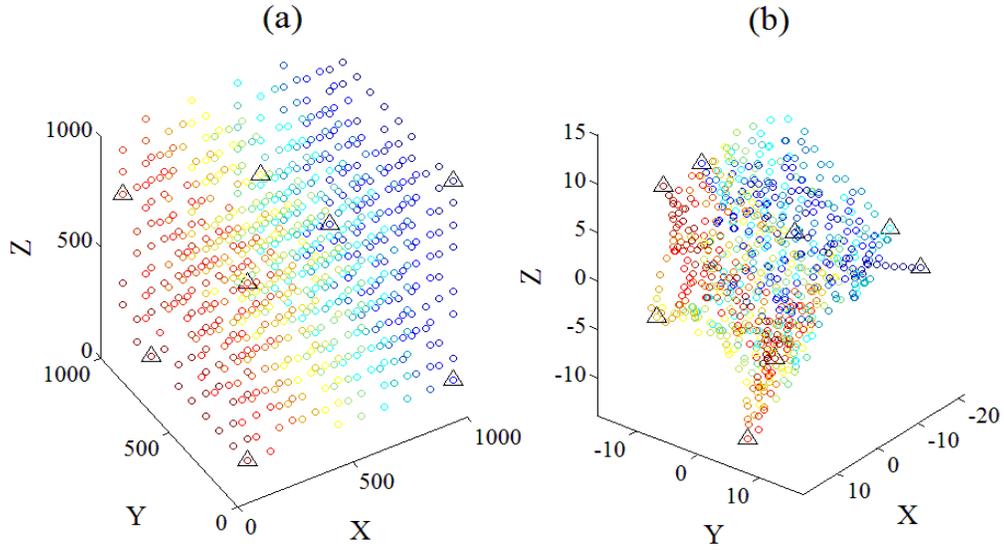


Figure 3.9: a) Geographic map of TN4(d) and b) generated TPM of TN4(d) with 8 ENS anchors

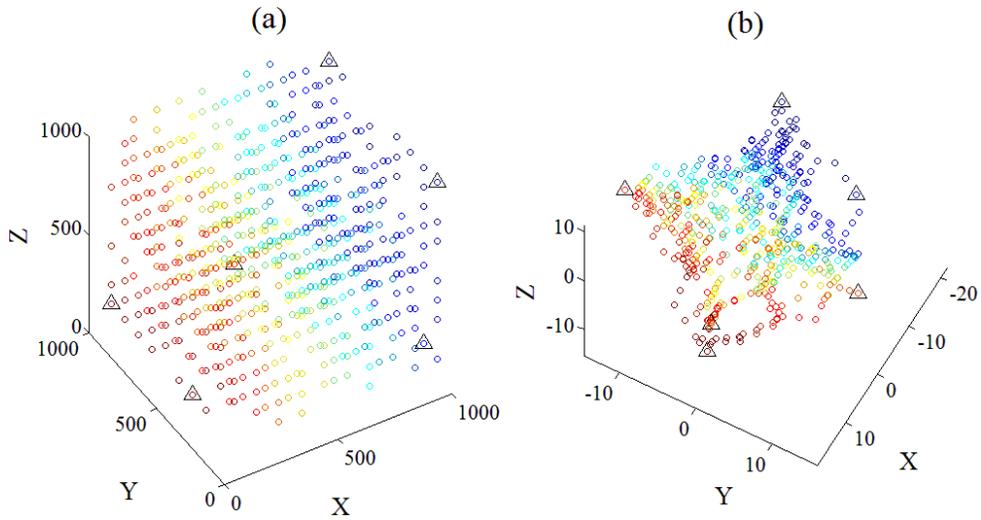


Figure 3.10: a) Geographic map of TN4(e) and b) generated TPM of TN4(e) with 6 ENS anchors

What's more, TN4(c) with 20% random nodes removed from full grid network is chosen for scalability related simulation for 3D-GLR algorithm. TN4(c) is scaled in 0.5times, 2times, 4times and 8 times of the volume of original TN4(c), which are named as TN4(f), TN4(g), TN4(h) and TN4(i). The sizes of TN4(f), TN4(g), TN4(h) and TN4(i) are given in Table 3.2. The number of nodes, number of ENS anchors and average node degree of TN4(f), TN4(g), TN4(h) and TN4(i) are given in Table 3.3.

Table 3.2: Size of TN4(f), TN4(g), TN4(h) and TN4(i)

Network Topology	Size
TN4(f)	800unit×800unit×800unit
TN4(g)	1200unit×1200unit×1200unit
TN4(h)	1600unit×1600unit×1600unit
TN4(i)	2000unit×2000unit×2000unit

Table 3.3: Number of nodes, number of ENS anchors and average node degree of TN4(f), TN4(g), TN4(h) and TN4(i)

	Network Topology			
	<i>TN4(f)</i>	<i>TN4(g)</i>	<i>TN4(h)</i>	<i>TN4(i)</i>
Number of nodes	411	1383	3276	6400
Number of ENS anchors	6	6	5	7
Average node degree	4.2	4.4	4.5	4.6

The last one Test Network5 (TN5)’s geographic map and generated TPM are shown in Figure 3.11 with 7 ENS anchors marked in black triangles. TN5 consists of 729 sensor nodes placed in random locations inside 1000unit×1000unit×1000unit cubic area with a random topology. The communication range of each sensor is 130units. TN5 is designed with two different average node degree. TN5(a) has high node degree of 7 and TN5(b) has low node degree of 4. Similarly, for each type of TN5, five random topologies are generated for simulation.

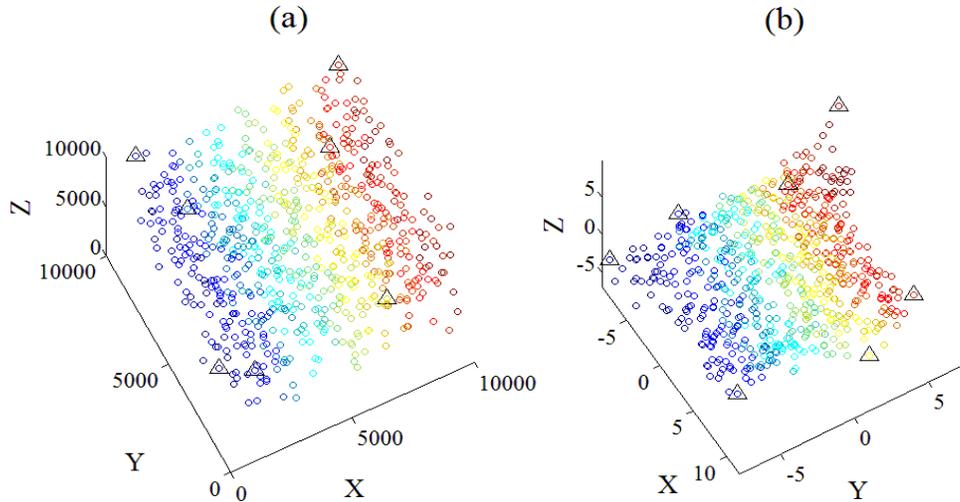


Figure 3.11: a) Geographic map of TN5 and b) generated TPM of TN5 with 7 ENS anchors

These five test networks include networks with fix topology (TN1, TN2 and TN3) and random topology (TN4 and TN5), in concave shape (TN3) and non-concave shape (TN1, TN2, TN3 and TN4), having physical voids inside network (TN1, TN2 and TN4) or not (TN3 and TN5) and in different scales (TN4). They are designed to fully evaluate the routing performance of 3D-GLR in various network environments. When simulating for evaluation metric average routability in eqn. (3.6), packets are generated from each sensor node in the network and routed to the rest of sensors in the network. For large-scale network like TN4(g), TN4(h) and TN4(i), packets are routed only to a certain number of random destination nodes from each node in the network. There are approximately 640,000 packets generated for each large-scale test network.

3.3.2 Effects of anchor selection on performance of 3D-GLR

In this section, we compare 3D-GLR algorithm using ENS, Double-ENS and random anchor selection placements, so as to validate the anchor selection effectiveness of ENS algorithm. Basically, in order to prove the effectiveness of ENS algorithm, the routing performance of 3D-GLR using ENS algorithm is expected to be better than the one using random topology. Additionally, the routing performance of 3D-GLR using Double-ENS algorithm with more extreme node anchors is expected not to outperform much than original ENS algorithm if original ENS algorithm already provides an optimal number of optimal anchor selection for the network.

In our simulation, random anchor placement shares the same number of anchors with ENS algorithm. Table 3.4 and Table3.5 show the number of anchor in TN1, TN2, TN3 and TN5 using different anchor placement. Double-ENS algorithm provides more extreme anchor node than ENS algorithm.

Table 3.4: Number of anchors in TN1, TN2, TN3 and TN5 using three anchor placements

Anchor Selection	Network Topology				
	<i>TN1</i>	<i>TN2</i>	<i>TN3</i>	<i>TN5(a)</i>	<i>TN5(b)</i>
ENS	6	9	6	8	8
Double-ENS	7	10	8	12	15
Random	6	9	6	8	8

Table 3.5: Number of anchors in TN4 with different sparseness using three anchor placements

Anchor Selection	Network Topology				
	<i>TN4(a)</i>	<i>TN4(b)</i>	<i>TN4(c)</i>	<i>TN4(d)</i>	<i>TN4(e)</i>
ENS	5	6	7	7	6
Double-ENS	8	7	8	8	8
Random	5	6	7	7	6

The routing performance of 3D-GLR algorithm using three anchor placements in TN1, TN2, TN3, TN4 and TN5 are shown in Figure 3.12, Figure 3.13 and Table 3.6 respectively. In all five types of networks, routing performance of 3D-GLR using ENS anchor selection greatly outperforms random anchor placement and almost the same with Double-ENS algorithm. For networks with low average node degree like TN1, TN2, TN3, TN4 and TN5(b). More extreme nodes chosen by Double-ENS algorithm provide almost the same routing performance for 3D-GLR algorithm, compared with original ENS algorithm. It also means the original ENS algorithm has already provided an optimal number of optimal anchors, to guarantee the effectiveness of anchors for routing performance. For networks with comparatively high average node degree like TN5(a), since more nodes within enlarged communication range will share the identical VCs due to the large node degree, the difficulty of finding extreme nodes with maximum/minimum DVC value increases, thus the amount of extreme node anchors may not be sufficient. As a result, using another DVC generated from another separate pair of random nodes will help in finding more extreme node anchors for networks. The conclusion is that both ENS and Double-ENS algorithm are able to provide good anchor selection for low connected networks however more anchors generated from Double-ENS algorithm will lead to more energy consumption, memory cost and overhead compared with ENS algorithm. For networks with high average node degree or where ENS algorithm cannot generate enough number of anchors as required, Double-ENS algorithm is a good choice to provide more extreme node anchors.

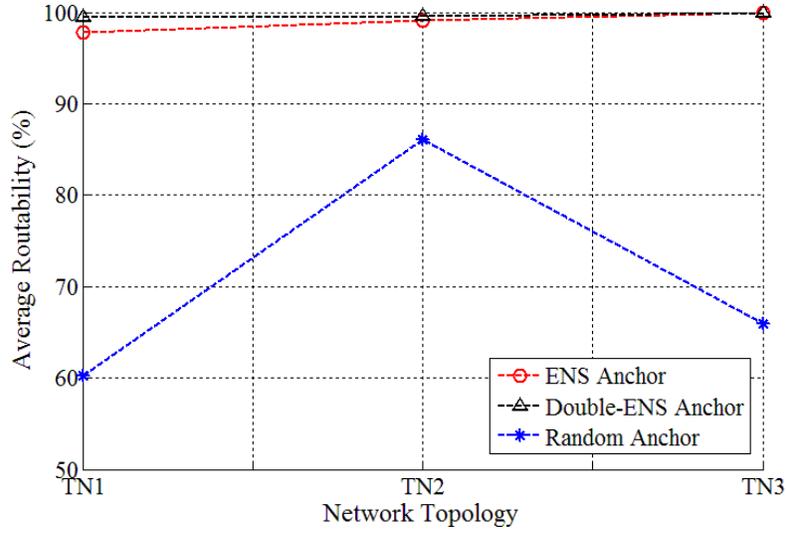


Figure 3.12: Average routability of 3D-GLR in TN1, TN2 and TN3 using three anchor placements

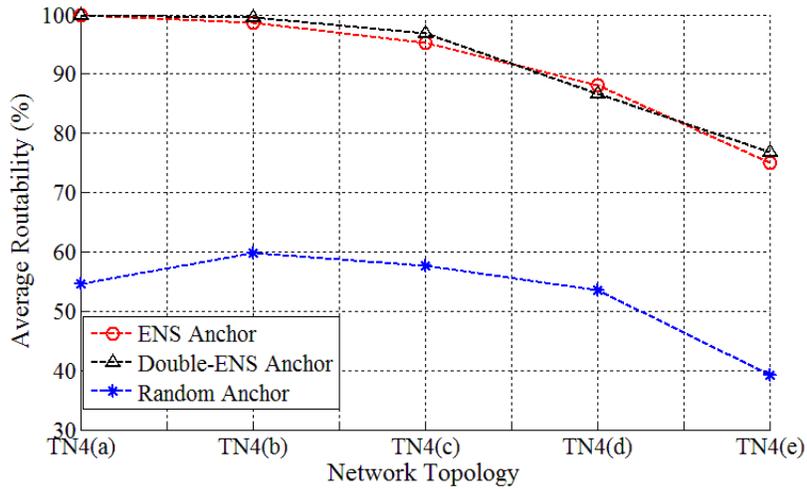


Figure 3.13: Average routability of 3D-GLR in TN4 with different sparseness using three anchor placements

Table 3.6: Average routability of 3D-GLR in TN5 with different average node degree using three anchor placements

Anchor Selection	Network Topology	
	<i>TN5(a)</i>	<i>TN5(b)</i>
ENS	81.86%	57.03%
Double-ENS	91.07%	60.26%
Random	60.23%	46.87%

3.3.3 Routing performance comparison between 3D-GLR and 3D-GDSTR

In this section, the performance comparison between 3D-GLR using anchors selected from ENS algorithm and GCs based 3D-GDSTR is presented. For comparison algorithm 3D-GDSTR, we continue to use two 2D hull trees and maximum of 5 children for each node in hull trees (generated from minimum spanning trees), which are proposed in [56].

3.3.3.1 Routing performance comparison in TN1, TN2 and TN3

The routing simulating results of 3D-GLR and 3D-GDSTR algorithm in TN1, TN2, and TN3 are shown in Table 3.7. Comparing 3D-GLR algorithm with 3D-GDSTR algorithm, 3D-GLR algorithm outperform the existing 3D-GDSTR algorithm in all three network types, achieving almost 100% average routability. It's reported that 3D-GDSTR can achieve full success delivery rate when the network size is small (less than 200 nodes) and average node degree is comparatively large (more than 10) [56]. However for 3D-GDSTR algorithm in TN3, routing performance of GCs based greedy forwarding can hardly be improved by increasing node degree because TN3 is designed to be in concave shape in which packet is easily stuck in local minima and thus GCs based greedy forwarding performance will be severely deteriorated. We increase the node degree from 5 to 13 for TN3 and the average routability of 3D-GDSTR is increased only to 89.45% from 87.53%, still lower than average routability of 3D-GLR without increasing node degree. Generated TPM in Figure 3.4(a) of concave-shaped TN4 somehow decreases the concave level of the network so that a neighbor node closer to the destination may be found. We also test 3D-GDSTR algorithm in TN3 using generated topology from TPM, in which 3D GCs are all replaced by generated 3D TCs. Without any increased node degree, TCs based 3D-GDSTR can achieve 93.76% average routability compared with GCs based 3D-GDSTR whose routability is 87.53%. High routing performance of 3D-GLR algorithm benefits not only from the combination of three routing modes in different coordinate domains but also from a more proper network topology representation for routing in topological domain for some types of networks like TN3, compared with original network topology in geographic domain.

Table 3.7: Average routability of 3D-GLR and 3D-GDSTR in TN1, TN2 and TN3

Routing Algorithm	Network Topology		
	<i>TN1</i>	<i>TN2</i>	<i>TN3</i>
3D-GLR	97.81%	99.12%	100.00%
3D-GDSTR	87.23%	83.30%	87.53%

3.3.3.2 Routing performance comparison in TN4 and TN5

In this section, we gradually decrease the average node degree of network using TN4(a), TN4(b), TN4(c), TN(d) and TN(e) and compare the routing performance of 3D-GLR and 3D-GDSTR in these five TN4 networks. The average routability of 3D-GLR in five TN4 with decreased average node degree is shown in Figure 3.14.

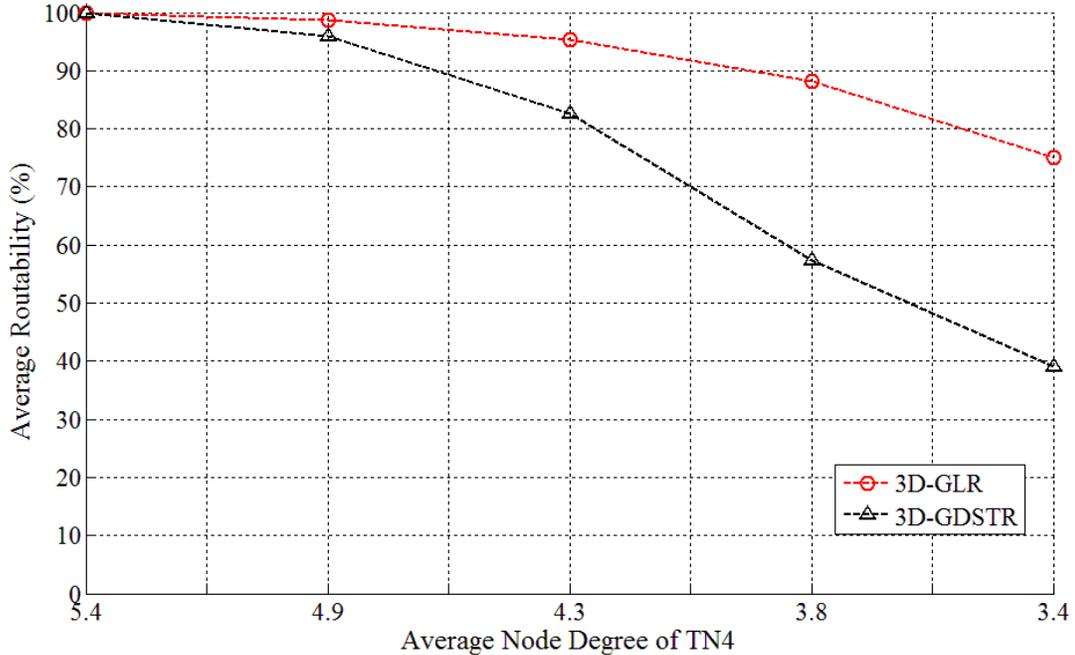


Figure 3.14: Average routability of 3D-GLR and 3D-GDSTR in TN4 with decreased average node degree

The routing performances of 3D-GLR and 3D-GDSTR algorithm are both affected by the decreased node degree since the connectivity of networks are deteriorated gradually. When TN4 is fully connected, both 3D-GLR and 3D-GDSTR can achieve full average routability. Average routability of 3D-GDSTR decreases drastically as the average node degree decreases, only achieving 57.37% and 39.04%

with low network connectivity (average node degree of 3.8 and 3.4). However, 3D-GDSTR can still achieve comparatively satisfactory average routability of 88.14% and 74.96% at such low average node degree. Compared with 3D-GDSTR, 3D-GLR not only achieves much better routing performance but also is more insensitive to the decrease of node degree of the network and provides a more stable routing performance as the node degree of the network decreases.

We also test the routing performances of another random topology network TN5 with high node degree 7 (TN5(a)) and low node degree 4 (TN5(b)) and the results are shown in Table 3.8. 3D-GLR still outperforms 3D-GDSTR in networks with nodes in random locations and topology. 3D-GDSTR shows very poor routing performance when the random topology network has low average node degree however 3D-GLR is able to provide a comparatively satisfactory routing performance.

Table 3.8: Average routability of 3D-GLR and 3D-GDSTR in TN5 with different average node degree

Routing Algorithm	Network Topology	
	<i>TN5(a)</i>	<i>TN5(b)</i>
3D-GLR	81.86%	59.89%
3D-GDSTR	81.34%	18.80%

3.3.3.3 Routing performance comparison in scaled TN4

In this section, we compare the routing performance of 3D-GLR and 3D-GDSTR in different scales of TN4(c) in which 20% random nodes are removed from full grid networks using TN4(f), TN4(g), TN4(h) and TN4(i), to explore the scalability of 3D-GLR algorithm. The size of network ranges from 800unit \times 800unit \times 800unit to 2000unit \times 2000unit \times 2000unit. The number of nodes ranges from approximately 400 to 6400. Figure 3.15 shows the average routability of five scaled TN4 using 3D-GLR and 3D-GDSTR algorithm. As we can see from Figure 3.15, the average routability of 3D-GLR stays the same around 95% in networks with 0.5times, 2 times, 4 times and 8 times volume of the original TN4(c). Moreover, the stable 95% average routability from 3D-GLR also greatly outperforms 3D-GDSTR algorithm which can only provide 82.94% average routability at the best. Simulation results show that 3D-GLR can provide satisfactory routing performance with great adjustability and scalability. For 3D-

GDSTR algorithm, generating multiple spanning/hull trees and increasing children number for each node can help in improving routing performance but this will also introduce large cost in both computation and power, which becomes impractical for large-scale WSNs. Compared with 3D-GDSTR, 3D-GLR algorithm shows more flexibility, adjustability and most importantly, better routing performance in different network environments. The generation of VCs and the computation of SVD components can be easily applied in 3D WSNs from 2D WSNs. From our simulations, VCS based ENS and 3D-GLR algorithms show great effectiveness and potential for routing application in 3D WSNs.

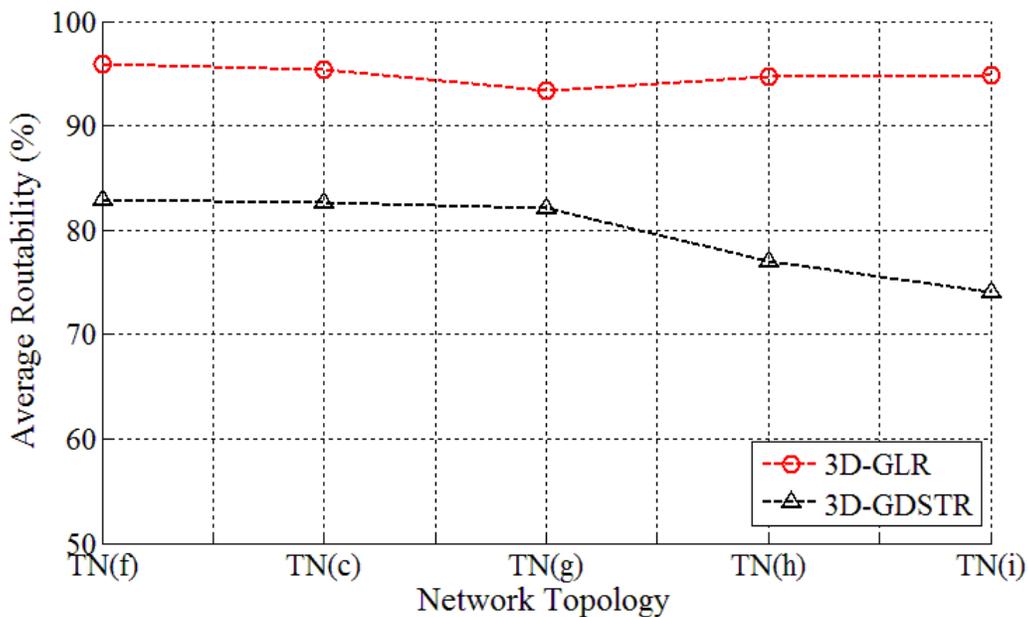


Figure 3.15: Average routability of 3D-GLR and 3D-GDSTR in TN4 with different scales

3.4 Summary

In this chapter, the extensions of 2D-GLR to 3D sensor networks have been accomplished. From simulation in five designed networks, the routing performance of 3D-GLR with ENS anchor placement show great advantage over current existing GC based routing algorithm 3D-GPSTR. The effectiveness of 3D-GLR algorithm is of great significance for 3D networks like USNs where geographic localization equipment and algorithm can hardly work.

CHAPTER 4

VIRTUAL COORDINATE GENERATION SCHEMES FOR MOBILE SENSOR

As stated in Chapter 1, mobile sensors can be seen as new sensors dynamically deployed in original network. They can be in placed manually by human labor or dropped from mobile vehicle or plane in one place for certain time (hours or days, etc.) for monitoring tasks or moving in very slow velocity. In this situation, mobile sensor node can be treated as static nodes newly inserted in a network and velocity can be ignored when considering communication of mobile sensors in the network. VCs for mobile node need to be generated properly at the first place so that location information of mobile nodes in virtual domain can be known by the network for applications such as VC based routing. VCs can be generated by network' flooding messages again from anchors but this will lead to expensive energy, time and message cost in large-scale WSNs. Thus simple re-flooding for mobile sensor is not an efficient VC generation method. Normally, sensors are equipped with radio and can communicate within a certain range. Mobile sensors are aware of their static neighbor sensor nodes within communication range and can obtain VCs from simple computation of neighboring nodes' coordinates in order to estimate their location in virtual domain. This is analogous to RSSI technique in geographic domain but only approximate virtual location will be obtained for mobile sensor.

In general, newly inserted mobile sensor nodes play three roles in routing: sending packets, receiving packets and passing packets to other sensor nodes. In this thesis, we only focus mobile sensors' receiving packets and passing packets performance. When mobile sensor receives packets, generated VCs can have great effects on calculating greedy forwarding distance [13]. Improper VC generation scheme can lead to very low receiving packet capability for mobile sensor nodes especially when mobile sensor nodes are located in a special position which we name as 'Ring' position. In 'Ring' position, the node is surrounded by neighbors whose gap is greater than two hops which can be seen in Figure 4.1(b). As in Figure 4.1(b), new sensor node marked in green color has two anchor neighbor nodes whose VCs are (0,6) and (6,0). Normally, if the communication range of sensor node is one unit, the largest distance between

node's neighbors should be no more than 2 units, which is shown in Figure 4.1(a). If this distance is more than two units, a gap in virtual domain between neighbors exists, which might result from the imperfectness or void of the network. As a result, two nodes with physical distance of two units are actually far away from each other in virtual domain. This distance mismatch may lead to poor routing performance for the new sensors in such position due to the local minima problem if improper VCs are generated for new sensors. In the example shown in Figure 4.1(b), if new sensor node in green color takes average of neighboring sensor nodes' VCs which is (3,3), a local minima exists when new sensor node tries to receive packets from static sensor node in red color whose VCs are (3,3). When passing packet in the network, new sensor nodes should cause as less negative effects to the entire network as possible. For example, the average routing path length of the original network shouldn't be increased much and the average routing performance shouldn't be impaired much.

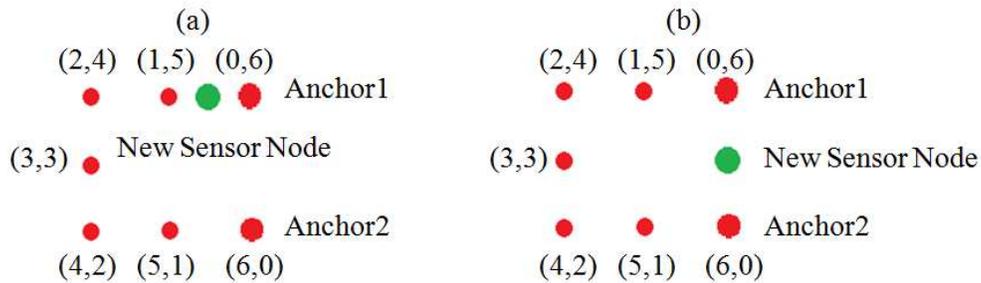


Figure 4.1: a) Example of new inserted mobile sensor in regular position and b) example of new inserted mobile sensor in “Ring” position

The organization of Chapter 4 is as followed. Three simple VC generation schemes for mobile sensors are introduced in Section 4.1. Simulation results for VC generation schemes are given in Section 4.2. More specifically, evaluation matrix is described in Section 4.2.1. Simulations results for 2D and 3D networks are analyzed in Section 4.2.2 and Section 4.2.3 correspondingly. Simple summary is given in Section 4.3

4.1 Virtual coordinate generation schemes for mobile sensor

Simple, energy and time efficient VC generating scheme would be ideal for newly inserted mobile sensors. Thus making advantages of neighbors and computing new VCs from neighboring sensors' VCs is our main idea. Three schemes are proposed to generate VCs and they are: Average Scheme, Mixture Scheme, and Minimum Scheme. Here are detailed descriptions of the three schemes to generate VCs for new sensor nodes:

- a. Average Scheme (Average): Take average of neighboring sensor nodes' VCs to the corresponding anchor as new node's VCs.
- b. Mixture Scheme (Mixture): Use average scheme when new node is located in regular position. When new node is located in 'Ring' position, firstly, it selects the neighbor with minimum sum of VCs and uses this neighbor's VCs as new node's VCs. Secondly, a random anchor is selected, plus one to new node's VC corresponding to this anchor.
- c. Minimum Scheme (Minimum): Take the minimum value of neighboring sensor nodes' VCs and then plus one to the corresponding anchor as new node's VCs.

Average Scheme is easy to understand and it basically takes the average location for inserted new node among the surrounding neighbors. To avoid identical coordinate problem, Average Scheme will add one to VC corresponding to a randomly picked anchor if and only if there is only one neighbor around the new mobile sensor node. Minimum Scheme is very like flooding process from anchor set. Mixture Scheme is designed based on Average Scheme. For new node in 'Ring' position, Mixture Scheme keeps the new node close to the neighbor which is comparatively nearest to anchors than other neighbors, instead of keeping new node in the middle of the gap in virtual domain. Figure 4.2 and Table 4.1 gives a simple example of three schemes, in which mobile sensor in green color is surrounded by four neighbor nodes in red color.

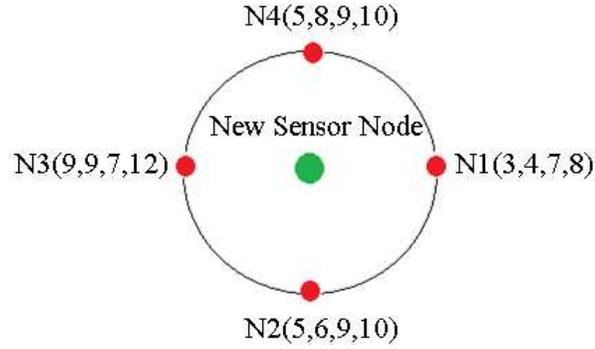


Figure 4.2: Example of newly inserted mobile sensor and its neighboring sensors

Table 4.1: Example of generated VCs for new node using three schemes

VC generation Scheme	Virtual Coordinates of New Node
Average Scheme	(5.5, 6.75, 8, 10)
Mixture Scheme	(4, 4, 7, 8)
Minimum Scheme	(4, 5, 8, 9)

The above schemes can be used in both 2D and 3D VCS based WSNs. Since VCs are generated from hop distances in multi-dimensional space and are independent on geographic dimension, sensor network with 2D or 3D geographic topology doesn't affect estimating VCs for mobile sensor nodes.

4.2 Simulation and discussion

4.2.1 Evaluation matrices

Average routability and average path length are two matrices for evaluating VC generation schemes' effectiveness in routing performance for mobile sensor and the network. The definition of them are given in eqn. (3.1) and (3.2). Routing performances of newly inserted mobile sensors include two parts: the average routability and path length of new nodes when new sensors receive packets from network. The other part is the average routability and path length of the original network when new nodes just play passing packets role in the original network. The selected routing algorithm for simulation is LCR algorithm [13] which is based on greeding forwarding in virtual domain.

$$\text{Average Routability [16][20][21]} = \frac{\text{Total number of packet reached destination}}{\text{Total number of packet generated}} \times 100\% \quad (4.1)$$

$$\text{Average Path Length [16][20][21]} = \frac{\text{Cumulative number of hops that each packet traversed}}{\text{Total number of packet generated}} \times 100\% \quad (4.2)$$

Since generated VCs come from neighboring nodes' VCs, new node's routing performance can be expected to be close to or better than its neighbors' routing performance if VC generation scheme is proper and effective. Also, when new mobile sensors are inserted, the routing performance of the original network should at least remain the same and could be even improved by the inserted new nodes.

4.2.2 Simulation results for 2D networks

For 2D network, 4 types of network are used in our simulation which can be seen in Figure 4.3. The number of nodes ranges from 300 to 800. They are: (a) Circular network with three voids of 496 nodes (Circle) [16][17][19][20][21], (b) Grid network with 100 holes network of 800 nodes (Grid with 100 Holes) [20], (c) Grid network with 200 holes network of 700 nodes (Grid with 200 Holes), (d) Pipe network with 368 nodes (Pipe). Holes in networks in Figure 4.3(b) and 4.3(c) are randomly picked. In Figure 4.3, nodes marked as red triangle are anchors selected by ENS algorithm.

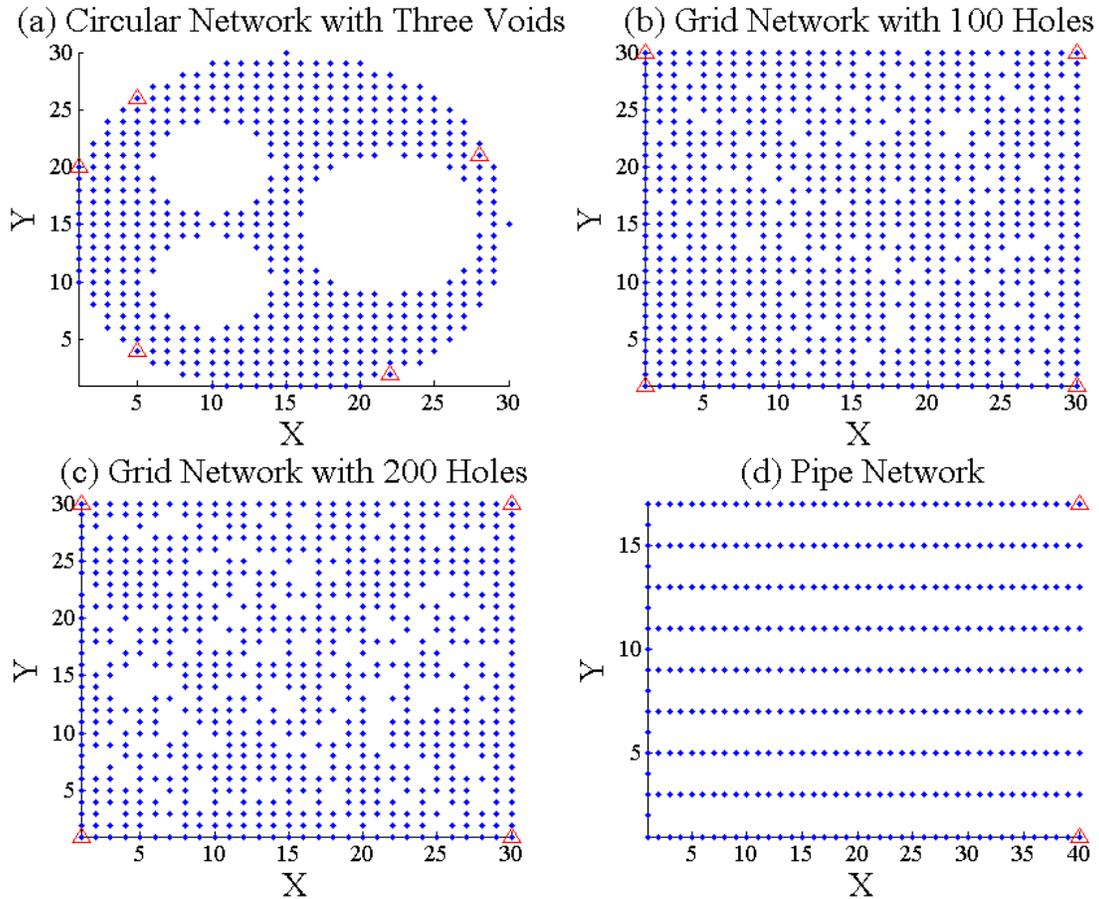


Figure 4.3: a) Circular network with three voids of 496 nodes; b) Grid with 100 holes network of 800 nodes; c) Grid with 200 holes network of 700 nodes and d) Pipe network of 368 nodes

For each network, anchor selection using ENS algorithm and random anchor placement (10 random anchors) are both used for routing performance simulation. In our simulation, we place 200 new sensor nodes randomly in circular network with three voids, 100 new nodes in the hole locations in grid with 100 holes network, 197 new nodes in the hole locations in grid with 200 holes network and 312 new nodes in the slot locations in pipe network. In circular network with three voids, new nodes are all located in regular positions. In the other three networks, new nodes are located in both regular position and ‘Ring’ position. The ratio of regular position over ‘Ring’ position varies in these three networks. Grid with 100 holes network has the smallest portion of ring positions for new nodes and pipe network has the largest

portion of ring positions for new nodes. The inserted new sensor nodes in four networks can be seen in Figure 4.4. The red star nodes are new inserted mobile sensors in four types of network.

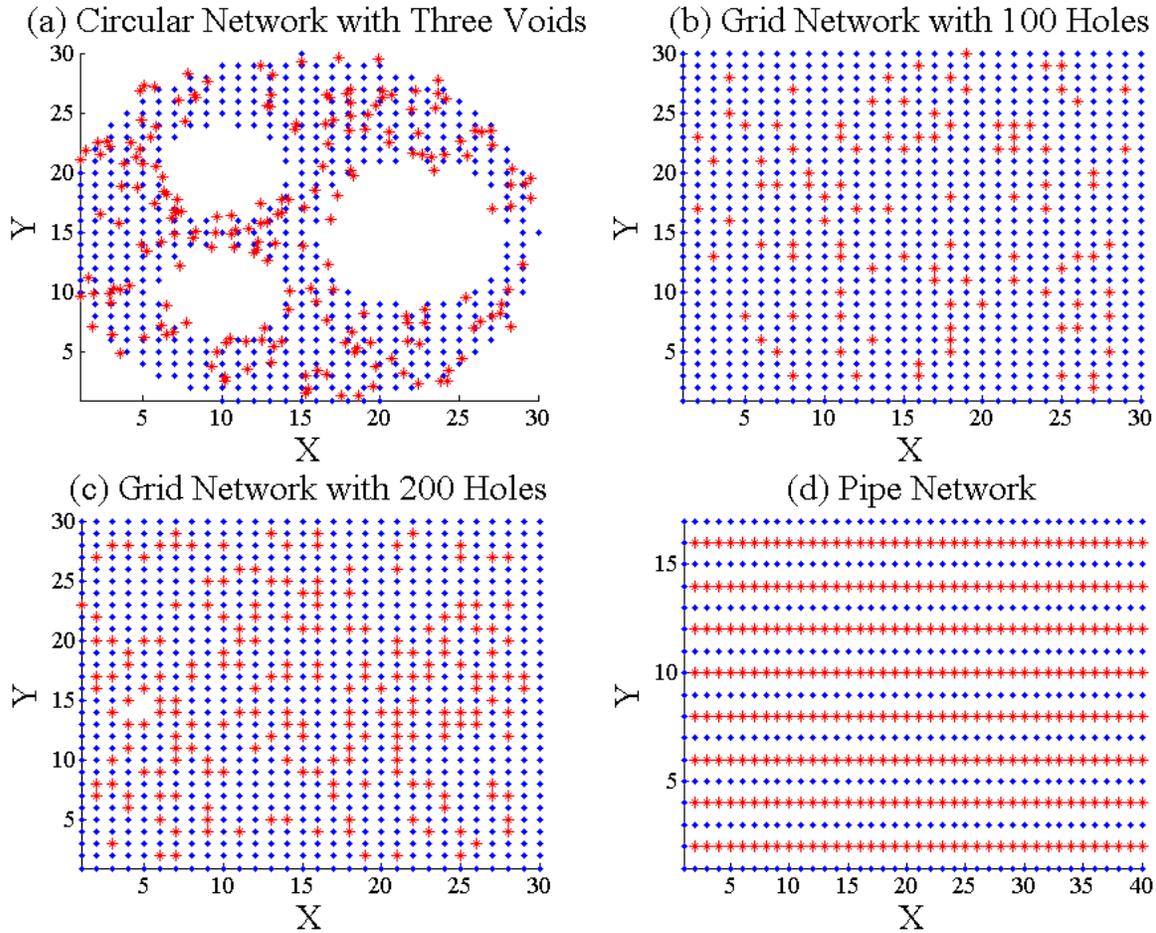


Figure 4.4: a) Circular network with three voids and 200 testing new nodes; b) Grid with 100 holes network and 100 testing new nodes; c) Grid with 200 holes network and 197 testing new nodes and d) Pipe network and 312 testing new nodes

4.2.2.1 Mobile sensors receive packets from 2D networks

In this section, the simulation results of average routability and path length of new nodes receiving packets in four test networks using ENS anchors are shown in Figure 4.5(a) and 4.5(b). Results for networks using 10 random anchors are shown in Figure 4.6(a) and Figure 4.6(b). Each new node acts as the destination receiving packets from other static nodes in the original networks. The average routability and path length are calculated by taking average of results obtained from each new testing sensor node using

three VC generation schemes, compared with average routability of original network without introduction of new sensors.

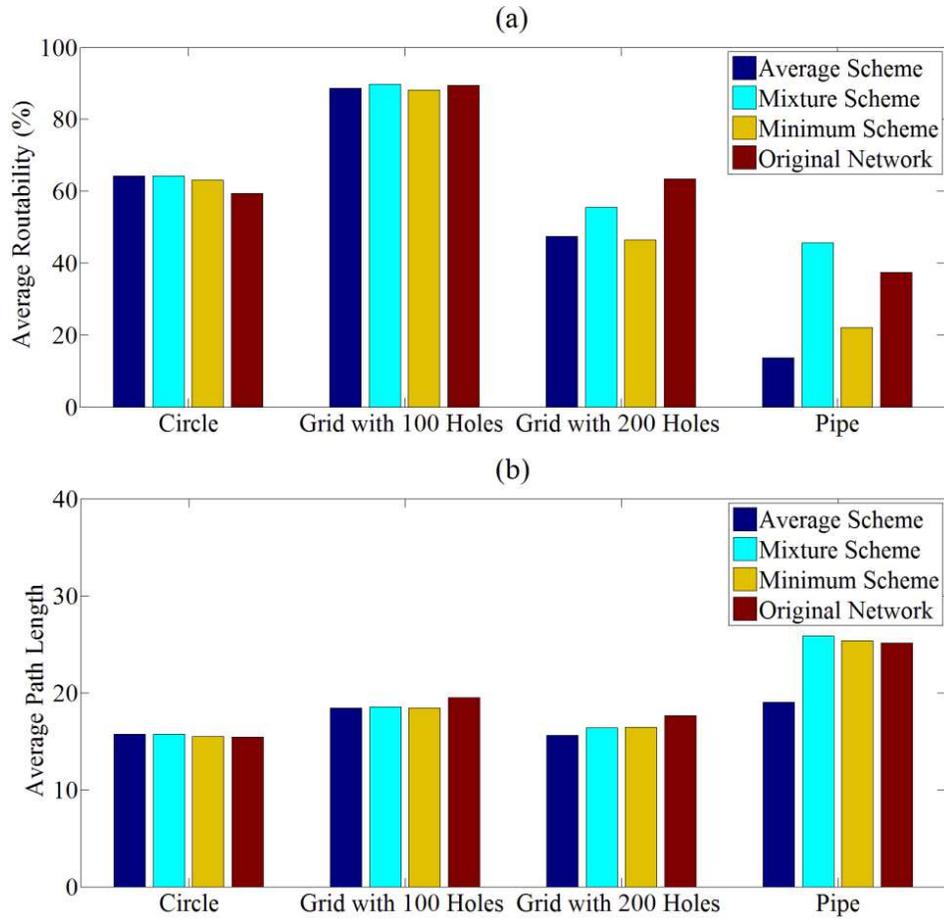


Figure 4.5: a) Average routability of new inserted mobile sensors in four networks with ENS anchor placement and b) average path length of new inserted mobile sensors in four networks with ENS anchor placement

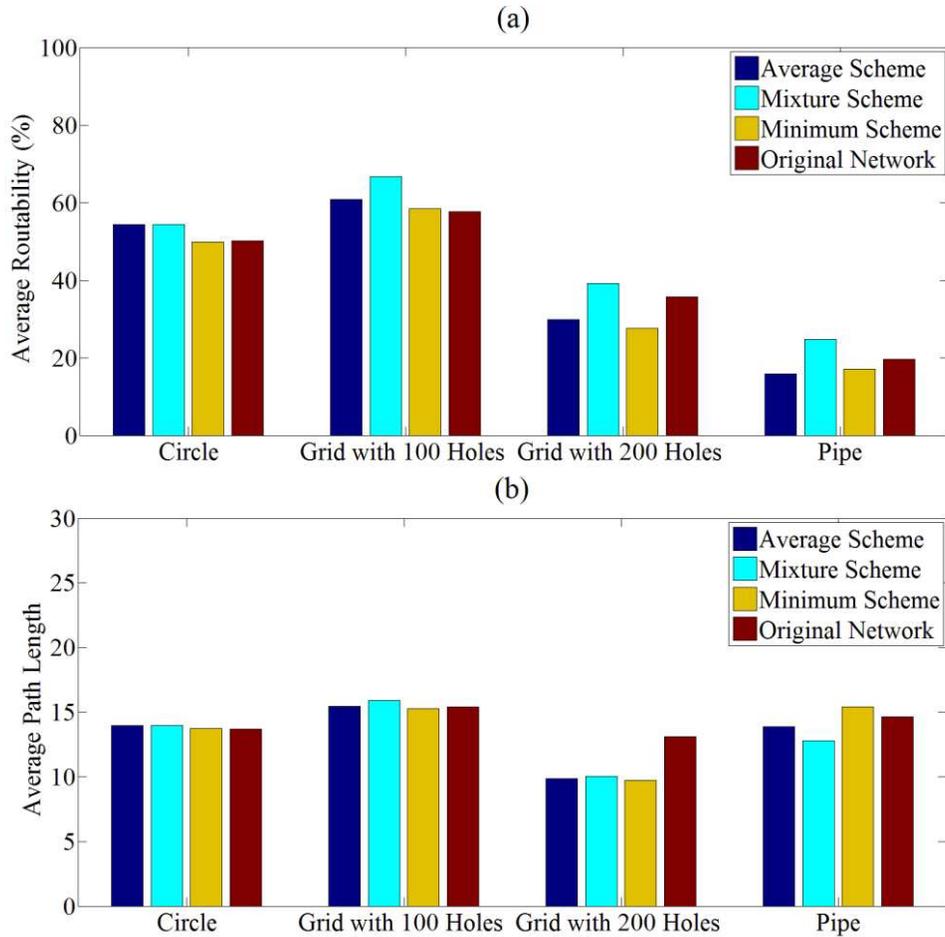


Figure 4.6: Average routability of new inserted mobile sensors in four networks with random anchor placement and b) average path length of new inserted mobile sensors in four networks with random anchor placement

From our simulations, we can see in regular locations like the testing new sensor nodes in circular network with three voids, new sensor nodes using Average scheme, Mixture scheme and Minimum Scheme can reach the almost same routability of the original network and even better. But in ‘Ring’ position which exists in the other three networks, Average Scheme and Minimum Scheme do not work well and routability of new nodes decreases. However, new nodes using Mixture Scheme can reach desirable routability. In simulations of four test networks with ENS anchor set and random anchor set, new nodes in Mixture Scheme can reach the routability of the original network except in grid with 200 holes network using ENS anchors but still better the other two schemes. Also, the average path length of

Mixture Scheme is less than the average path length of the original network. In all, when mobile nodes are inserted and receive packets from the network, Mixture Scheme can be good choice for VC generation.

4.2.2.2 Mobile sensors pass packets in 2D network

In simulation for this section, we set number of newly inserted mobile sensor nodes as 1%, 2%, 4%, 6%, 8% and 10% of total number of static nodes in original network. A certain number of new nodes will be inserted into the original static networks. The locations are randomly selected from new mobile nodes' locations from Figure 4.5. All new nodes act as intermediate nodes passing packets to other static nodes in the original network. The average routability and path length are calculated by taking average of results obtained from all static nodes in the original network.

Table 4.2, Table 4.3, Table 4.4 and Table 4.5 show the average routability of four types of networks in Figure 4.3(a), 4.3(b), 4.3(c) and 4.3(d) correspondingly. Table 4.6, Table 4.7, Table 4.8 and Table 4.9 show the average path length of our types of networks in Figure 4.3(a), 4.3(b), 4.3(c) and 4.3(d) correspondingly. From simulations results, we can see that as more new nodes are inserted, more changes might be brought to the network. When a very small number of new nodes are inserted like 1% and 2%, the average routability of static nodes in original network remain almost the same. When a comparatively larger number of new nodes are inserted like 10%, the average routability of original network can be greatly decreased if using improper schemes or be improved by using proper schemes for inserted new nodes. In regular locations, average routability of circular network with three voids with 10% inserted new mobile sensor nodes in Average Scheme, Mixture Scheme and Minimum Scheme remain almost the same with the original network. However, in other 3 networks which have 'Ring' position, Mixture Scheme brings drastic decrease for routability of the original network. In this case, Average Scheme can make new mobile sensor nodes behave like a bridge connecting two neighbors far away from each other in virtual domain. Mixture Scheme can hardly do that because it is designed to make new sensor node very close to one of its neighbors. Compared with Minimum Scheme, Average Scheme has less average path length especially when 'Ring' position problem is severe like pipe network, which can be seen in

Table 4.9. In all, when mobile nodes pass packets, Average Scheme can be a good choice. Average routability and path length of four types of networks using 10 random anchors show very similar trend to networks using ENS anchors.

Table 4.2: Average routability of original circular network with three voids with different percentage of inserted new nodes

% New nodes	ENS anchor placement				Random anchor placement			
	Network	Average	Mixture	Minimum	Network	Average	Mixture	Minimum
1%	59.41%	59.34%	58.50%	59.10%	50.16%	50.05%	49.99%	49.81%
2%	59.41%	59.23%	59.27%	59.22%	50.16%	50.22%	49.62%	49.98%
4%	59.41%	60.04%	59.80%	58.67%	50.16%	50.04%	49.93%	49.05%
6%	59.41%	60.11%	59.91%	58.40%	50.16%	50.03%	50.03%	48.66%
8%	59.41%	59.43%	59.49%	58.10%	50.16%	49.69%	49.84%	47.46%
10%	59.41%	60.98%	60.55%	59.95%	50.16%	50.17%	50.10%	46.16%

Table 4.3: Average routability of original grid with 100 holes network with different percentage of inserted new nodes

% New nodes	ENS anchor placement				Random anchor placement			
	Network	Average	Mixture	Minimum	Network	Average	Mixture	Minimum
1%	89.48%	90.80%	89.88%	90.67%	57.61%	58.14%	54.62%	58.18%
2%	89.48%	91.82%	90.78%	91.82%	57.61%	58.75%	55.66%	58.86%
4%	89.48%	92.19%	90.16%	91.82%	57.61%	58.16%	50.67%	58.58%
6%	89.48%	94.78%	85.64%	93.83%	57.61%	59.47%	51.04%	59.64%
8%	89.48%	95.71%	84.43%	94.91%	57.61%	60.67%	48.51%	60.72%
10%	89.48%	97.50%	87.10%	96.23%	57.61%	60.88%	47.40%	61.36%

Table 4.4: Average routability of original grid with 200 holes network with different percentage of inserted new nodes

% New nodes	ENS anchor placement				Random anchor placement			
	Network	Average	Mixture	Minimum	Network	Average	Mixture	Minimum
1%	63.33%	63.84%	62.12%	63.41%	35.70%	35.62%	35.16%	35.63%
2%	63.33%	63.35%	59.95%	61.92%	35.70%	35.58%	33.87%	35.79%
4%	63.33%	66.51%	57.98%	64.81%	35.70%	35.11%	32.37%	35.63%
6%	63.33%	62.61%	51.84%	60.64%	35.70%	34.40%	30.54%	34.65%
8%	63.33%	66.75%	54.13%	64.53%	35.70%	34.47%	30.60%	35.73%
10%	63.33%	62.90%	50.18%	59.76%	35.70%	34.11%	29.06%	35.31%

Table 4.5: Average routability of original pipe network with different percentage of inserted new nodes

% New nodes	ENS anchor placement				Random anchor placement			
	Network	Average	Mixture	Minimum	Network	Average	Mixture	Minimum
1%	37.42%	36.67%	37.27%	37.36%	19.66%	19.95%	20.42%	20.96%
2%	37.42%	37.41%	38.12%	37.98%	19.66%	20.41%	20.48%	21.20%
4%	37.42%	38.43%	39.17%	38.97%	19.66%	21.17%	21.60%	23.27%
6%	37.42%	38.33%	38.35%	39.70%	19.66%	21.79%	21.65%	23.82%
8%	37.42%	36.78%	38.94%	39.07%	19.66%	22.77%	22.00%	25.82%
10%	37.42%	39.47%	38.45%	41.67%	19.66%	23.99%	23.25%	27.69%

Table 4.6: Average path length of original circular network with different percentage of inserted new nodes

% New nodes	ENS anchor placement				Random anchor placement			
	Network	Average	Mixture	Minimum	Network	Average	Mixture	Minimum
1%	15.40	15.37	15.20	15.26	13.69	13.67	13.67	13.65
2%	15.40	15.37	15.39	15.40	13.69	13.68	13.69	13.61
4%	15.40	15.44	15.37	14.99	13.69	13.70	13.68	13.52
6%	15.40	15.55	15.57	15.12	13.69	13.67	13.65	13.37
8%	15.40	15.41	15.43	15.16	13.69	13.63	13.67	13.22
10%	15.40	15.61	15.46	15.36	13.69	13.75	13.71	12.95

Table 4.7: Average path length of original grid with 100 holes network with different percentage of inserted new nodes

% New nodes	ENS anchor placement				Random anchor placement			
	Network	Average	Mixture	Minimum	Network	Average	Mixture	Minimum
1%	19.50	19.58	19.39	19.56	15.41	15.48	14.88	15.47
2%	19.50	19.69	19.63	19.68	15.41	15.54	14.94	15.55
4%	19.50	19.62	19.29	19.56	15.41	15.39	13.92	15.43
6%	19.50	19.83	18.57	19.70	15.41	15.69	14.03	15.62
8%	19.50	19.92	18.26	19.80	15.41	15.70	13.48	15.65
10%	19.50	20.05	18.65	19.86	15.41	15.73	13.14	15.72

Table 4.8: Average path length of original grid with 200 holes network with different percentage of inserted new nodes

% New nodes	ENS anchor placement				Random anchor placement			
	Network	Average	Mixture	Minimum	Network	Average	Mixture	Minimum
1%	17.66	17.92	17.46	17.70	13.12	13.14	12.93	13.13
2%	17.66	17.53	16.91	17.25	13.12	12.83	12.46	12.91
4%	17.66	18.11	16.60	17.70	13.12	12.83	12.05	12.94
6%	17.66	17.40	15.16	16.95	13.12	12.49	11.51	12.55
8%	17.66	17.83	15.93	17.16	13.12	12.57	11.41	12.73
10%	17.66	17.19	14.92	16.43	13.12	12.40	11.14	12.60

Table 4.9: Average path length of original pipe network with different percentage of inserted new nodes

% New nodes	ENS anchor placement				Random anchor placement			
	Network	Average	Mixture	Minimum	Network	Average	Mixture	Minimum
1%	25.13	23.50	24.94	24.96	14.62	13.55	14.65	14.50
2%	25.13	23.86	25.10	25.07	14.62	13.87	14.66	14.51
4%	25.13	24.18	25.36	25.32	14.62	12.57	14.73	14.42
6%	25.13	22.36	24.66	25.05	14.62	12.28	14.55	14.41
8%	25.13	20.50	23.45	24.84	14.62	11.81	14.34	14.35
10%	25.13	20.97	22.80	24.94	14.62	11.61	14.70	14.30

4.2.3 Simulation results for 3D networks

For 3D network, a volume network model in 1000unit × 1000unit × 1000unit cubic area is developed, which is similar to USNs. 712 sensor nodes are randomly placed in the volume cube area. The communication range of sensors is 130units. The network is shown in Figure 4.7(a) and 8 ENS anchors are marked in red triangle using Double-ENS algorithm. 200 mobile nodes are randomly placed in network area and they are located in both regular positions and “Ring” positions. The inserted new mobile nodes are marked as red star in Figure 4.7(b). Again, in simulation of average routability and path length, we choose 8 ENS anchors and 15 random anchors for test network.

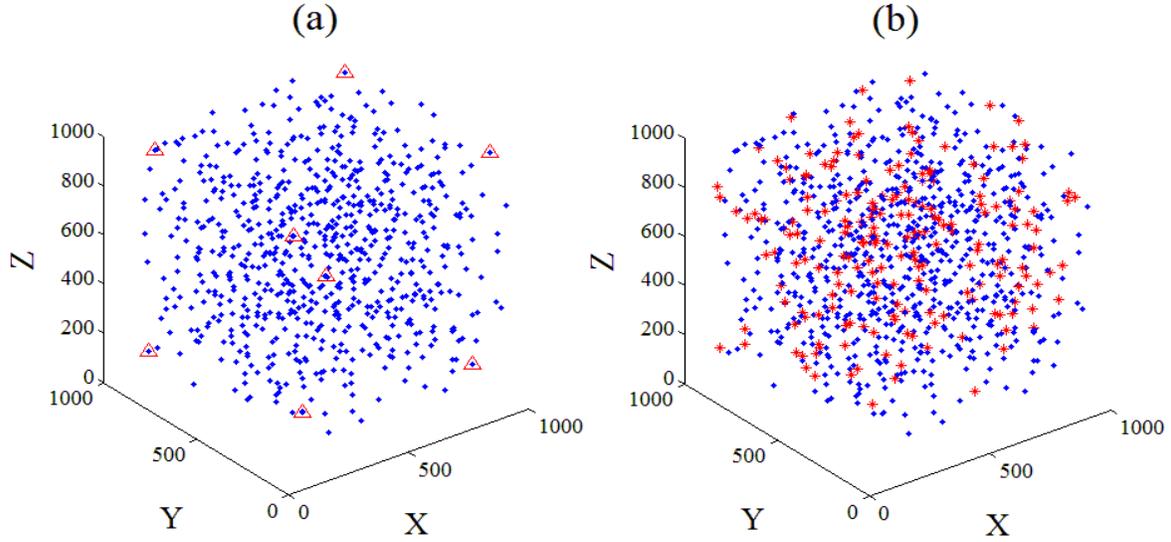


Figure 4.7: a) 3D USN of 712 sensor nodes with 8 ENS anchors and b) 200 new mobile sensors inserted in 3D USN

4.2.3.1 Mobile sensors receive packets from 3D network

In 3D network, mobile sensors are considered as static to receive packets from other nodes in the network. The average routability and path length of new mobile sensors receiving packets in 3D networks are shown in Figure 4.8(a). The simulation results are quite similar to the results in 2D network. Mixture scheme still out performs than the other two schemes in new mobile sensors' routability, both in ENS anchors and random anchors based network. Also, Mixture scheme shows shorter average path length compared with other two schemes.

4.2.3.2 Mobile sensors pass packets in 3D network

In our simulation, we still set number of new testing nodes as 1%, 2%, 4%, 6%, 8% and 10% of total number of network static nodes which are inserted to the original network. The locations are randomly selected from locations in Figure 4.7(b). When mobile sensors only pass packets in network, Table 4.10 and Table 4.11 give the average routability and path length of original network. Also same to 2D network simulation results, Average Scheme shows good performance when new nodes are inserted as intermediate nodes passing packets to other nodes in network. When a comparatively small number of mobile nodes which are new to the network are inserted, the impacts on routing performance are little

when proper VC generation scheme is selected. From simulation in both ENS and random anchor set based 3D networks, results for routing performance share the same trend with 2D network.

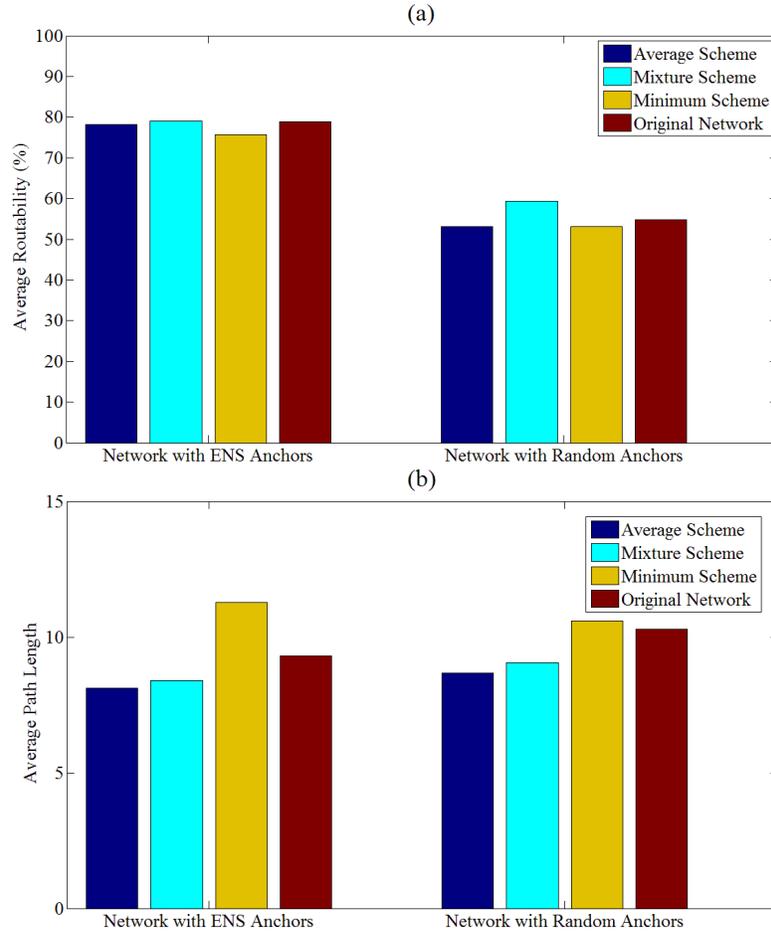


Figure 4.8: a) Average routability of new inserted mobile sensors in 3D USN and b) average path length of new inserted mobile sensors in 3D USN

Table 4.10: Average routability of original 3D underwater network with different percentage of inserted new nodes

% New nodes	ENS anchor placement				Random anchor placement			
	Network	Average	Mixture	Minimum	Network	Average	Mixture	Minimum
1%	78.82%	78.77%	77.10%	78.84%	54.80%	55.05%	53.96%	54.96%
2%	78.82%	78.81%	74.73%	78.96%	54.80%	55.17%	53.49%	55.07%
4%	78.82%	78.98%	73.13%	79.01%	54.80%	55.39%	52.39%	55.24%
6%	78.82%	79.76%	71.78%	79.60%	54.80%	55.73%	51.10%	55.53%
8%	78.82%	79.89%	70.15%	79.76%	54.80%	56.04%	50.15%	55.84%
10%	78.82%	80.30%	67.52%	79.98%	54.80%	56.63%	49.48%	56.42%

Table 4.11: Average path length of original 3D underwater network with different percentage of inserted new nodes

% New nodes	ENS anchor placement				Random anchor placement			
	Network	Average	Mixture	Minimum	Network	Average	Mixture	Minimum
1%	9.31	9.27	9.14	9.28	10.29	10.27	10.21	10.33
2%	9.31	9.16	8.90	9.29	10.29	10.24	10.14	10.33
4%	9.31	9.07	8.69	9.28	10.29	10.16	10.01	10.45
6%	9.31	8.99	8.46	9.21	10.29	10.09	9.85	10.40
8%	9.31	8.89	8.20	9.16	10.29	9.92	9.62	10.42
10%	9.31	8.78	7.89	9.11	10.29	9.81	9.50	10.61

4.3 Summary

So far, three VC generation schemes are proposed, discussed and simulated for routing performance of new mobile sensor nodes' receiving and passing packets in both 2D and 3D networks. Average Scheme is a good choice when new nodes receive and pass packets in the network and they are placed in regular positions. Average Scheme fails when new nodes are placed in "Ring" positions and new nodes' receiving packets ability severely decreases. In this case, Mixture Scheme can replace Average Scheme in order to get closer to one of the neighbors for mobile sensor to improve receiving packets performance. Average and Mixture Schemes are effective in both 2D and 3D VCS based WSNs for newly deployed mobile sensors.

CHAPTER 5

TOPOLOGY PRESERVING MAPS MODIFICATION SCHEMES

Although TPMs technique provides good substitute map for mapping [17], routing [20] and boundary detection [18], the compression resulted from ignoring first significant component from SVD introduces topology inaccuracy in TPMs, especially for the sensor nodes at the outer boundary of TPMs which are far from the center in radial distance, due to the lack of radial distance information. This compression thus leads to significant errors and nonlinear distortion compared with a physical map. The accuracy of tracking performance at the edges of network will suffer from this inaccuracy. For example, if TPMs are used as mobility guide map providing location information for calculating current velocity and direction, the distorted edges with inaccurate location information will cause wrong results for calculating velocity and direction. Thus in this chapter, modification schemes for TPMs are proposed in order to resolve the edge compression problem. Basic TPMs generation algorithm is reviewed in Section 5.1. Two TPMs modification schemes are discussed and related simulation results are shown in Section 5.2 and Section 5.3 correspondingly. Simple summary is given in Section 5.4.

5.1 Topology preserving maps generation

A VCS is based on a set of anchors, which correspond to a subset of nodes, selected randomly or by an anchor selection strategy [17][19]. Each node in the network, including anchors, is characterized by a VC vector, consisting of shortest hop distances to each of the anchors [13][17][19][37]. Directional and geographic information are not available in VCS as the VCs propagate radially. A method to extract directional information from VCs in the form of TCs is presented in [17]. TPM of a network based on TCs is a somewhat distorted version of the physical map of the network. It however has been shown to preserve relative position information, and is a good substitute for physical information in applications for WSNs such as routing [20] and boundary detection [18]. Birds' eye view of an area is an example of a distorted physical map, which can still be used for such functions, even though it is not an exact physical map. In

this paper, we consider dealing with tracking and mobility related applications directly in the TC domain instead of the physical domain, which is somewhat analogous to using a bird's eye view map. In this chapter, the distortion problem of TPMs and modification schemes are discussed, in order to provide more accurate topology map as preparation for mobility tracking in Chapter 6.

Consider a network with M anchors and N sensor nodes. Thus each node is characterized by a VC vector of length M , the i^{th} element of which corresponds to the minimum number of hops from the node to the i^{th} anchor. Let P be the $N \times M$ matrix containing VCs of all sensor nodes in the network. Efficient and sensor network-friendly implementation of the SVD computation is addressed in [17]. Let P be a $N \times N$ matrix containing VCs of all sensor nodes in the network. Generating TCs from P set using SVD is presented below [17]:

$$P = U \cdot S \cdot V^T \quad (5.1)$$

$$P_{SVD} = P \times V \quad (5.2)$$

$$[X_T, Y_T] = [P_{SVD}^{(2)}, P_{SVD}^{(3)}] \quad (5.3)$$

P_{SVD} is a $N \times M$ matrix containing sensor nodes' principal components. The first column $P_{SVD}^{(1)}$ is the most significant component but contains 1-dimensional radial information which is not sufficient for 2-dimensional TMPs [2]. Meanwhile second column $P_{SVD}^{(2)}$ and third column $P_{SVD}^{(3)}$ contain the topological information that can be translated to angular information and these two columns can be seen as 2D Cartesian coordinate set for sensor nodes in topological domain [2]. X_T and Y_T , are both $N \times 1$ column vectors and $[X_T, Y_T]$ in eqn. (5.3) is the TC set for the network. i.e., its i^{th} row corresponds to the TCs of the i^{th} node. Considering the extensive computation cost of SVD in deriving TCs from VC set of all sensor nodes which is matrix P , calculating SVD components from anchor set is economical choice for large-scale WSNs. Let A be a $M \times M$ matrix containing VCs of anchor sensor nodes. Generating TCs from anchor nodes set using SVD is presented in eqn. (5.4) to eqn.(5.6) [17]:

$$A = U_A \cdot S_A \cdot V_A^T \quad (5.4)$$

$$P_{SVD} = P \times V_A \quad (5.5)$$

$$[X_T, Y_T] = [P_{SVD}^{(2)}, P_{SVD}^{(3)}] \quad (5.6)$$

TPMs generation methods from SVD based on entire network node set P and anchor node set A are both used for TPMs modification schemes simulations.

5.2 Modification scheme1 and simulation results

Although $P_{SVD}^{(2)}$ and $P_{SVD}^{(3)}$ from eqn. (5.3) and (5.6) which are derived from either entire network node set or anchor node set reconstruct the lost directional map, ignoring the most significant component $P_{SVD}^{(1)}$ leads to compression, especially for the sensor nodes at the outer boundary of TPM which are far from the center in radial distance, due to the lack of radial distance information. This compression thus introduces significant errors and non-linear distortion compared with a physical map. The accuracy of TCs at the edges of network will suffer from this inaccuracy.

Therefore we use the following modification to generate a 2D TPM that is less distorted. We still keep the 2D angular information obtained from $P_{SVD}^{(2)}$ and $P_{SVD}^{(3)}$ and at the same time take radial information in $P_{SVD}^{(1)}$ into consideration. Consider a node in sensor network with TCs (x_T, y_T) . Note that (x_T, y_T) is the row corresponding to the node in $[X_T, Y_T]$ given by eqn. (5.3) and (5.6). The corresponding SVD components for this node are $p_{SVD}^{(1)}, p_{SVD}^{(2)}, p_{SVD}^{(3)}$, etc., which are extracted from this node's corresponding vector in P_{SVD} . We keep the directional information as the angle of sensor node's TCs to the origin in TPM:

$$\theta_T = \tan^{-1}(y_T/x_T) \quad (5.7)$$

The distance r_T between sensor node and the origin in TPM is:

$$r_T = \sqrt{x_T^2 + y_T^2} = \sqrt{(p_{SVD}^{(2)})^2 + (p_{SVD}^{(3)})^2} \quad (5.8)$$

We modify r_T by weighting r_T by the radial information in $p_{SVD}^{(1)}$ as followed:

$$r'_T = \sqrt{(p_{SVD}^{(1)})^2 + (p_{SVD}^{(2)})^2 + (p_{SVD}^{(3)})^2} \quad (5.9)$$

Modified TCs of this node (x'_T, y'_T) can be rewritten as:

$$x'_T = r'_T \times \cos \theta_T \quad (5.10)$$

$$y'_T = r'_T \times \sin \theta_T \quad (5.11)$$

We denote by $[X'_T, Y'_T]$ the matrix of modified TCs obtained by applying eqn. (5.7), (5.9) – (5.11) to $[X_T, Y_T]$ in eqn. (3) and eqn. (6). In this modification, for each node the directional information in $p_{SVD}^{(2)}$ and $p_{SVD}^{(3)}$ is kept and the radial distance in $p_{SVD}^{(1)}$ is also considered in modified r'_T so that the compression at map edges can be reduced. $[X'_T, Y'_T]$ are referred to as TCs in the remainder of this chapter.

Three networks from [17] are used to evaluate modification schemes and they are circular network with three voids in Figure 5.1(a) with 496 nodes, big circle network in Figure 5.1(b) with 596 nodes and odd network in Figure 5.1(c) with 550 nodes. At the same time, Test Network (TN6) is also used for evaluation, whose geographic map can be seen in Figure 5.2. TN6 is in 120unit×90unit irregular field and consists of 5137 sensor nodes, each with communication range of 1 unit. In these four networks, nodes with red triangle in network are the anchors. Anchors for networks in Figure 5.1(b) and Figure 5.2 are chosen by ENS algorithm. 15 anchors for networks in Figure 5.1(a) and Figure 5.1(c) are chosen randomly.

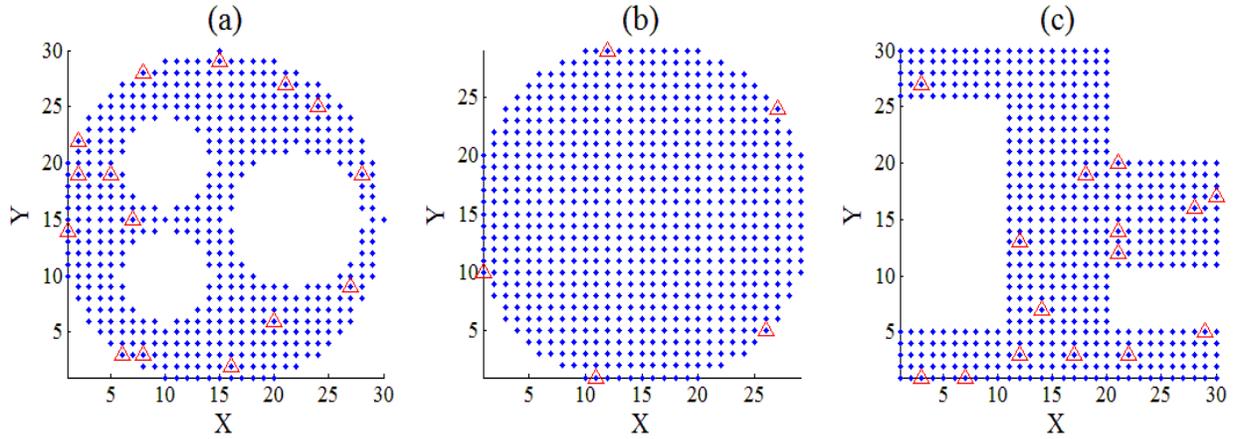


Figure 5.1: Geographic maps of a) circular network with three voids with 15 random anchors; b) big circle network with 5 ENS anchors and c) odd network with 15 random anchors

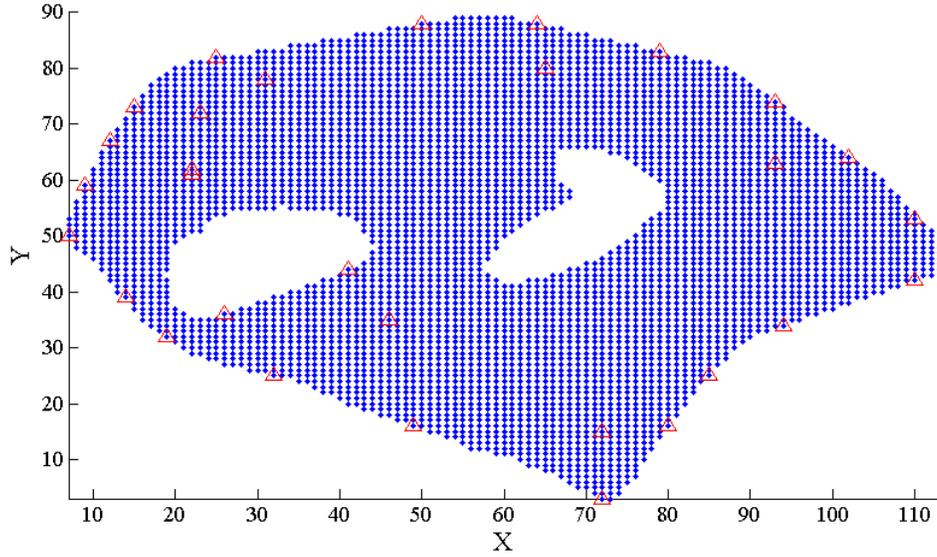


Figure 5.2: Geographic map of TN6 with 30 ENS anchors

Unmodified and modified TPMs for four networks are shown in Figure 5.3, Figure 5.4 and Figure 5.5. Figure 5.3(a), Figure 5.4(a) and Figure 5.5(a) show original TPMs generated using eqn. (5.1) – (5.3) and Figure 5.3(b), Figure 5.4(b) and Figure 5.5(b) show modified TPMs for networks in Figure 5.3(a), Figure 5.4(a) and Figure 5.5(a) based on eqn. (5.7), (5.8) – (5.11). Figure 5.3(a), Figure 5.4(c) and Figure 5.5(c) show original TPMs generated using eqn. (5.4) – (5.6) and Figure 5.3(d), Figure 5.4(d) and Figure 5.5(d) show modified TPMs for networks in Figure 5.3(a), Figure 5.4(c) and Figure 5.5(c) based on eqn. (5.7), (5.8) – (5.11). From modified maps shown in Figure 5.3(b), Figure 5.4(b), Figure 5.5(b), Figure 5.3(d), Figure 5.4(d) and Figure 5.5(d), edges are decompressed significantly. Modification scheme1 works well with different SVD component generation methods. Considering large computation cost of SVD in deriving TCs from VC set of all sensor nodes for TN6 in Figure5.2 calculating SVD components from anchor set using eqn. (4) – (6) is used. In Figure 5.6(b), edges are decompressed significantly compared to those of Figure 5.6(a).

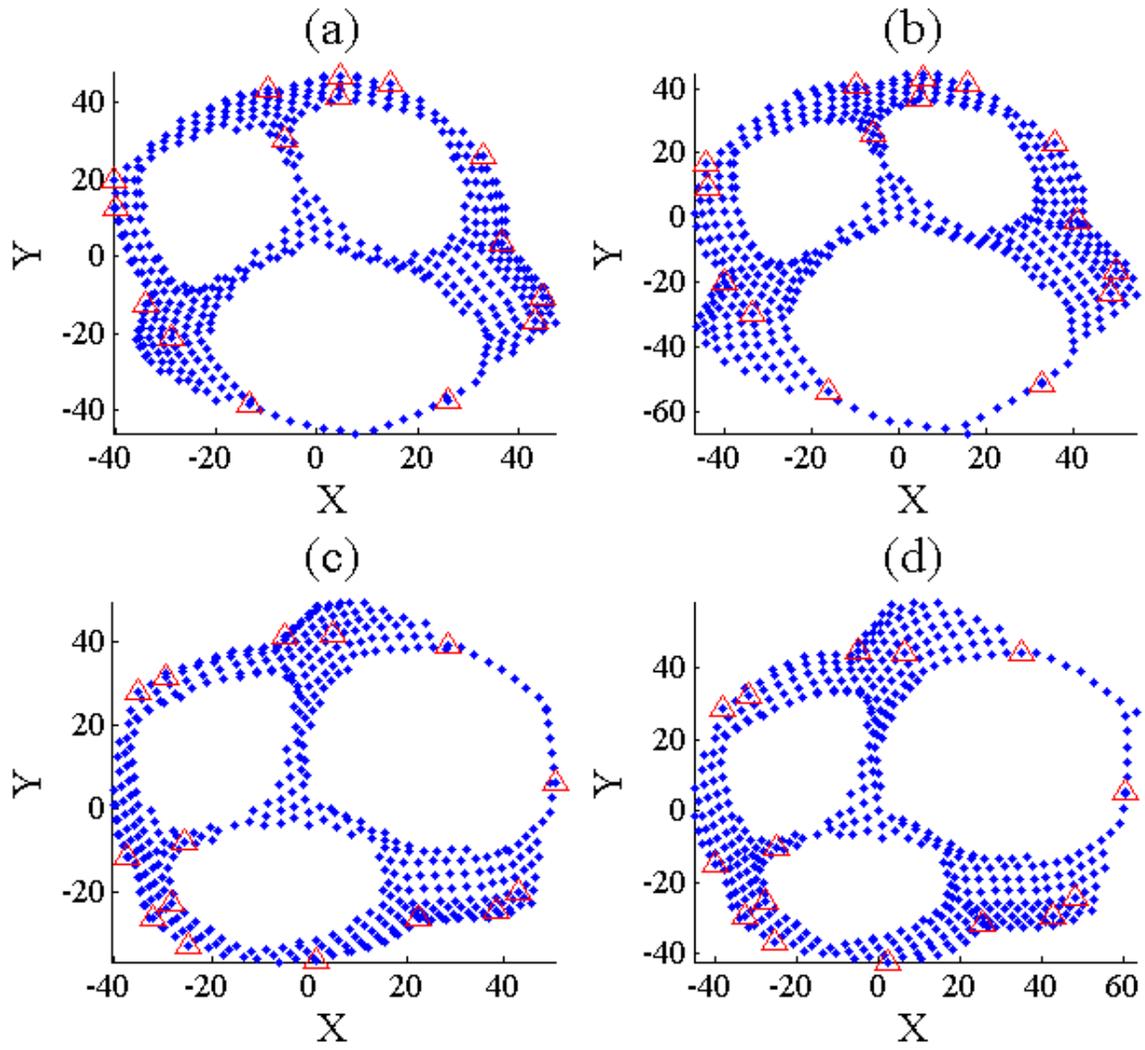


Figure 5.3: Circular network with three voids' a) TPM generated from SVD based on entire network node set; b) modified TPM of network in Figure 5.3(a); c) TPM generated from SVD based on anchor node set and d) modified TPM of network in Figure 5.3(c)

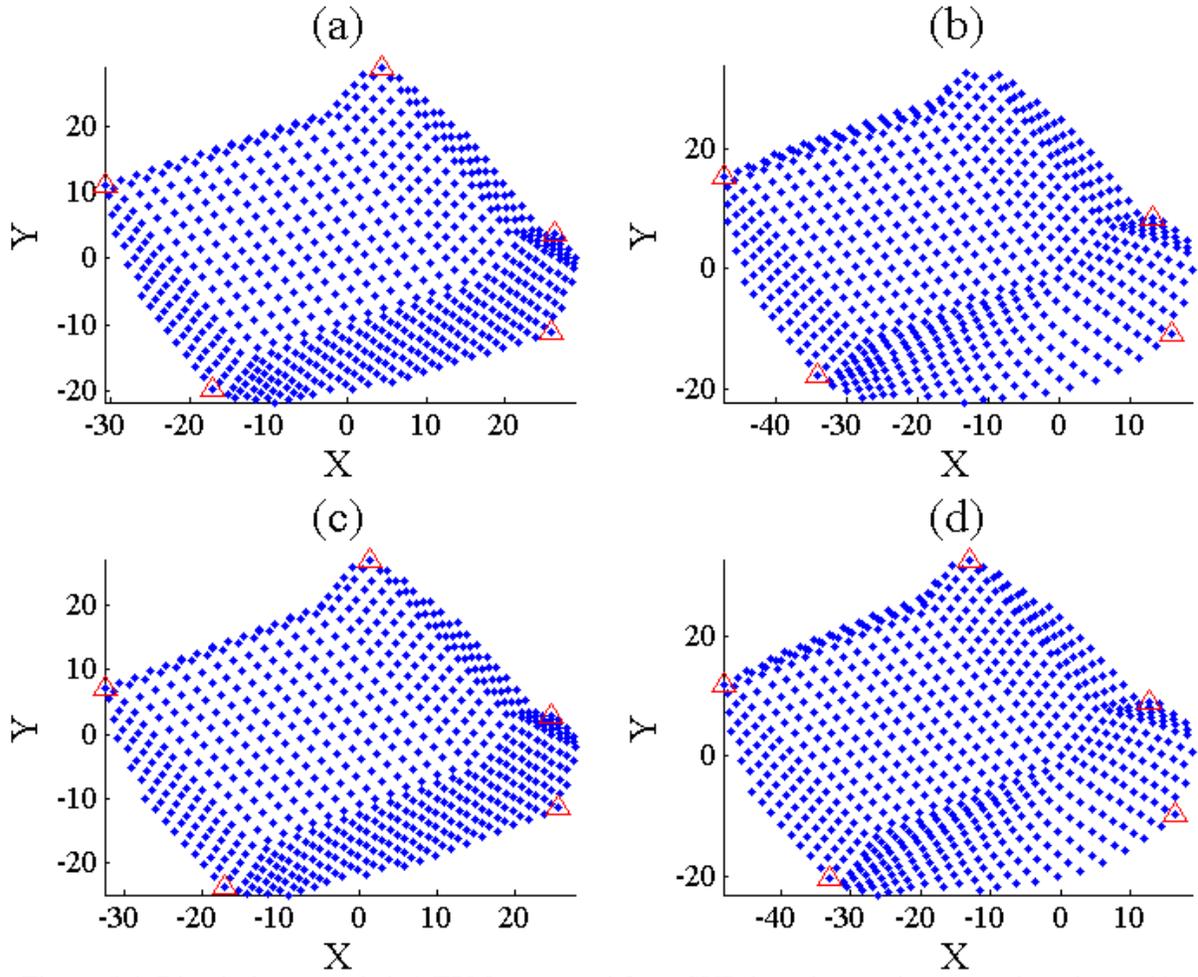


Figure 5.4: Big circle networks' a) TPM generated from SVD based on entire network node set; b) modified TPM of network in Figure 5.4(a); c) TPM generated from SVD based on anchor node set and d) modified TPM of network in Figure 5.4(c)

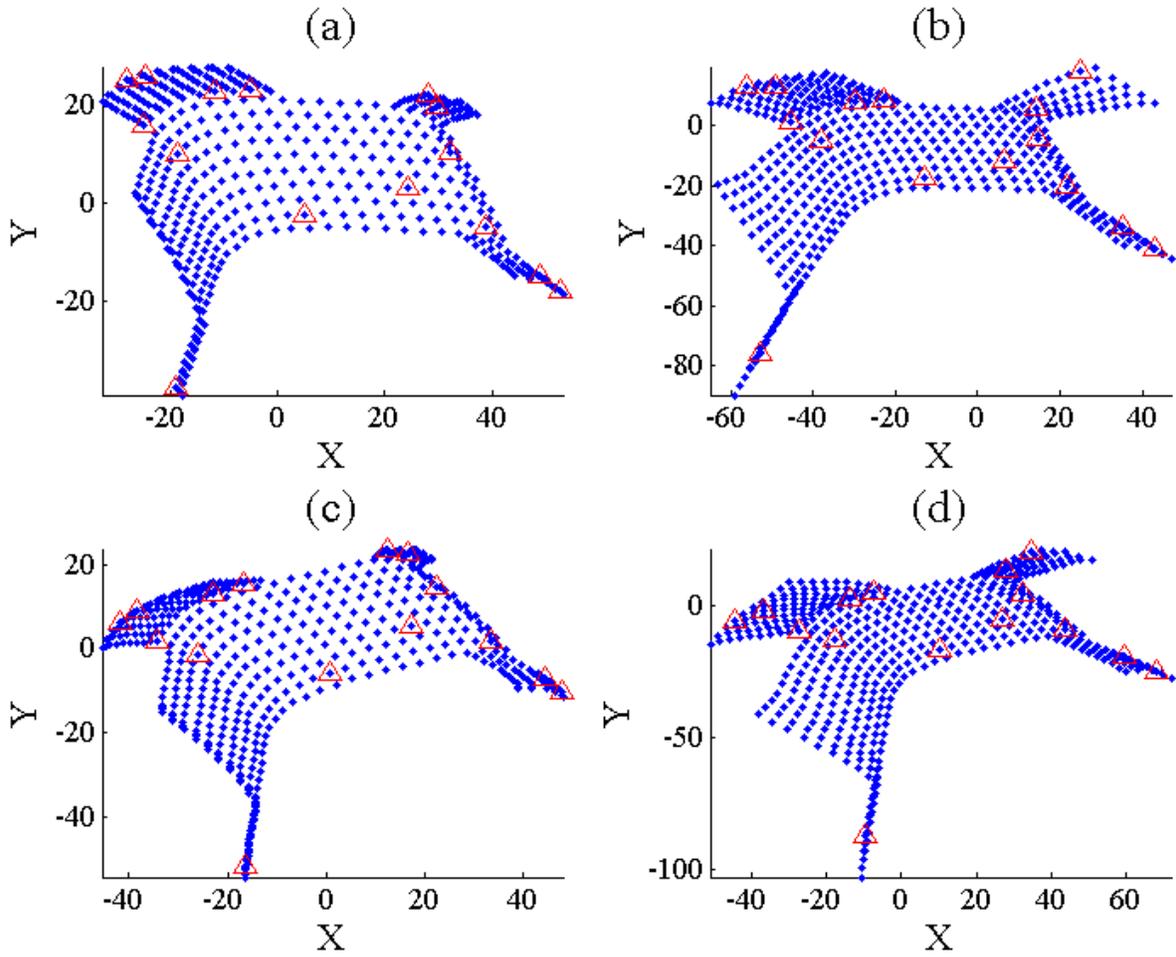


Figure 5.5: Odd network's a) TPM generated from SVD based on entire network node set; b) modified TPM of network in Figure 5.5(a); c) TPM generated from SVD based on anchor node set and d) modified TPM of network in Figure 5.5(c)

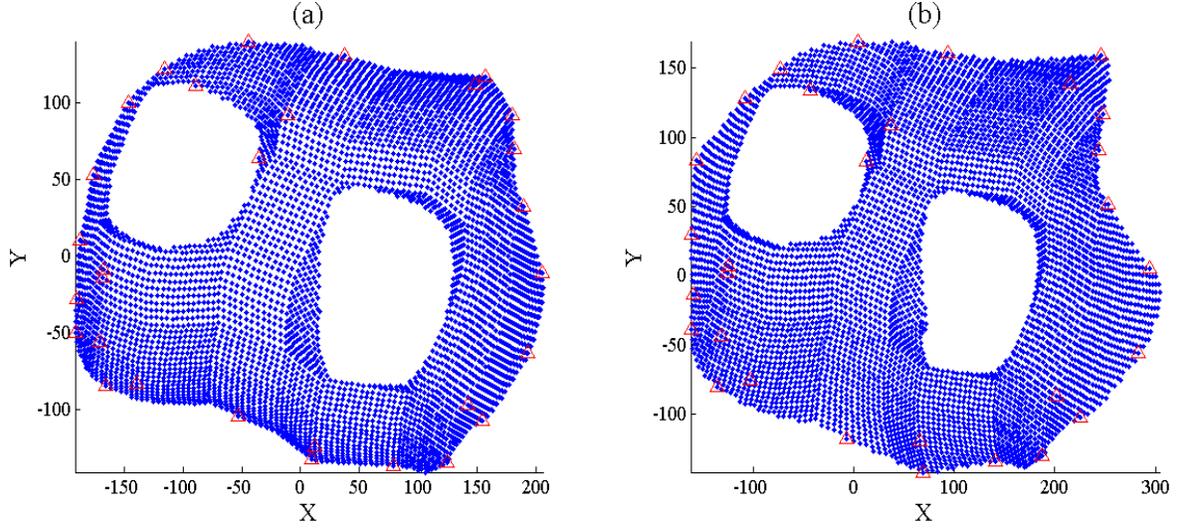


Figure 5.6: TN6's a) original TPM generated from SVD based on anchor node set and b) modified TPM generated from SVD based on anchor node set

5.3 Modification scheme2 and simulation results

Similar to scheme1, in scheme2, directional information as the angle of sensor node's TCs to the origin in TPM is still kept in eqn. (5.12)

$$\theta_T = \tan^{-1}(y_T/x_T) \quad (5.12)$$

The distance r_T between sensor node and the origin in TPM is modified and simplified as the first SVD component since it is the most significant component. We modify r_T by weighting r_T by the radial information in $p_{SVD}^{(1)}$ as followed:

$$r'_T = p_{SVD}^{(1)} \quad (5.13)$$

Modified TCs of this node (x'_T, y'_T) can be rewritten as:

$$x'_T = r'_T \times \cos \theta_T \quad (5.14)$$

$$y'_T = r'_T \times \sin \theta_T \quad (5.15)$$

So $[X'_T, Y'_T]$ is the matrix of modified TCs obtained by applying eqn. (5.12), (5.13) – (5.15) to $[X_T, Y_T]$ in eqn. (5.3) and eqn. (5.6). In this modification, for each node the directional information in $p_{SVD}^{(2)}$ and $p_{SVD}^{(3)}$ is kept and the radial distance in $p_{SVD}^{(1)}$ is modified r'_T in eqn. (5.13) so that the compression at map edges can be reduced. The calculation is less for large-scale WSNs however accuracy

in the center of TPM decreases. Modification scheme1 and scheme2 can be selected according to different requirements of application.

In evaluations, two SVD generation methods one based on anchor node set is used under scheme2. Modified TPMs for four networks using scheme2 are shown Figure 5.7(c), Figure 5.8(c) and Figure 5.9(c) and Figure 5.10(b). Compared with TPMs without any modification, the edges are well unfolded but center folding is introduced due to the ignorance of other significant components in $p_{SVD}^{(2)}$ and $p_{SVD}^{(3)}$. Modification scheme2 brings obvious unfolding effects at the edges however decreases the accuracy of TCs in the center of TPMs. If the application only requires more accurate topology information at the edges, modification scheme2 can be considered since modification computation is reduced compared with modification scheme1.

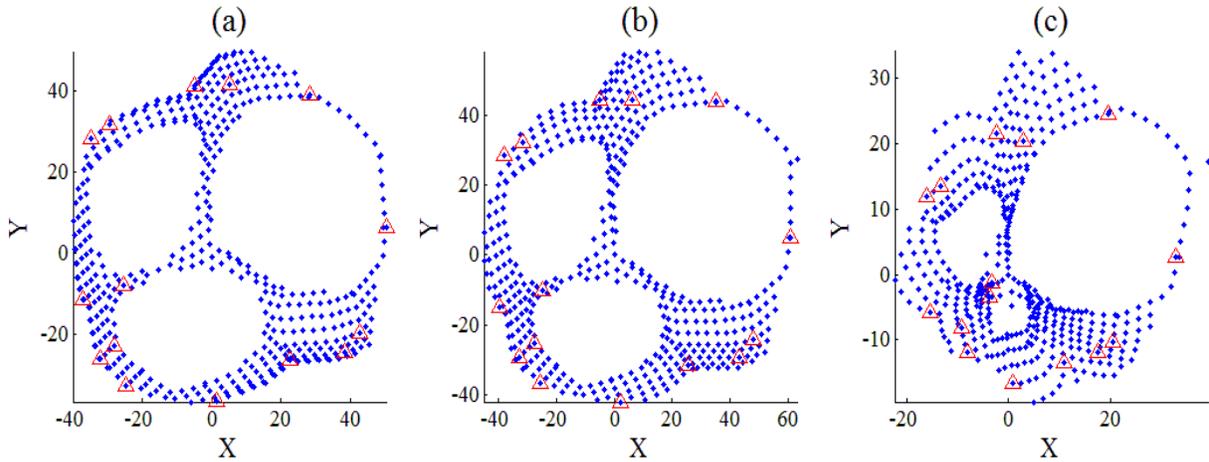


Figure 5.7: Circular network with three voids' a) original TPM generated from SVD based on anchor node set; b) modified TPM based on modification scheme1 and c) modified TPM based on modification scheme2

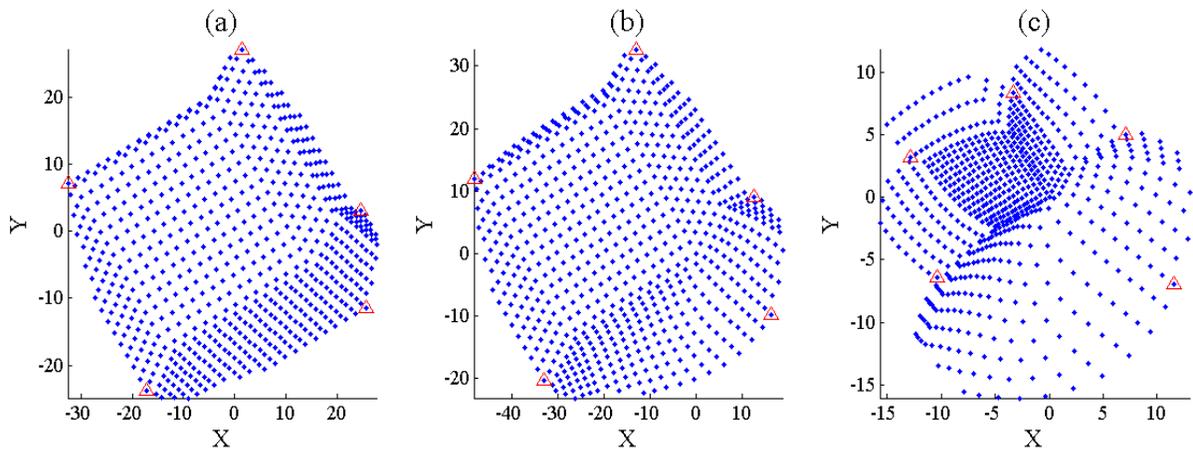


Figure 5.8: Big circle network's a) original TPM generated from SVD based on anchor node set; b) modified TPM based on scheme1 and c) modified TPM based on scheme2

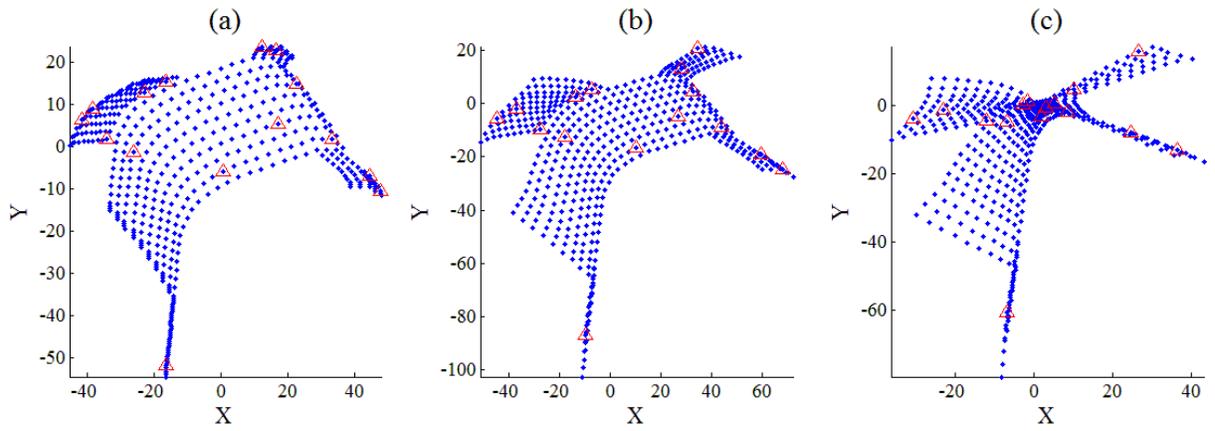


Figure 5.9: Odd network's a) original TPM generated from SVD based on anchor node set; b) modified TPM based on scheme1 and c) modified TPM based on scheme2

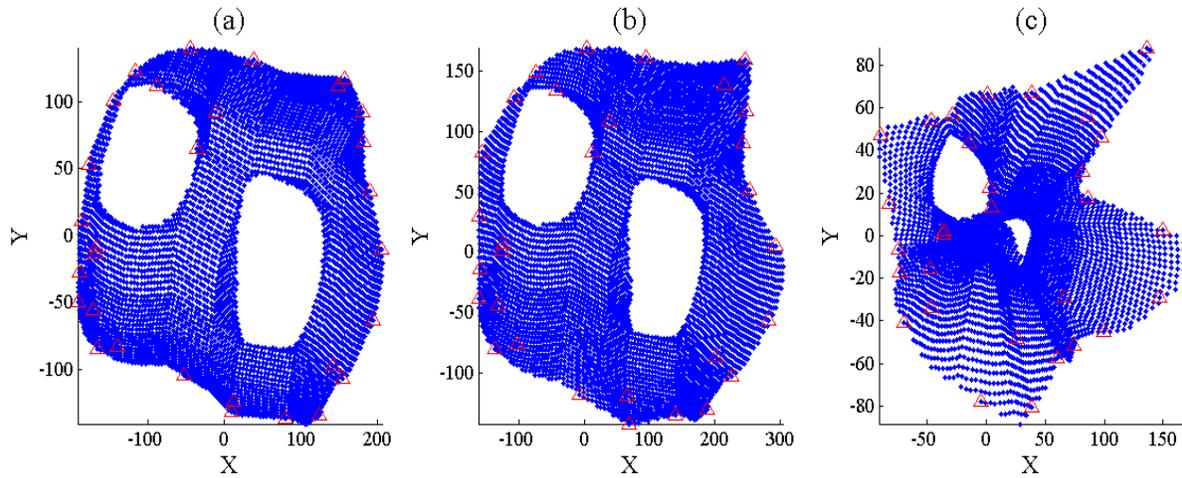


Figure5.10: TN6's a) original TPM generated from SVD based on anchor node set; b) modified TPM based on scheme1 and c) modified TPM based on scheme2

5.4 Summary

In this chapter, two modification schemes to improve the folding edge at the boundary of TPMs are designed. Two schemes both provide unfolded edges for TPMs and increase the accuracy of TPMs at the boundary. The modification of TPMs gives necessary basis for TPMs related techniques especially for TCTP algorithm which will be discussed in next chapter because TPMs act as a guide map of tracking instead of traditional geographic maps.

CHAPTER 6

TOPOLOGICAL COORDINATE BASED MOBILITY TRACKING AND PREDICTION

In this chapter, 2D-TCTP algorithm is proposed, which is TC based approach for performing sensor tracking as well as prediction. To our knowledge, this is the first tracking algorithm that operates in such a virtual domain thus not requiring geographic information based on physical distance measurements. Modification scheme1 from Chapter 5 is used to reduce edge distortions in 2D TPMs to enhance its accuracy. The VCs of the mobile node are derived without resorting to additional flooding's by anchors. The detecting performance of 2D-TCTP algorithm in mobile sensor tracking and prediction is compared with the same approach based on geographic or physical information. Additionally, we also extend 2D-TCTP algorithm to 3D WSNs. In this chapter, Section 6.1 introduces the TC generation scheme for mobile sensor. Detailed description of 2D-TCTP algorithm is given in Section 6.2. Section 6.2.1 shows simulation results of 2D-TCTP using three mobility models. 3D-TCTP is presented and simulated in Section 6.3. Last Section 6.4 is a brief summary.

6.1 Topological coordinate generation scheme for mobile sensor

Recalling SVD from anchor set A in deriving TCs from VCs is used in 2D-TCTP in eqn. (6.1) and eqn. (6.2). X_T and Y_T , are both $N \times 1$ column vectors and $[X_T, Y_T]$ in eqn. (6.3) is the TC set for the network [17].

$$A = U_A \cdot S_A \cdot V_A^T \quad (6.1)$$

$$P_{SVD} = P \times V_A \quad (6.2)$$

$$[X_T, Y_T] = [P_{SVD}^{(2)}, P_{SVD}^{(3)}] \quad (6.3)$$

TPMs modification scheme1 proposed in Chapter 5 is used in TCTP, which can be seen in eqn. (6.4), (6.6)–(6.8).

$$\theta_T = \tan^{-1}(y_T/x_T) \quad (6.4)$$

r_T is modified as:

$$r'_T = \sqrt{(p_{SVD}^{(1)})^2 + (p_{SVD}^{(2)})^2 + (p_{SVD}^{(3)})^2} \quad (6.5)$$

Modified TCs of this node (x'_T, y'_T) can be rewritten as:

$$x'_T = r'_T \times \cos \theta_T \quad (6.6)$$

$$y'_T = r'_T \times \sin \theta_T \quad (6.7)$$

To do the tracking in TC domain, TCs of mobile sensor need to be generated. To generate TCs of the mobile node at a point using existing approach, VCs of it is generated using Average Scheme from Chapter 4. Firstly, VCs of mobile sensor is obtained by taking average of neighboring sensors' VCs. The averaged VCs provide mobile sensor's location in VC domain. Secondly by applying SVD in eqn. (6.3) and modification in eqn. (6.4)-(6.7), TCs of mobile sensor are generated which can be seen as an estimated location in TPM. Note that the accuracy of TCs of the mobile node in the network highly depends on the accuracy of this approximation.

Consider mobile sensor that is surrounded by n neighbor sensors. Let MT , a $1 \times M$ vector, be the mobile sensor's VCs and P_{N_i} , a $1 \times M$ vector, be the i^{th} neighbor sensor's VCs. So MT can be obtained by eqn. (6.8):

$$MT = \sum_{j=1}^n P_{N_i} / n \quad (6.8)$$

$$MT_{SVD} = MT \times V_A \quad (6.9)$$

In eqn. (6.9), MT_{SVD} is a $1 \times M$ vector of mobile sensor's SVD components. Second and third element of MT_{SVD} are selected as original TCs (x_{MT}, y_{MT}) for mobile sensor. Modified TC (x'_{MT}, y'_{MT}) can be obtained using eqn. (6.4)-(6.7).

6.2 2D Topological coordinate based tracking and prediction algorithm

This section proposes the 2D-TCPC algorithm for mobility tracking and prediction in TC domain.

Following terms are used in TCTP algorithm:

1. Sampling time: the time difference between two consecutive sensing locations for mobile sensor.

2. Detection ellipse: The area surrounding mobile sensor's predicted position, in which the mobile sensor is predicted to appear at a given future time. The area is in the shape of an ellipse. Detection ellipse compensates for errors in prediction of the position. The major axis and minor axis of detection ellipse can be adjusted so that it can cover area of different sizes to meet the requirements of different applications.
3. Detecting sensors: Static sensors inside detection ellipse. The number of detecting sensors varies with the size of ellipse.
4. (Prediction) Time Window: Mobile sensor's track information is sampled at time t_1 and its position is to be predicted at future time $t_f = t_1 + t_p$ where t_p is prediction time. Mobile sensor is expected to appear at predicted location at time t_f . We set up a time window Δt and we expect mobile sensor's arrival to be within time $(t_f - \Delta t, t_f + \Delta t)$.
5. Detection failure rate: Detection failure rate is the probability that the mobile sensor is not detected by any detecting sensor in the time window Δt . Detection failure rate is the main evaluation metric for proposed algorithm.

In TCTP algorithm and related simulation, we make the following assumption:

1. There is one mobile sensor node (sensor) travelling in the network. BS tries to track and monitor this mobile sensor in TC domain.
2. The time delay caused by communication among sensors is considered to be negligible compared to the time it takes the node to change its neighborhood.
3. BS possesses the network's TPM, receives mobile sensor's averaged location in VC domain, calculates its corresponding TCs, tracks its current position in TC domain, predicts the future location and then alerts the sensors in the detection ellipse so that they can wait for the arrival of mobile sensor.

Traditional tracking and prediction algorithms in geographic domain operate by recording the motion track. Information such as current motion velocity and direction are then used to linearly predict

mobile sensor's position at a future time [28][53][54]. 2D-TCTP algorithm follows this basic idea for prediction, but replaces GCs by TCs for current velocity and direction calculation as well as future position prediction. Mobile sensor's TC position at current time t_i is (x_{MT_i}, y_{MT_i}) and at previous time t_{i-1} is $(x_{MT_{i-1}}, y_{MT_{i-1}})$. The TC domain velocity V_T and direction angle α_T can be calculated using eqn. (6.10) and (6.11):

$$V_T = \frac{\sqrt{(x_{MT_i} - x_{MT_{i-1}})^2 + (y_{MT_i} - y_{MT_{i-1}})^2}}{t_i - t_{i-1}} \quad (6.10)$$

$$\alpha_T = \cos^{-1} \frac{x_{MT_i} - x_{MT_{i-1}}}{\sqrt{(x_{MT_i} - x_{MT_{i-1}})^2 + (y_{MT_i} - y_{MT_{i-1}})^2}} \quad (6.11)$$

$$x_{MT_{i+1}} = x_{MT_i} + V_T t_p \cos(\alpha_T) \quad (6.12)$$

$$y_{MT_{i+1}} = y_{MT_i} + V_T t_p \sin(\alpha_T) \quad (6.13)$$

$(x_{MT_{i+1}}, y_{MT_{i+1}})$ are calculated using eqn. (6.12) and (6.13) as the estimated future position at time t_{i+1} for mobile sensor in TC domain after prediction time t_p from current time t_i .

There are three phases in TCTP algorithm as follows:

1. Estimation of TCs of mobile sensor (sampling)

Every sampling time, mobile sensor communicates with neighbor sensors within 1-hop range. VCs of mobile sensor's current position are calculated using eqn. (6.9) and sent to BS, e.g., by mobile sensor. BS uses SVD method to calculate corresponding modified TCs in eqn. (6.10) and eqn. (6.4)-(6.7).

2. BS predicts and sets up detecting sensors

BS receives updated VCs of mobile sensor every sampling time and records its motion history in TC domain. To predict mobile sensor's future position, BS calculates current TC velocity and direction using eqn. (6.11) and (6.12). Then based on current TC velocity and direction, BS calculates mobile sensor's position in future using eqn. (6.13) and (6.14). After locating the future position in TPM, BS sets the detection area as an ellipse. The predicted position is the center of ellipse and major axis of ellipse is set to be perpendicular to the estimated direction of motion in TC domain.

Figure 6.1 shows an example of ellipse detection area. The solid line with arrow is the linear prediction path for mobile sensor in TC domain, and the dashed line with arrow is its actual motion path. The setup of detecting sensors in ellipse is designed to guarantee the detection of mobile sensor although its track is nonlinear in TC domain. In the case where the predicted position is out of the boundary of TPM, detection ellipse will be adjusted so that the boundary nodes surrounding the projected path are considered to be the candidate sensors downstream for tracking. Figure 6.2 shows detecting sensors on the boundary that are alerted when the prediction position is out of network's boundary.

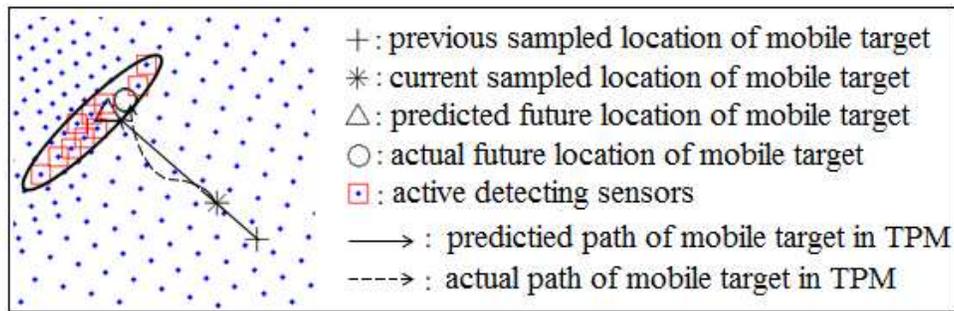


Figure 6.1: Detecting sensors setup when prediction position is within network's boundary

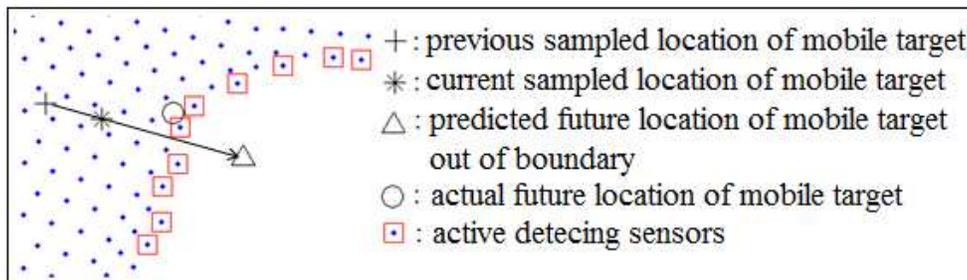


Figure 6.2: Detecting sensor setup when prediction position is out of network's boundary

3. Detecting sensors wait for mobile sensor's arrival

Detection ellipse is determined at the BS. Any message that is to be delivered to the mobile node around future time t_f is sent to the detecting sensors inside detection ellipse. Alternatively, in other scenarios of operation, detecting sensors may be woken up by BS to wait for mobile sensor's arrival.

6.2.1 Simulation results

In this section we evaluate the performance of TCTP algorithm. A simulator was developed using MATLAB® 2012a. First we use test network TN6 with generated TPM in Figure 6.3(a) and 6.3(b). We evaluate the algorithm using two mobility models to generate the movement in the physical domain, namely, the random direction and random waypoint models [15]. In random direction model, mobile sensor travels in a random direction until reaching the boundary of the network. After pausing for a certain time, mobile sensor node continues traveling in a new random direction. In random waypoint model, mobile sensor randomly selects a physical position as destination to move to, and after reaching it, mobile sensor randomly selects another position as next destination [15]. The results presented are based on approximately 900 prediction test positions along motion track for each mobility model.

Mobile sensor's tracks in geographic domain with two models are shown in Figure 6.4(a) and 6.4(b). Mobility of the node occurs in the geographic domain, i.e., moving in a straight line corresponding to one in geographic domain. The velocity of mobile sensor in geographic domain is constant at 0.5 unit/s.

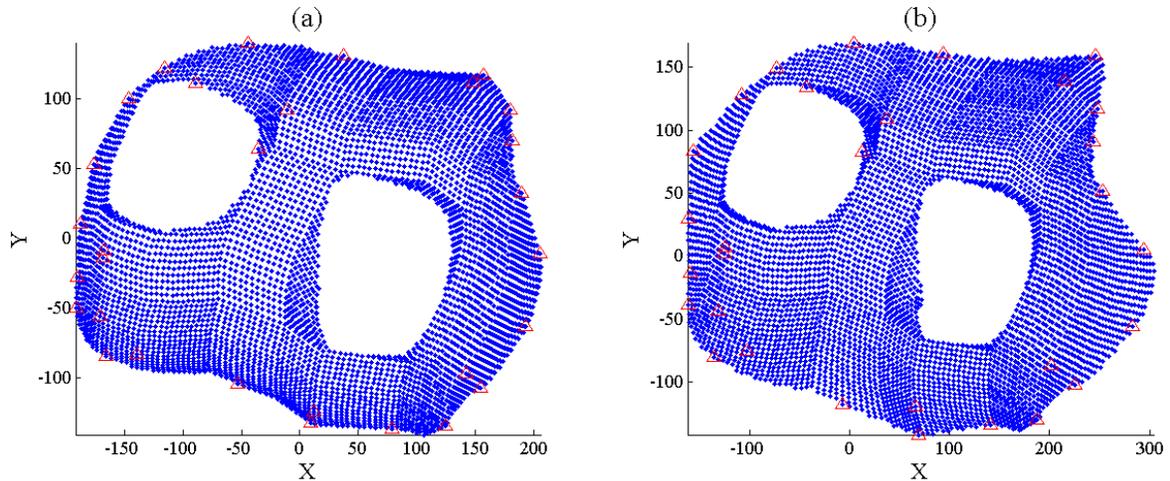


Figure 6.3: TN6's a) original TPM generated from SVD based anchor nodes and b) modified TPM generated from SVD based on anchor node set

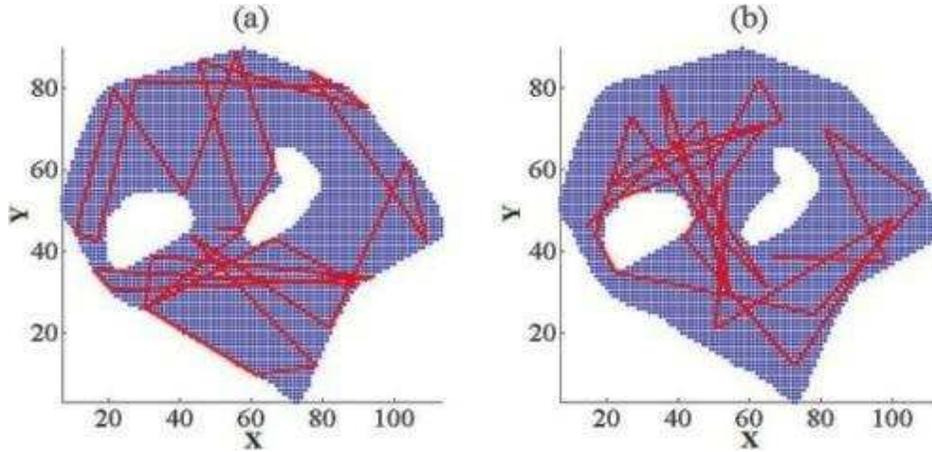


Figure 6.4: Mobile sensor's motion track in TN6 in a) random direction mobility model and b) random waypoint mobility model

To evaluate the effectiveness of TCTP algorithm that operates in the TC domain without any physical information, we also use the same tracking algorithm in geographic domain, in which case the tracking and prediction are both based on GCs. This is the existing approach, and as such it serves as the baseline for comparison. Note that the GC based approach has the added advantage of having a constant velocity due to the mobility models used. In our simulation, we sample the position of the mobile node every 1s, 2s, 4s, 6s, 8s and 10s. Every sampling time, the nodes position in the appropriate coordinate system is informed to BS. We set time window for prediction as 0s, 2s, 4s, 6s and 8s. We predict mobile sensor's position after prediction time 5s, 10s and 20s. For the number of detecting sensors, we choose 10 and 20 per each prediction test position, which corresponds on average to 0.19% and 0.39% of sensors in the network respectively.

Before comparing tracking in TC and GC domains, we point out that the geographic distance measurement provides continuous changes, whereas TCs of the mobile node change only at discrete instances as its neighborhood changes. This is due to the fact that the VCs of the sensor are obtained by averaging neighbors' VCs. As such tracking in TC domain is not very effective when sampling is done at very fine granularity. This is different from GC based tracking and prediction algorithm where shorter sampling time leads to optimal detection failure rate [54]. Figure 6.5(a) shows the variation of detection

failure rate vs. sampling time for TN6 in random direction mobility model in time window 4s when there are 20 detecting sensors in detection ellipse. When sampling time is short, i.e., when the travelled distance during the period is shorter than the communication range, this estimated average position is less accurate compared with actual position, due to the average location approximation. If the distance travelled in sampling interval is longer, the accuracy increases. However, if sampling time is too long, the tracking information cannot be updated in time so tracking performance deteriorates in both TC and GC based methods.

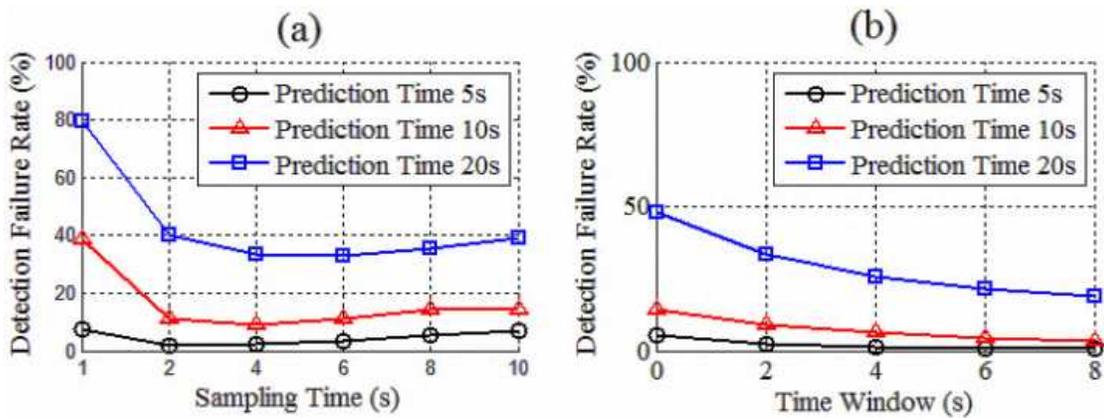


Figure 6.5: Detection failure rate for TN5 in random direction mobility model with 20 detection sensors vs. sampling time a) when time window is 4s and b) when sampling time is 4s

Figure 6.6 and Figure 6.7 compare the variation of detection failure rate for TN6 in different time windows for tracking in TC and GC domains when sampling time is 4s. For short-term prediction like 5s and 10s under motion velocity of 0.5 unit/s, we can see that the detection failure rate in TC and GC domains are quite close to each other when time window is 2s or longer. For long-term prediction, e.g., 20s into future, the longer the time window the lower detection failure rate will be obtained. It's a challenge to do long term prediction, both in TC and GC domains, due to the linear prediction model. This weakness can be improved by increasing the number of detecting sensor nodes or area of detection ellipse. However, the important conclusion is that the TC domain tracking is very competitive with that in the GC domain. Furthermore, the lower cost of TC domain tracking, e.g., in terms of power due to absence of GPS or

localization algorithms, also means that we may be able to increase the size of the detection ellipse in TC domain, thus enhancing its effectiveness.

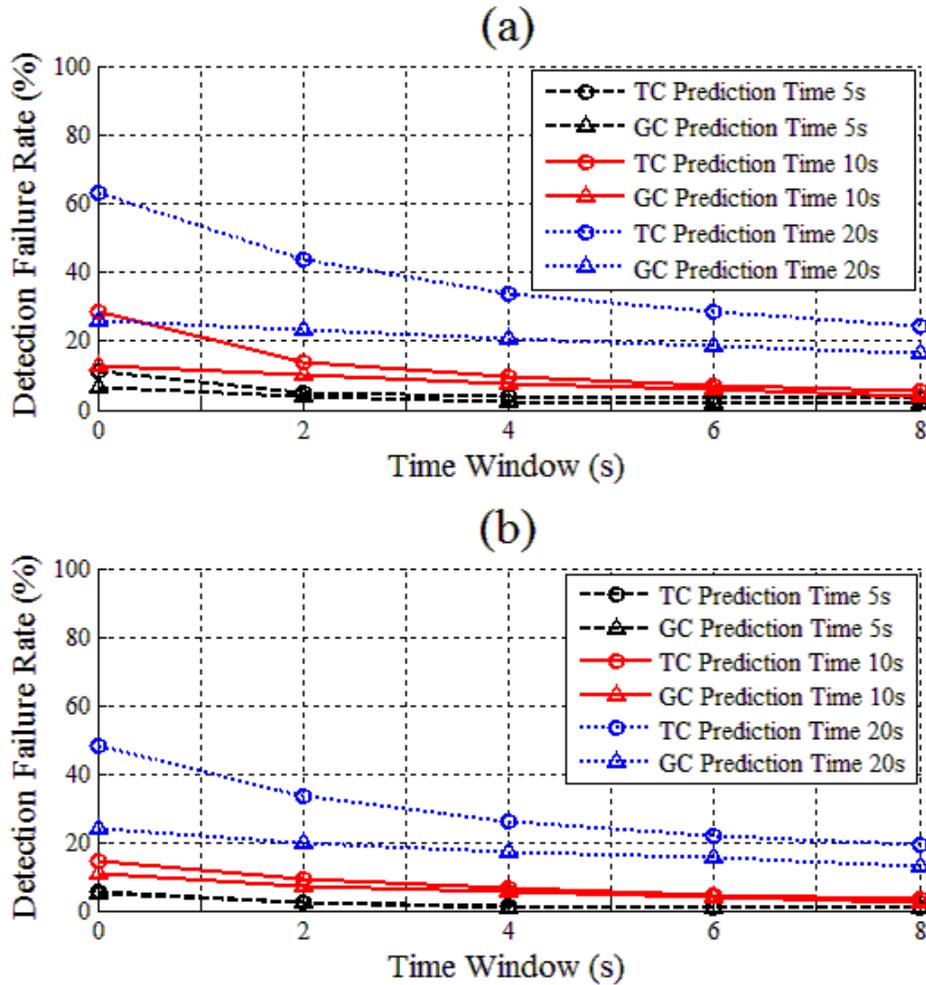


Figure 6.6: Detection failure rate comparison between TC domain and GC domain for TN6 in random direction mobility model, with a) 10 detecting sensors and b) 20 detecting sensors

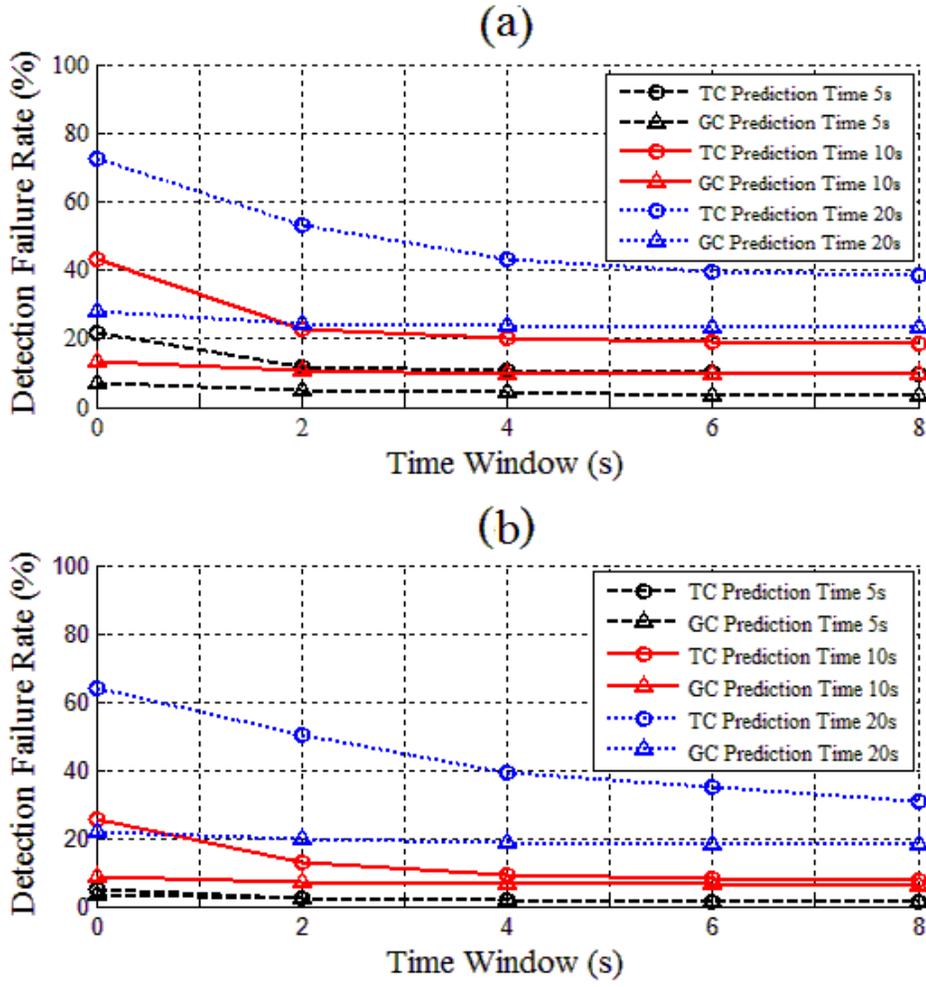


Figure 6.7: Detection failure rate comparison between TC domain and GC domain for TN6 in random waypoint mobility model a) with 10 detecting sensors and b) with 20 detecting sensors

We have also evaluated TCTP algorithm in Test Network 7 (TN7) with 2500 sensor nodes in random placement, a part of which is shown in Figure 6.8(a). Unlike TN6, the distances between adjacent nodes in TN7 are not constant. Mobile node follows random waypoint mobility model and the number of detecting sensors is 10. Mobile node's velocity is 0.5 units/s. Figure 6.9 shows the detection failure rate when sampling time is 10s. TC based approach comes within 20% of the GC based approach for a wide range of time windows, with lower rates at higher time windows (within 10% of GC based approach after time window 6s). TCTP shows competitive tracking and detection performance compared with GC based approach although physical distances among sensor nodes are random and unknown. It's significant that,

although without any geographic information, TCTP is able to achieve similar failure rate compared with geographic information based approaches.

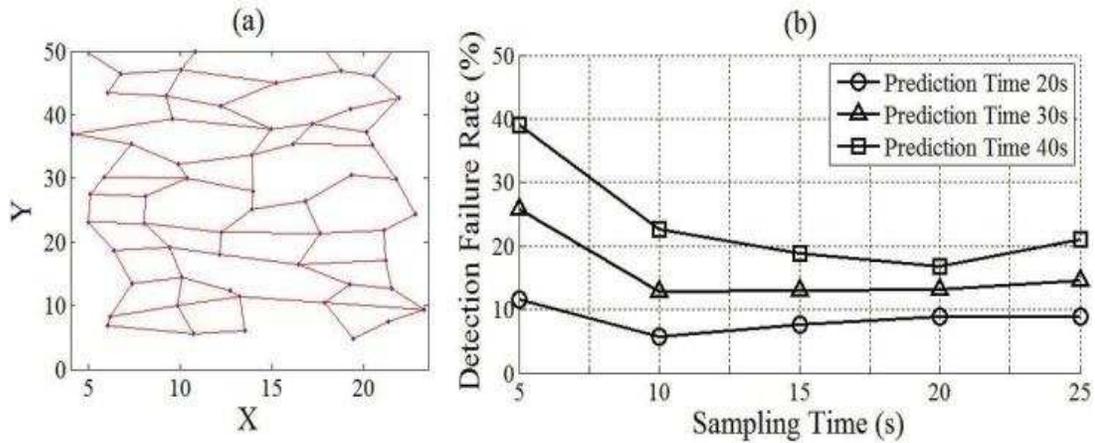


Figure 6.8: a) Part of physical topology of TN7 with random placed nodes and b) detection failure rate vs. sampling time for TN7 in time window 6s

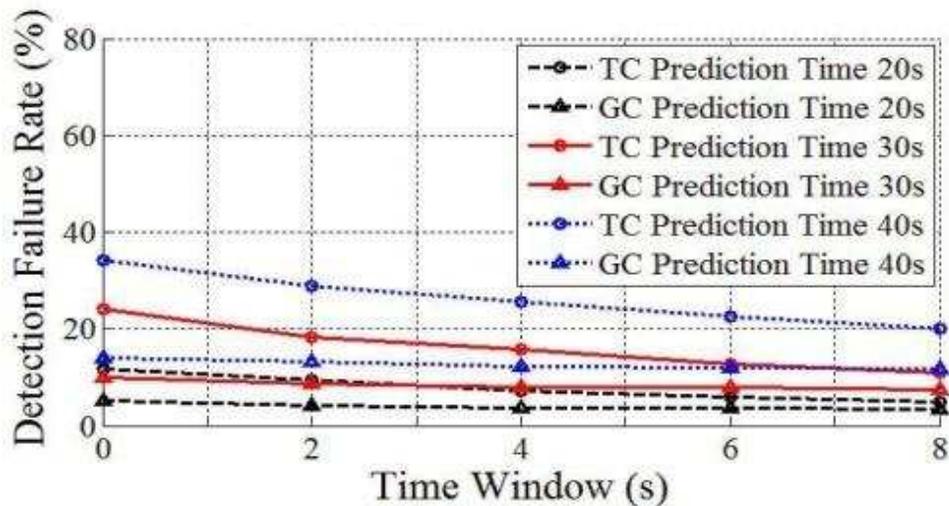


Figure 6.9: Detection failure rate comparison between TC domain and GC domain for TN7 when sampling time is 10s

There is large amount of research and study on mobility models because different mobility models have great effects on the performance of ad hoc network. Traditional and classical mobility models include random walk mobility model, random direction mobility model and random waypoint mobility model, which have been widely applied in research [12][24][29][33][46][58]. These mobility models are reported

to be unrealistic in real physical world due to the limitations such as sharp turn and speed decay [15][57]. A novel mobility model called Semi-Markov Smooth (SMS) mobility model [57] combines existing Gauss-Markov mobility model and actual speeding up/down phases to mimic movements in real physical world. In SMS model, three phases exist in movement: speed-up phase, middle smooth Gauss-Markov phase and slow-down phase [57]. It is assumed that mobile object accelerates before reaching a stable velocity and decelerates to full stop in straight line, with certain acceleration rate and no direction change. In the middle Gauss-Markov phase, suppose the mobile object reaches velocity v_a in direction ϕ_a after speed-up phase, the following velocity and direction in each time slot fluctuate with respect to v_a and ϕ_a , which can be seen in eqn. (6.14) and eqn. (6.15) [57]. In eqn. (6.14) and eqn. (6.15), j indicates the time slot. ζ is parameter for memory level and $\zeta \in [0,1]$. \widetilde{V}_{j-1} and $\widetilde{\phi}_{j-1}$ are two random Gaussian variables with zero mean and unit variance [15][57]. By adjusting ζ , the degree of temporal correlation of velocity and direction between two consecutive time slots can be controlled and the velocity and direction in current time slot fluctuate around v_a and ϕ_a . This mobility model is validated to have no average speed decay problem and avoid sharp turn problem. Thus it's used for research and simulation in work sited in [7][25][26][47]. Besides simulation work is based on simulation results from random direction and random way point mobility models. Here, SMS mobility model is also elected for simulation to see how proposed schemes and algorithms work in more realistic motion tracks.

$$v_j = \zeta v_{j-1} + (1 - \zeta)v_a + \sqrt{1 - \zeta^2}\widetilde{V}_{j-1} \quad (6.14)$$

$$\phi_j = \zeta \phi_{j-1} + (1 - \zeta)\phi_a + \sqrt{1 - \zeta^2}\widetilde{\phi}_{j-1} \quad (6.15)$$

We continue to use TN6 and the same method to modify TPM in eqn. (6.4)-(6.7). The mobility model is selected as SMS. In SMS mobility model, ζ which is degree of temporal correlation of velocity and direction between two consecutive time slots is set to 0.5. Same with static schemes for mobile sensors, simulation for TCTP algorithm will be done. The speed is still 0.5 unit/sec, but the direction changes time to time. In our simulation, we sample the position of the mobile node every 1s, 2s, 4s, 6s, 8s and 10s. Every sampling time, the nodes position in the appropriate coordinate system is informed to BS. We set

time window for prediction as 0s, 0.5, 1s, 2s, 4s, 6s and 8s. We predict mobile sensor's position after prediction time 3s, 6s and 10s. For the number of detecting sensors, we choose 20 per each prediction test position, which corresponds on average to 0.39% of sensors in the network. The tracks in geographic domain and topological domain are shown in Figure 6.10.

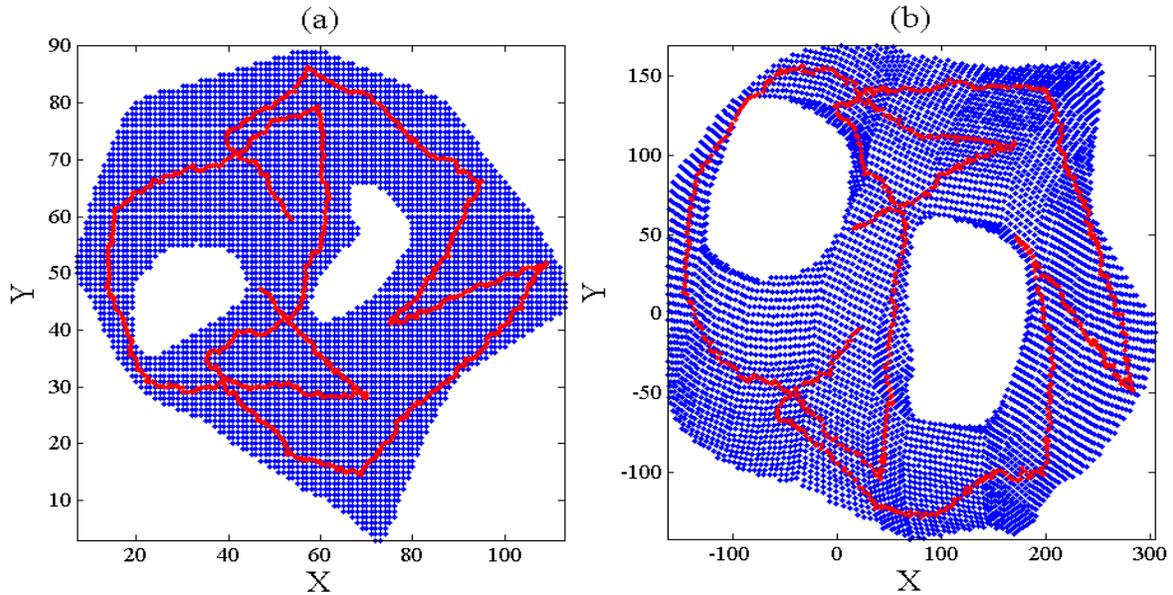


Figure 6.10: Mobile sensor's motion track in TN6 in SMS mobility model in a) geographic domain and b) topological domain

We also test the TCTP algorithm in the same testing conditions in which the static sensors in the network are equipped with GPS and know their GCs. Mobile sensor is not equipped with GPS and GCs are calculated by taking average of neighbors' GCs. Figure 6.11(a) shows the variation of detection failure rate vs. sampling time for test network in SMS mobility model in time window 2s when there are 20 detecting sensors in detection ellipse.

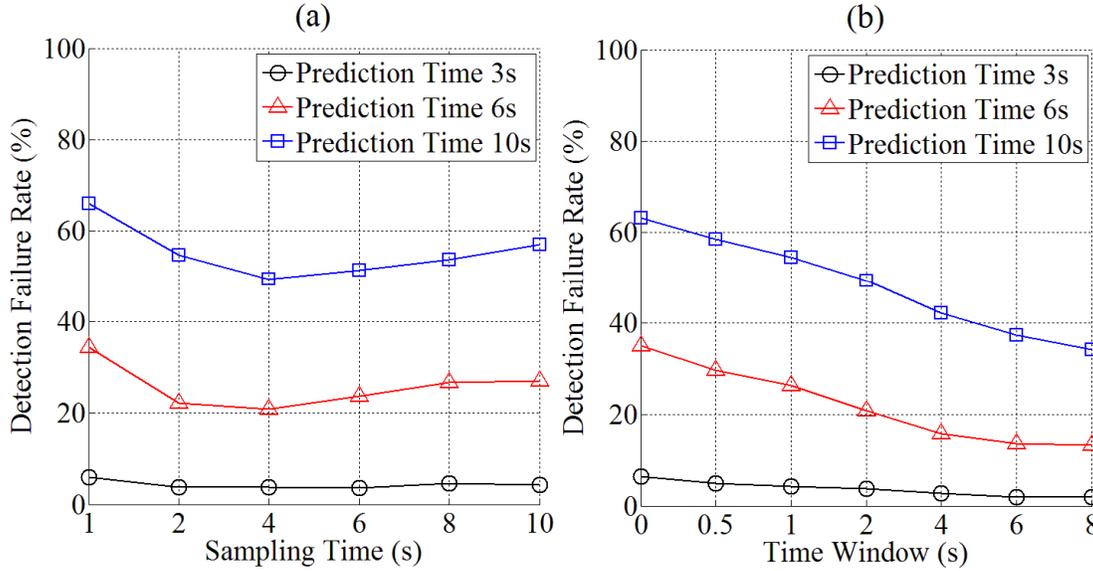


Figure 6.11: Detection failure rate for TN6 in SMS mobility model with 20 detection sensors vs. sampling time a) when time window is 2s and b) when sampling time is 4s

Figure 6.12 compares the variation of detection failure rate for TN6 in different time windows for tracking in TC and GC domains when sampling time is 6s. For short-term prediction like 3s and 6s under motion velocity of 0.5 unit/s, we can see that the detection failure rate in TC and GC domains are quite close to each other. For long-term prediction, e.g., 10s into future, the longer the time window the lower detection failure rate will be obtained. TC based approach comes within 12% of the GC based approach for a wide range of time windows, with lower rates at higher time windows (within 5% of GC based approach after time window 2s). The detecting performance difference between 2D-TCTP and GC based tracking and prediction algorithm become smaller if motion in chosen mobility model is more realistic. This also demonstrate the feasibility of applying in 2D-TCTP in tracking application in real WSNs because GC based tracking and prediction algorithm's decreased detection accuracy when motion velocity and direction change in unpredicted way.

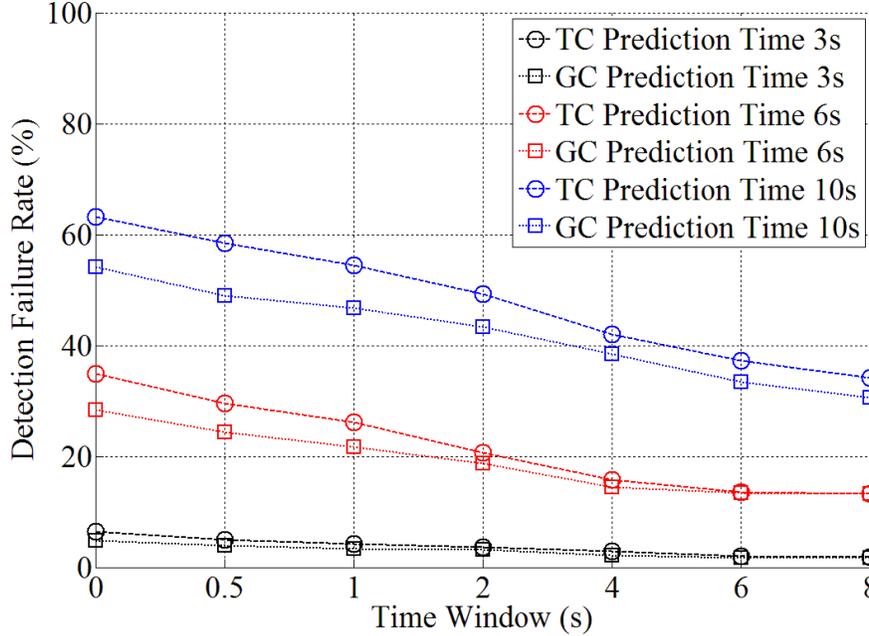


Figure 6.12: Detection failure rate comparison between TC domain and GC domain for TN6 in SMS model with 20 detecting sensors when sampling time is 4s

6.3 3D Topological coordinate based tracking and prediction algorithm

In previous sections, mobility tracking and prediction has been accomplished in 2D WSNs without geographic information and physical distance measurement, which can be used for mobile sensor communication and detection in 2D WSNs. Such mobility tracking and prediction algorithm is also needed in 3D WSNs. For example, SkyMedia camera system tries to catch flying objects' pictures and report them back to BS. Protected animals living underwater are equipped with sensor and USN tries to communicate with this mobile sensor. In such scenario where GPS has limitation to work in, a mobility tracking prediction algorithm without geographic information is highly desired. In this section, 2D-TCTP algorithm is extended to 3D WSNs and named as 3D-TCTP algorithm. A 3D volume networks is designed for simulation which can be applied in underwater, sky or indoor building environments and simulation results are presented in Section 6.3.1

2D-TCTP algorithm proposed before will be extended for 3D network and the basic idea of 2D-TCTP algorithm will remain the same. Predicted location at future time is linearly calculated based on

current sampled velocity and direction. In 2D-TCTP algorithm, mobile object equipped with sensor gets its VC/TC by taking average of neighbors' VC/TC, which can be extended to 3D network without any change. 3D velocity sampling and prediction calculation procedures are similar to 2D and 3D TCs will replace 2D TCs in current 3D velocity and direction calculation. Suppose current velocity and direction is sampled from current location $(x_{MT_i}, y_{MT_i}, z_{MT_i})$ at current time t_i and previous location $(x_{MT_{i-1}}, y_{MT_{i-1}}, z_{MT_{i-1}})$ at previous time t_{i-1} , sampled velocity and direction are given in eqn. (6.17), (6.18) and (6.19). Predicted future location $(x_{MT_{i+1}}, y_{MT_{i+1}}, z_{MT_{i+1}})$ after time t_p from current time t_i is given in eqn. (6.20), (6.21) and (6.22). D_T in eqn. (6.16) is the distance between current location and previous location in 3D topological domain.

$$D_T = \sqrt{(x_{MT_i} - x_{MT_{i-1}})^2 + (y_{MT_i} - y_{MT_{i-1}})^2 + (z_{MT_i} - z_{MT_{i-1}})^2} \quad (6.16)$$

$$V_T = \frac{D_T}{t_i - t_{i-1}} \quad (6.17)$$

$$\theta_T = \sin^{-1} \frac{z_{MT_i} - z_{MT_{i-1}}}{D_T} \quad (6.18)$$

$$\alpha_T = \cos^{-1} \frac{x_{MT_i} - x_{MT_{i-1}}}{\cos(\theta_T) D_T} \quad (6.19)$$

$$x_{MT_{i+1}} = x_{MT_i} + V_T t_p \cos(\alpha_T) \cos(\theta_T) \quad (6.20)$$

$$y_{MT_{i+1}} = y_{MT_i} + V_T t_p \sin(\alpha_T) \cos(\theta_T) \quad (6.21)$$

$$z_{MT_{i+1}} = z_{MT_i} + V_T t_p \sin(\theta_T) \quad (6.22)$$

The terms of sampling time, detecting sensors, prediction time window and detection failure rate will remain the same from 2D-TCTP algorithm. The Number of wall sensors is set up to a fixed number according to requirements of different applications. Differently, the shape of detection ellipse in 2D-TCTP is changed to 3D detection sphere. The center of detection sphere is the future location of mobile sensor estimated from eqn. (6.20), (6.21) and (6.22). Detection sphere will tolerate prediction errors in all directions and try to keep TCTP algorithm and related calculation still simple when extended for 3D network. The radius will be enlarged until the number of wall sensors reaches the fixed value. Detection

failure rate is the main evaluation metric for 3D-TCTP algorithm. Detection failure rate is still the probability that the mobile object is not detected by any detecting sensor in the time window Δt .

6.3.1 Simulation results

When evaluating the performance of 3D-TCTP algorithm, random waypoint mobility model [15] is used to generate the movement of mobile object in the physical domain for 3D-TCTP. Figure 6.13(a) shows the 3D geographic map and generated TPM of Test Network 8(TN8) used for 3D-TCTP algorithm. TN8 is in $20\text{unit} \times 20\text{unit} \times 20\text{unit}$ cube area and consists of 8000 sensor nodes in random topology. The communication range of each sensor is 1.3unit. The average node degree of TN8 is 8. 23 ENS anchors are selected using Double-ENS algorithm and the generated TPM of TN8 is shown in Figure 6.13(b). ENS anchor nodes are marked red triangles in Figure 6.13(a) and 6.13(b).

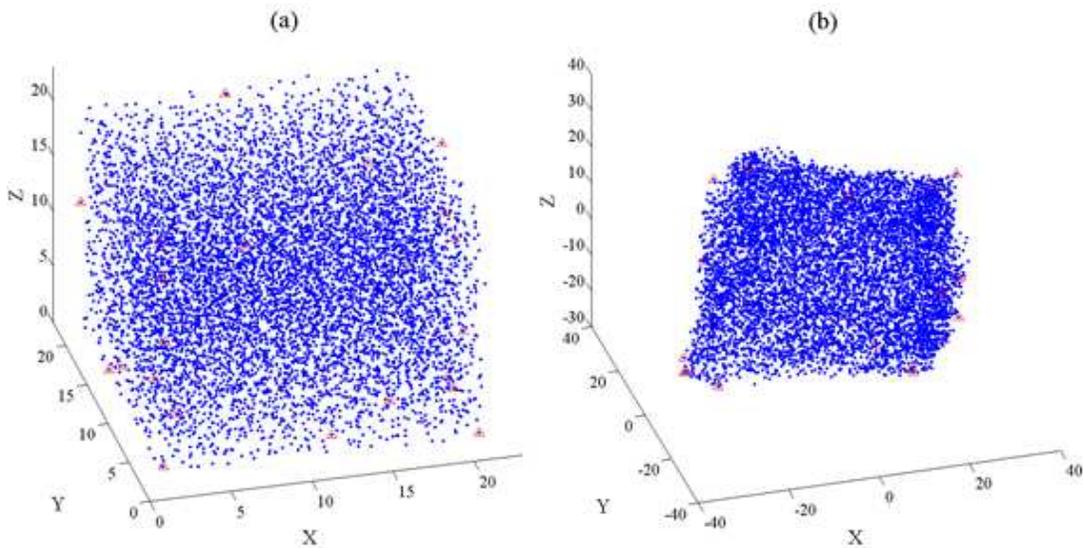


Figure 6.13: TN8's a) geographic map with 23 ENS anchors and b) generated TPM with 23 ENS anchors

Mobile object is set to move in a straight line in geographic domain. The velocity of mobile object in geographic domain is constant at 0.5units/s. Figure 6.14(a) gives mobile object's motion track in geographic domain which is marked as red star and Figure 6.14(b) gives the corresponding track in topological domain. There are approximately 1000 valid prediction test points along the track.

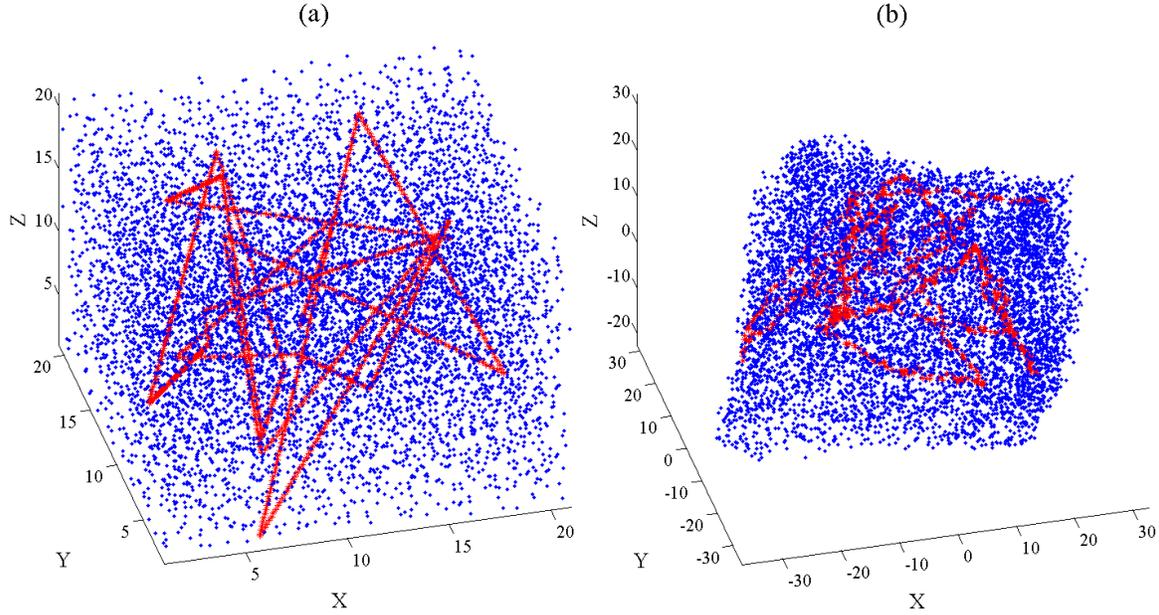


Figure 6.14: Mobile sensor's motion track in TN8 in a) geographic domain and b) topological domain

Similar to 2D simulation of TCTP presented in previous section, the mobile object's position is sampled every 0.5s, 1s, 2s, 3s, 4s and 5s. Every sampling time, the mobile object's VC/TC position in the appropriate coordinate system is informed to BS. Time window for prediction is set to 0s, 0.5, 1s, 2s, 4s, 6s and 8s.

We predict mobile object's position after prediction time 3s, 6s and 10s. For the number of detecting sensors, we still choose 20 per each prediction test position, which corresponds to 0.25% of sensors in the network. We also test the 3D-TCTP algorithm in the same testing conditions in which the static sensors in the network are equipped with GPS and know their GCs. Mobile object is not equipped with GPS and GCs are calculated by taking average of neighbors' GCs. Figure 6.15(a) shows the variation of detection failure rate vs. sampling time for TN8 under random waypoint mobility model in time window 2s when there are 20 detecting sensors in detection sphere. Very similar to the 2D-TCTP's simulation results, sampling time for 3D-TCTP can neither be too short nor too long in order to obtain accurate average position for current velocity and direction calculation.

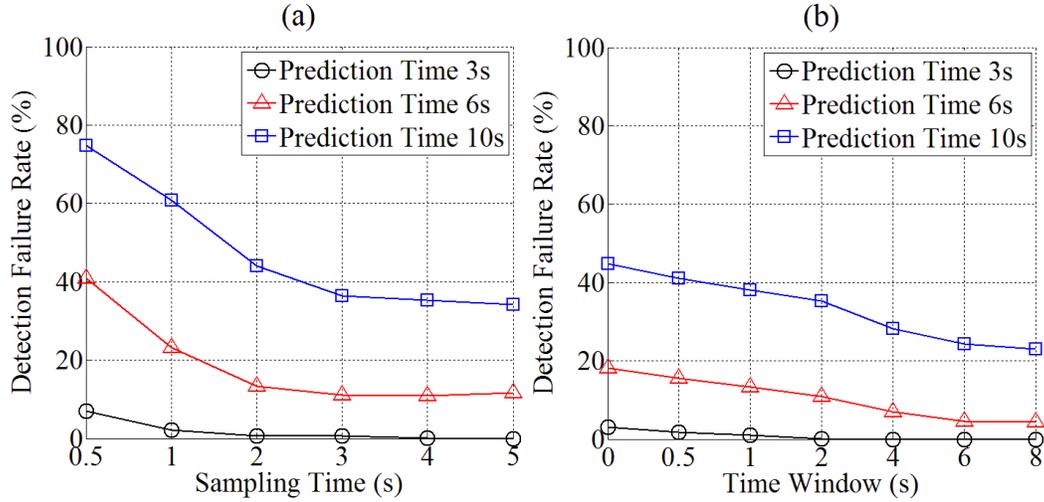


Figure 6.15: Detection failure rate for TN8 in random waypoint mobility model with 20 detection sensors vs. sampling time a) when time window is 2s and b) when sampling time is 4s

Figure 6.16 compares the variation of detection failure rate for TN8 in different time windows in TC and GC domains when sampling time is 4s. Still similar to 2D-TCTP simulation results in previous section, detection failure rate in TC and GC domains are quite close to each other for short-term prediction like 3s and 6s under motion velocity of 0.5units/s. Long-term prediction like 10s into future in 3D sensor network is still a challenge both in TC and GC domain. TC based approach comes within 12% of the GC based approach for a wide range of time windows, with lower rates at higher time windows (within 6% of GC based approach after time window 2s). Figure 6.17 shows detection failure rate results when the number of detection sensors is increased to 40 (0.50% of sensors in the network), TC based approach comes within 5% of the GC based approach. After TCTP is extended in 3D sensor network, performance of TC domain tracking and prediction algorithm is very close to the one in GC domain. More importantly, the 3D extension of TCTP is free of complicated modification and reduces power due to absence of localization equipment and algorithms. 3D-TCTP can be used for tracking and prediction in sky or underwater sensor networks where more hardware limitation from environment exists.

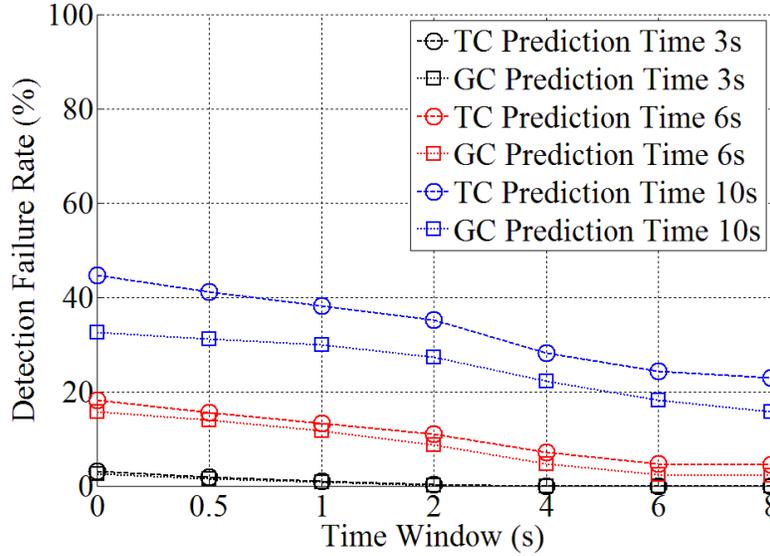


Figure 6.16: Detection failure rate comparison between TC domain and GC domain for TN8 with 20 detection sensors when sampling time is 4s

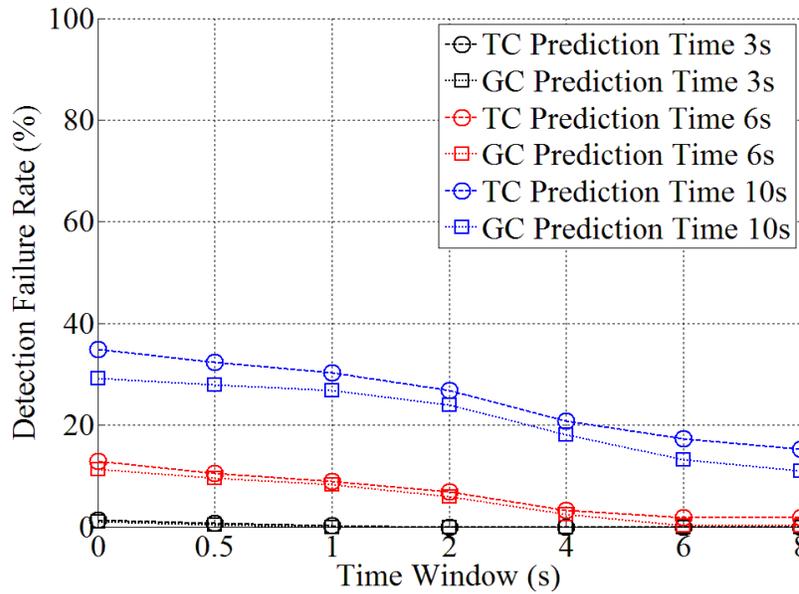


Figure 6.17: Detection failure rate comparison between TC domain and GC domain for TN8 with 40 detection sensors when sampling time is 4s

6.4 Summary

In this chapter, a prediction based tracking algorithm for 2D WSNs called 2D-TCTP to reach and detect mobile sensor node in continuous movement is presented and simulation results are analyzed.

Simulation based results demonstrate that 2D-TCTP achieves similar performance compared with physical information based approaches. This paper also paves the way for use of TC domain for many other sensor network applications that usually rely on GCs. TCTP shows effectiveness in 2D network and competitive detection performance can be achieved without any geographic and physical localization method. Additionally, Extension of 2D-TCTP for 3D WSNs to reach and detect mobile sensor node in continuous movement is also presented and related simulation results are analyzed. 3D-TCTP shows effectiveness in 3D volume network and competitive detection performance can be achieved without using any geographic localization methods in 3D WSNs.

CHAPTER 7

SUMMARY AND FUTURE WORK

7.1 Summary and conclusion

Wireless Sensor Networks (WSNs) consist of distributed sensors to monitor physical or environmental condition such as temperature and humidity. Sensors cooperatively exchange their data through the network among each other or to a Base Station (BS) or other terminal. As the demands from WSNs application grow day by day, WSNs need to be widely deployed in all kinds of environments, e.g., underwater, indoor building, sky, and factories. Also, the recent concept of Internet of Things imposes additional intelligence requirements for sensor networks that bridging the gap between physical world and computer processors. In all, technology changes require future WSNs to be able to contain large number of sensors with low deployment and energy costs, and at the same time require powerful actuation performances from the sensors in various physical environments.

Virtual Coordinate System (VCS) shows great potential for these future WSNs because it's free of geographic distance measurement requirements, and are also adjustable to different physical environments. In this thesis, the main focus is on the extensions of current VCS based techniques in such as Extreme Node Search (ENS) for anchor placement and 2D Geo-Logical Routing (2D-GLR) algorithm for routing in 3D WSNs. In Chapter 3, the current ENS and 2D-GLR algorithms are extended for 3D sensor networks. 3D-GLR with ENS anchor placement achieves much greater routing performance compared with 3D Greedy Distributed Spanning Tree Routing (3D-GDSTR), a routing algorithm that requires the Geographic Coordinates (GCs), in both 3D networks with random and fixed topologies, in both concave and non-concave shapes, with low and high average node degree and in different scales.

This thesis for the first time demonstrates how tracking and prediction of the position of mobile nodes can be done purely using the Topological Coordinate (TC) system which only contains topology information of network without exact geographic distance measurement. TC based tracking and prediction algorithms are presented for both 2D and 3D networks. VC generation schemes for newly

deployed sensors are proposed in Chapter 4, without requiring the regeneration of coordinates for the entire network. Among the three schemes evaluated, the average based scheme is able to provide effective VCs for a new sensor to get its approximate location in virtual domain. Effectiveness of new coordinates is characterized by the routing performance of new sensors in both receiving and passing packets in the network. The existing Topology Preserving Maps (TPMs) exhibit folding problem closer to the edges, which decreases the accuracy of layout information for tracking purposes. Modification schemes are developed for TPMs in Chapter 5, in order to improve this boundary compression. The improved accuracy of TPMs results in better guide map for Topological Coordinate based Tracking and Prediction (TCTP), an algorithm for mobility tracking, prediction and detection in 2D WSNs, which is proposed in Chapter 6 and named as 2D-TCTP for 2D WSNs. TCs are used as 2D location coordinates to obtain current velocity and direction of a mobile node and to predict the future location using linear extrapolation. Not limited to 2D networks, TCTP is also easily extended to 3D WSNs for tracking and prediction, which is named as 3D-TCTP in Chapter 6.

In summary, extension of existing VC related algorithms, ENS and 2D-GLR, to 3D sensor networks has been accomplished. 3D-GLR algorithm doesn't involve any planarization computation or geographic localization hardware/software. With the help of the ENS anchor placement, 3D-GLR algorithm shows very strong effectiveness and high efficiency for 3D sensor networks, compared with existing 3D-GPSTR algorithm that is based on GCs with more adjustability and energy saving. Simple VC generation schemes in virtual domain are proposed for mobile sensor nodes for communication with dynamically deployed sensor nodes in the network. By acquiring the VCs of mobile nodes' neighbors, the mobile node estimates its location in the virtual domain. Three schemes for generating VCs are evaluated for routing performance when mobile sensor nodes act as destination and intermediate nodes in network. Generating VCs is the very initial step for mobile sensor's various applications in VCS based WSNs. The schemes which we evaluated can satisfy both effectiveness and efficiency in VCS based 2D and 3D networks. Considering the velocity of mobile sensors, tracking and prediction of position of mobile targets using sensor networks have been accomplished using TCs in the proposed new tracking and

prediction algorithm called TCTP algorithm. TCTP does not require physical distance information. Instead of doing tracking in geographical domain, it does the tracking in TC domain. Our simulation results show that even without any geographic information, tracking and prediction of positions of a mobile target using TCs is effective. Both 2D-TCTP and 3D-TCTP algorithm have competitive performance compared with the same algorithm operating in the GC domain for a wide range of mobility parameters in both 2D and 3D networks.

The research work presented in this thesis paves the way for applying and designing VCS based techniques in 3D WSNs and MWSNs for the first time. Routing and mobility tracking and prediction in 3D WSNs have been accomplished without any geographic information and physical distance measurements.

7.2 Future work

Research in this thesis paves the way for applying VCS based techniques for anchor placement and routing in 3D sensor networks. 3D sensor networks are mainly used in underwater area so far. In our research, we assume that sensors obtain their VCs from anchors' initial floodings of VC generation messages, and the nodes hold the same VCs all the time. However, in real USNs, one big challenge in localization comes from the dynamic topology changes. This is due to the nature of underwater environment where ocean currents and waves can easily reorganize the placement of sensors in unpredictable ways [40]. The change of geographic topology results in location change in virtual or topological domain for sensor nodes. In other words, network topology in virtual and topological domain changes over time. In this case, sensor networks with dynamic topology related to time domain can also be researched and explored. Thus a dynamic VC generation scheme is required for networks to avoid re-flooding VC generation messages from anchors over and over again. As a result, dynamic VC generation and intelligent self-reorganization in VCS based 3D USNs are important and promising research topics, which will help extend application of VCS from research environments to real application environments.

Additionally, future work also includes further proof of ENS algorithm's effectiveness in anchor selection and design of topology evaluation matrix for 3D TPMs. More evaluation matrix for 2D-GLR and 3D-GLR algorithm rather than average routability and average path length can also be designed.

REFERENCES

- [1] H. Akcan, V. Kriakov, H. Bronnimann, and A. Delis, "Gps-free Node Localization in Mobile Wireless Sensor Networks," in *MobiDE '06: Proceedings of the 5th ACM international workshop on Data engineering for wireless and mobile access*, pp. 35–42, 2006.
- [2] I. Aumndson and X.D.Koutsoukos, "A Survey on Localization for Mobile Wireless Sensor Networks," *Workshops on Mobile Entity Localization and Tracking (MELT), Springer Lecture Notes in Computer Science (LNCS5801)*, pp. 235–254, Orlando, Florida, 2009.
- [3] I. F. Akyildiz, S. Weilian, Y. Sankarasubramaniam and E. E. Cayirci, "A Survey on Sensor Networks," *Proc. IEEE Communications Magazine*, pp. 102–114, 2002.
- [4] J. N. Al-Karaki and A. E. Kamal, "Routing Techniques in Wireless Sensor Networks: A Survey," *Wireless Communications, IEEE*, vol. 11, no. 6, pp. 6–28, 2004.
- [5] K. Akkaya and M. Younis, "A Survey on Routing Protocols for Wireless Sensor Networks", *journal of Ad Hoc Networks*, pp. 325–349, 2005.
- [6] A. Boukerche, H.A.B.Oliveira, E. F. Nakamura and A. A. F. Loureiro, "Localization Systems for Wireless Sensor Networks," *Wireless Communications, IEEE*, vol. 14, no. 6, pp.6–12, 2007.
- [7] A. B. Bagayoko, B. Paillassa and R. Dhaou, "Practical Link Reliability for Ad Hoc Routing Protocol," *IEEE Vehicular Technology Conference (VTC Fall)*, pp. 1–5, 2011.
- [8] J. Bruck, J. Gao and A. Jiang, "Localization and Routing in Sensor Networks by Local Angle Information", *Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing (MobiHoc)*, pp. 181–192, New York, USA, 2005.
- [9] J. Bachrach and C. Taylor, "Localization in Sensor Networks," Ch. 9, *Handbook of Sensor Networks*, Stojmenovic (Editor), John Wiley 2005.
- [10] A. Caruso, S. Chessa, S. De and A. Urpi, "GPS free Coordinate Assignment and Routing in Wireless Sensor Networks," *Proc. 24th IEEE Joint Conf. of Computer and Communications Societies*, vol. 1, pp. 150–160, 2005.
- [11] C. Y. Chong and S. P. Kumar, "Sensor Networks: Evolution, Opportunities and Challenges," *Proceedings of the IEEE*, vol. 91, no. 8, pp. 1247–1256, 2003.
- [12] G. Carofiglio, C. Chiasserini, M. Garetto and E. Leonardi, "Route Stability in MANETs under the Random Direction Mobility Model," *IEEE Transactions on Mobile Computing*, vol. 8, no. 9, pp. 1167–1179, 2009.
- [13] Q. Cao and T. Abdelzaher, "Scalable Logical Coordinates Framework for Routing in Wireless Sensor Networks," *ACM Transactions on Sensor Networks*, vol. 2, no. 4, pp. 557–593, 2006.
- [14] S. Capkun, M. Hamdi, and J.-P. Hubaux, "Gps-free Positioning in Mobile Ad Hoc Networks," *Cluster Computing*, vol. 5, no. 2, pp. 157–167, 2002.

- [15] T. Camp, J. Boleng and V. Davies, "A Survey of Mobility Models for Ad Hoc Network Research", *Wireless Communication and Mobile Computing: Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, vol. 2, no. 5, pp. 483–502, 2002.
- [16] D. C. Dhanapala and A. P. Jayasumana, "CSR: Convex Subspace Routing Protocol for WSNs," *Proc. 33rd Annual IEEE Conference on Local Computer Networks*, pp. 1–6, Zürich, Switzerland, 2009.
- [17] D. C. Dhanapala and A. P. Jayasumana, "Topology Preserving Maps from Virtual Coordinates for Wireless Sensor Networks," *Proc. 35th IEEE Conf. on Local Computer Networks*, pp. 136–143, Denver, USA, 2010.
- [18] D. C. Dhanapala and A. P. Jayasumana, "On Boundary Detection of 2-D and 3-D Wireless Sensor Networks", *Proc. IEEE Global Telecommunications Conference (Globecom)* , pp. 1–5, Houston, USA, 2011.
- [19] D. C. Dhanapala and A. P. Jayasumana, "Anchor Selection and Topology Preserving Maps in WSNs — A Directional Virtual Coordinate based Approach", *Proc. IEEE 36th Conf. on Local Computer Networks*, pp. 571–579, Bonn, Germany, 2011.
- [20] D. C. Dhanapala and A. P. Jayasumana, "Geo-Logical Routing in Wireless Sensor Networks", *Proc. 8th IEEE Conf. on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, pp. 305 – 313, Salt Lake City, USA, 2011.
- [21] D. C. Dhanapala and A. P. Jayasumana, "Directional Virtual Coordinate Systems for Wireless Sensor Networks," *IEEE International Conference on Communications (ICC)*, pp. 1–6, 2011.
- [22] R. Flury and R. R. Wattenhofer, "Randomized 3D Geographic Routing," *INFOCOM 2008, The 27th Conference on Computer Communications. IEEE* , pp. 13–18, 2008.
- [23] C. F. García-Hernández, P.H. Ibarguengoytia-González, J. García-Hernández, and J.A. Pérez-Díaz, "Wireless Sensor Networks and Applications: A Survey," *Int. Journal of Computer Science and Network Security(IJCSNS)*, pp. 264–273, 2007.
- [24] J. J. Garcia-Luna-Aceves and M. Spohn, "Source-tree Routing in Wireless Networks," In *Proceedings of the 7th International Conference on Network Protocols (ICNP)*, pp. 273–282, 1999.
- [25] L. Guo, K. Harfoush and H. Xu, "Analyzing Path Dynamics in Mobile Ad Hoc Networks Using a Smooth Mobility Model," *IEEE Wireless Communications and Networking Conference*, pp.1–6, 2009.
- [26] L. Guo, K. Harfoush and H. Xu, "The Impacts of Network Configuration and Node Mobility on Path Dynamics in Mobile Ad Hoc Networks," *IEEE Symposium on Computers and Communications*, pp. 851–856, 2009.
- [27] J. Heidemann, W. Ye, J. Wills, A. Syed and L. Yuan, "Research Challenges and Applications for Underwater Sensor Networking," *Wireless Communications and Networking Conference*, vol. 1, pp. 228–235, 2006.

- [28] J. Hsu, C. C. Chen and C. Chi, "Short-term Prediction-based Optimistic Object Tracking Strategy in Wireless Sensor Networks", Proc. Int. Conf. on Innovative Mobile and Internet Services in Ubiquitous Computing, pp. 78–85, 2011.
- [29] L. Hanzo II, S. M. Mostafavi and R. Tafazolli, "Connectivity-Related Properties of Mobile Nodes Obeying the Random Walk and Random Waypoint Mobility Models," IEEE Vehicular Technology Conference, pp. 133–137, 2008.
- [30] L. Hu and D. Evans, "Localization for Mobile Sensor Networks," in MobiCom '04, Philadelphia, pp. 45–57, PA, USA, 2004.
- [31] A. P. Jayasumana, Q. Han and T. Illangasekare, "Virtual Sensor Networks - A Resource Efficient Approach for Concurrent Applications," Proc. 4th International Conference on Information Technology: New Generations (ITNG 2007), pp. 111–115, Las Vegas, NV, 2007.
- [32] Y. Jiang, D. C. Dhanapala and A. P. Jayasumana, "Tracking and Prediction of Mobility without Physical Distance Measurements in Sensor Networks," In Proceedings of IEEE International Conference on Communications (ICC), 2013.
- [33] P. Johansson, T. Larsson, N. Hedman, B. Mielczarek, and M. Degermark, "Routing Protocols for Mobile Ad Hoc Networks - A Comparative Performance Analysis," In Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM), pp. 195–206, 1999.
- [34] B. Karp and H. T. Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks," Proc. 6th ACM/IEEE Int. Conf. Mobile Computing and Networking (MobiCom 2000), pp. 243–254, 2000.
- [35] B. Leong, B. Liskov and R. Morris, "Geographic Routing without Planarization", Proceedings of the 3rd conference on Networked Systems Design & Implementation, pp. 25–25, 2006.
- [36] C. Liu and J. Wu, "Efficient Geometric Routing in Three Dimensional Ad Hoc Networks," IEEE INFOCOM 2009, pp. 2751–2755, 2009.
- [37] K. Liu and N. Abu-Ghazaleh, "Aligned Virtual Coordinates for Greedy Routing in WSNs," Proc. IEEE Int. Conf. on Mobile Adhoc and Sensor Systems, pp. 377–386, Vancouver, Canada, 2006.
- [38] S. Munir, B. Ren, W. Jiao, B. Wang, D. Xie, and J. Ma, "Mobile Wireless Sensor Network: Architecture and Enabling Technologies for Ubiquitous Computing," in AINAW '07: 21st International Conference on Advanced Information Networking and Applications Workshops, vol. 2, pp. 113–120, 2007.
- [39] N. Naderan, M. Dehghan, H. Pedram, "Mobile Object Tacking Techniques in Wireless Sensor Networks," Ultra Modern Telecommunications & Workshops, pp. 1–8, 2009.
- [40] S. S. Naik and M. J. Nene, "Realization of 3D Underwater Wireless Sensor Networks and Influence of Ocean Parameter on NodeLocation Estimation", International Journal of Wireless & Mobile Networks (IJWMN), vol. 4, no. 2, 2012.
- [41] M. Neri, A. Campi, R. Suffritti, F. Grimaccia, P. Sinogas, O. Guye, C. Papin, T. Michalareas, L. Gazdag and I. akkolainen, "SkyMedia - UAV-based Capturing of HD/3D Content with WSN

Augmentation for Immersive Media Xperiences,""Multimedia and Expo (ICME), 2011 IEEE International Conference on, pp.1–6, 2011.

- [42] C. D. Ortiz, J. M. Puig, C. E. Palau, and M. Esteve, "3D Wireless Sensor Network Modeling and Simulation," in *SENSORCOMM '07: Proceedings of the 2007 International Conference on Sensor Technologies and Applications*, IEEE Computer Society, pp. 307–312, Washington, DC, USA, 2007.
- [43] H. A. B. F. Oliveira, E. F. Nakamura, A. A. F. Loureiro and A. Boukerche, "Error Analysis of Localization Systems for Sensor Networks," *Proc. 13th ACM Int. Workshop on Geographic Information Systems*, pp. 71–78, 2005.
- [44] S. Poduri, S. Patten, B. Krishnamachari and G. S. Sukhatme, "Sensor Network Configuration and The Curse of Dimensionality," *Proc. Third Workshop on Embedded Networked Sensors (EmNets 2006)*, Cambridge, MA, USA. 2006.
- [45] A. Rao, S. Ratnasamy, C. Papadimitriou, S. Shenker, and I. Stoica, "Geographic Routing without Location Information," *Proc. 9th Int. Conf. on Mobile Computing and Networking*, pp. 96–108, San Diego, USA, 2003.
- [46] I. Rubin and C. Choi, "Impact of The Location Area Structure on The Performance of Signaling Channels in Wireless Cellular Networks," *IEEE Communications Magazine*, pp. 108–115, 1997.
- [47] M. Sichitiu and V. Ramadurai, "Localization of Wireless Sensor Networks with A Mobile Beacon", *IEEE International Conference on Mobile Ad-hoc and Sensor Systems*, pp. 174–183, Fort Lauderdale, USA, 2004.
- [48] A. Y. Teymorian, W. Cheng, L. Ma, X. Cheng, X. Lu and Z. Lu, "3D Underwater Sensor Network Localization," *Mobile Computing*, *IEEE Transactions on*, vol. 8, no. 12, pp. 1610–1621, Dec. 2009.
- [49] M. Tubaishat and S. Madria, "Sensor Networks: An Overview," *IEEE Potentials*, pp. 20-23, 2003.
- [50] M. J. Tsai, H. Y. Yang and W. Q. Huang, "Axis-Based Virtual Coordinate Assignment Protocol and Delivery-Guaranteed Routing Protocol in Wireless Sensor Networks," *INFOCOM 2007, 26th IEEE International Conference on Computer Communications*. IEEE, pp. 2234–2242, 2007.
- [51] D. V. K. Viswanath, K. M. Krishna, H. Hexmoor and S. Singh, "Robot Navigation in a 3D World Mediated by Sensor Networks," *Integration of Knowledge Intensive Multi-Agent Systems, International Conference on KIMAS 2007*, pp. 272–276, 2007.
- [52] M. K. Watfa, "Practical applications and connectivity: Algorithms in Future Wireless Sensor Networks," *International Journal of Information Technolog*, vol. 4, pp. 18–28, 2007.
- [53] Y. Xu, J. Winter and W. C. Lee, "Dual Prediction-Based Reporting for Object Tracking Sensor Networks," *Proc. First Int. Conf. on Mobile and Ubiquitous Systems: Networking and Services*, pp. 154–163, 2004.
- [54] H. Yang, and B. Sikdar, "A Protocol for Tracking Mobile Targets Using Sensor Networks," *Proc. First IEEE Int. Workshop on Sensor Network Protocols and Applications*, pp. 71–81, 2003.

- [55] C. Zhu, L. Shu, T. Hara, L. Wang and S. Nishio, "Research Issues on Mobile Sensor Networks," Communications and Networking in China (CHINACOM), 2010 5th International ICST Conference, pp. 1–6, Beijing, China, 2010.
- [56] J. Zhou, Y. Chen, B. Leong, and P. Sundar, "Practical 3D Geographic Routing for Wireless Sensor Networks," in Proc. of SenSys, pp. 337–350, 2010.
- [57] M. Zhao and W. Wang, "A Novel Semi-Markov Smooth Mobility Model for Mobile Ad Hoc Networks," IEEE Global Telecommunications Conference, GLOBECOM '06, pp.1–5, 2006.
- [58] M. Zonoozi and P. Dassanayake, "User Mobility Modeling and Characterization of Mobility Pattern," IEEE Journal on Selected Areas in Communications, pp. 1239–1252, 1997.
- [59] Gorgia Institue of Technology: Underwater Acoustic Sensor Networks, available at: <http://www.ece.gatech.edu/research/labs/bwn/UWASN/work.html>
- [60] Ohio State University: Wireless Multimedia Sensor Networks , available at: http://www2.ece.ohio-state.edu/~ekici/res_wmsn.html
- [61] Wikipedia: Sensor node, available at: http://en.wikipedia.org/wiki/Sensor_node

APPENDIX A

SIMULATOR OF TCTP ALGORITHM

MATLAB codes for TCTP algorithm are given below. Section A.1 is the main function of TCTP algorithm.

A.1 Main function

```
%INPUT: Network, Mobile sensor's track, Sampling time, Prediction time,
%Time Window, Number of detection sensors
%OUTPUT: Detection failure rate in time window
TimeSlot = Tsample/Ttest; % get time slot of sampling time over testing time
%do SVD operation to matrix of anchor set
anchormatrix = zeros(Anchor_NO,Anchor_NO);
    for j =1:Anchor_NO
        for i = 1:Anchor_NO
            anchormatrix(i,j) = Grid1(i,AnchorArray(1,j));
        end
    end
[U,S,V] = svd(anchormatrix'); %get SVD components of network

[size1,size2] = size(trackxy);
j1 = 1;%flag for testtrack
for i = 1:1:size2
    if mod(single(trackxy(1,i)),single(Ttest)) == 0%if it's testing time
        c = 0; %neighbor counter
        for i2 = 1:NoNodes
            if sqrt((trackxy(2,i)-XY(i2,1))^2 + (trackxy(3,i)-XY(i2,2))^2) <=
                communicaterange %find neighbors within communication range
                c = c+1; %how many valid neighbors
                AddinodeNeighbor(c+1,1) = i2;
            end
        end
        %get average VC from neighbors
        AddinodeNeighbor(1,1)=c;
        ADDIVC = zeros(Anchor_NO(1,1),1);
        for i3 = 1:Anchor_NO(1,1)
            for j = 2:c+1
                if AddinodeNeighbor(j,1) ~= 0
                    C(1,j-1) = Grid1(i3,AddinodeNeighbor(j,1));
                else
                    C(1,j-1) = 0;
                end
            end
        end
        %if only one neighbor is found
        if AddinodeNeighbor(1,1) == 1
            ADDIVC(i3,1) = sum(C(1,:));
            A2 = randperm(Anchor_NO(1,1));
            ADDIVC(A2(1,1),1) = ADDIVC(A2(1,1),1)+1;
        else
            ADDIVC(i3,1) = sum(C(1,:))/AddinodeNeighbor(1,1);
        end
    end
end
```

```

end
MAXI = ADDIVC'*V; %do SVD operation to average VC of mobile sensor
%get location of mobile sensor in TC domain
currentX = MAXI(1,2);
currentY = MAXI(1,3);

%record motion track
ADDIVC = [];
testtrackxy(1,j1) = trackxy(1,i);%current time
testtrackxy(2,j1) = trackxy(2,i);%real x coordinate in XY for testing
testtrackxy(3,j1) = trackxy(3,i);%real y coordinate in XY for testing
testtrackxy(4,j1) = currentX;%sampled average x coordinate in TC
testtrackxy(5,j1) = currentY;%sampled average y coordinate in TC
j1 = j1+1;
end
end

TestNO=j1-1;% testing points including invalid ones
predictTC = zeros(3,TestNO);% predicted future location in TC domain
for i = 1:TestNO
    currentTCx = testtrackxy(4,i);
    currentTCy = testtrackxy(5,i);
    %if the time exceeds the time which can be predicted
    if single(testtrackxy(1,i)+ Tpredict) > single(testtrackxy(1,TestNO)) || single(testtrackxy(1,i)) < single(Tsample)
        PredictTC(1,i) = 0;
        PredictTC(2,i) = 0;
        PredictTC(3,i) = 0;
    else
        %if mobile sensor stops moving
        if testtrackxy(4,i-TimeSlot) == currentTCx && testtrackxy(5,i-
            TimeSlot) == currentTCy %
            PredictTC(1,i) = 0;
            PredictTC(2,i) = 0;
            PredictTC(3,i) = 0;
        else
            PredictTC(1,i) = 1;
            Vtc(1,i) = testtrackxy(1,i);% Vtc for all testing points
            DistanceTC = sqrt((testtrackxy(4,i)-testtrackxy(4,i-
                TimeSlot))^2+(testtrackxy(5,i)-testtrackxy(5,i-TimeSlot))^2);
            Vtc(2,i) = DistanceTC/Tsample;% Velocity
            if (Vtc(2,i) == 0 && Vtc(3,i) == 0) %if mobile sensor stops moving
                PredictTC(1,i) = 0;
                PredictTC(2,i) = 0;
                PredictTC(3,i) = 0;
            else
                predictdistance1 = Vtc(2,i)*Tpredict; %linear prediction
                [predictTCx,predictTCy]=solve('(predictTCx-prea)/pre = (prea-
                    prec)/sqrt((prea-prec)^2+(preb-pred)^2)',
                    '(predictTCy-preb)/pre = (preb-pred)/sqrt((prea-
                    prec)^2+(preb-pred)^2)',predictTCx,predictTCy);
                prea = currentTCx;
                preb = currentTCy;
                prec = testtrackxy(4,i-TimeSlot);
                pred = testtrackxy(5,i-TimeSlot);
                pree = predictdistance1;
                predictTCx=eval(predictTCx);
            end
        end
    end
end

```

```

    predictTCy=eval(predictTCy);
    PredictTC(2,i) = predictTCx; %predicted X coordinate in TC domain
    PredictTC(3,i) = predictTCy; %predicted Y coordinate in TC domain
    PredictTC(4,i) = 0;
end
end
end
WALLSENSOR = zeros(13,TestNO);%detection sensors
Windowresult = zeros(TimewindowNO,TestNO);
OutofBoundaryCounter = 0;
for i = 2:TestNO
    if PredictTC(1,i) ~= 0 % valid testing point
        detectrangea = 1; % major axis of detection ellipse
        detectrangeb = 0.2; % minor axis of detection ellipse
        ee=1; %counter of wall sensors
        mode = 1;
        if outboundary == 0
            while ee < NOWallsensor+1
                if mode == 1
                    ellipsea = detectrangea;
                    ellipseb = detectrangeb;
                end
                if mode ~= 1
                    ellipsea = (1+mode/100)*ellipsea;
                    ellipseb = (1+mode/100)*ellipseb;
                end
                % draw the ellipse
                ellipsetan = (testtrackxy(5,i)-testtrackxy(5,i-
                TimeSlot))/(testtrackxy(4,i)-testtrackxy(4,i-TimeSlot));
                if ellipsetan < 0
                    ellipseangle = pi+atan(ellipsetan);
                else
                    ellipseangle = atan(ellipsetan);
                end
                ellipseangle = ellipseangle-pi/2;
                % get detecting sensors inside ellipse
                for e= 1:NoNodes
                    X =
                    PSVD(e,1)*cos(ellipseangle)+PSVD(e,2)*sin(ellipseangle);
                    Y = PSVD(e,2)*cos(ellipseangle)-
                    PSVD(e,1)*sin(ellipseangle);
                    newx0 = x0*cos(ellipseangle)+y0*sin(ellipseangle);
                    newy0 = y0*cos(ellipseangle)-x0*sin(ellipseangle);
                    if (((X-newx0)^2)/(ellipsea^2))+(((Y-
                    newy0)^2)/(ellipseb^2))<=1
                        WALLSENSOR(ee+3,i) = e;%ID of detecting sensor nodes
                        ee=ee+1;
                    end
                end
            end
            if ee < NOWallsensor+1 %remove extra detecting sensor nodes
                mode=mode+1;
                for eee = 1:ee+3
                    WALLSENSOR(eee,i) = 0;
                end
                ee=1;
            end
        end
    end
end

```

```

end
TCNO = ee-1;
WALLSENSOR(1,i) = TCNO;%number of wall sensors
WALLSENSOR(2,i) = ellipsea;%long axis of detect ellipse
WALLSENSOR(3,i) = ellipseb;%short axis of detect ellipse
end
end
if outboundary == 1 %if mobile sensor moving of out boundary of network
    OutofBoundaryCounter = OutofBoundaryCounter+1;
    for h = 1:BOUNDARYNODENO %get nearest boundary nodes of network
        boundarydistance(h,1) = BOUNDARYNODEID(h,1);
        boundarydistance(h,2) = sqrt((PSVD(BOUNDARYNODEID(h,1),1)-
            x0)^2+(PSVD(BOUNDARYNODEID(h,1),2)-y0)^2);
    end
    boundarydistance = sortrows(boundarydistance,2);
    WALLSENSOR(1,i) = NOWallsensor;
    WALLSENSOR(2,i) = 0;
    WALLSENSOR(3,i) = 0;
    for h = 1:NOWallsensor
        WALLSENSOR(3+h,i) = boundarydistance(h,1);
    end
    boundarydistance = [];
end
%remove extra detecting sensor nodes
if WALLSENSOR(1,i) > NOWallsensor
    for mm = 1:WALLSENSOR(1,i)
        wallsort(mm,1) = WALLSENSOR(mm+3,i);
        wallsort(mm,2) = sqrt((XY(wallsort(mm,1),1)-
            x0)^2+(XY(wallsort(mm,1),2)-y0)^2);
        wallsort = sortrows(wallsort,2);
    end
    for mm = 1:WALLSENSOR(1,i)
        WALLSENSOR(3+mm,i) = 0;
    end
    for mm = 1:NOWallsensor
        WALLSENSOR(3+mm,i) = wallsort(mm,1);
    end
    WALLSENSOR(1,i) = NOWallsensor;
    wallsort = [];
end
%check if mobile sensor is detected by detecting sensor at future time
futuretime = testtrackxy(1,i)+Tpredict; % future time
for i2 = 1:TimewindowNO %check in different time window
    timewindow = TimeWindow(1,i2);
    for i3 = 1:size2 %
        if single(trackxy(1,i3)) >= single(futuretime-timewindow) &&
            single(trackxy(1,i3)) <= single(futuretime+timewindow)
            for i4 = 4:3+NOWallsensor
                wallsensorx = XY(WALLSENSOR(i4,i),1);
                wallsensory = XY(WALLSENSOR(i4,i),2);
                if sqrt((wallsensorx-trackxy(2,i3))^2+(wallsensory-
                    trackxy(3,i3))^2) <=communicaterange
                    Windowresult(i2,i) = 1;%Windowresult each row corresponds to detection result in each
                    timewindow
                end
            end
            break;
        end
    end
end

```

```

        end
    end
    if Windowresult(i2,i) == 1
        break;
    end
end
end
if Windowresult(i2,i) == 1
    for i5 = i2:TimewindowNO
        Windowresult(i5,i) = 1;
    end
    break;
end
end
end
end
end
ValidTestNO = sum(PredictTC(1,:)); %number of valid testing points along the track
%calculate detection results in different time windows
windowresult1 = 1-sum(Windowresult(1,:))/ValidTestNO;
windowresult2 = 1-sum(Windowresult(2,:))/ValidTestNO;
windowresult3 = 1-sum(Windowresult(3,:))/ValidTestNO;
windowresult4 = 1-sum(Windowresult(4,:))/ValidTestNO;
windowresult5 = 1-sum(Windowresult(5,:))/ValidTestNO;
windowresult6 = 1-sum(Windowresult(6,:))/ValidTestNO;
windowresult7 = 1-sum(Windowresult(7,:))/ValidTestNO;

```

LIST OF ABBREVIATIONS

AOA	Angle of Arrival
BS	Base Station
CSR	Convex Subspace Routing
DOA	Direction of Arrival
DPT	Distributive Predictive Tracking
DVC	Directional Virtual Coordinate
DVCR	Directional Virtual Coordinate Routing
ENS	Extreme Node Search
GC	Geographic Coordinate
GLR	Geo-Logical Routing
GDSTR	Greedy Distributed Spanning Tree Routing
GPS	Global Positioning System
GPSR	Greedy Perimeter Stateless Routing
LCR	Logical Coordinate based Routing
MEMS	Microelectromechanical Systems
MWSN	Mobile Wireless Sensor Network
OSTN	Object Tracking Sensor Network
POOT	Prediction-based Optimistic Object Tracking
RF	Radio Frequency
RSSI	Received Signal Strength Indication
SMS	Semi-Markov Smooth (Mobility Model)
SVD	Singular Value Decomposition
TOA	Time of Arrival
TPM	Topology Preserving Map
TC	Topological Coordinate
TCTP	Topological Coordinate based Tracking and Prediction (2D-TCTP and 3D-TCTP)
TN	Test Network
USN	Underwater Sensor Network
VC	Virtual Coordinate
VCS	Virtual Coordinate System
WSN	Wireless Sensor Network