Clueless Nodes to Network-Cognizant Smart Nodes: Achieving Network Awareness in Wireless Sensor Networks

Dulanjalie C. Dhanapala and Anura P. Jayasumana
Department of Electrical and Computer Engineering,
Colorado State University, Fort Collins, CO 80523, USA
{dulanjalie.dhanapala, Anura.Jayasumana}@Colostate.edu

***Abstract*—A novel scheme is presented that allows individual nodes in sensor networks to achieve network/topology-awareness by listening to regular packets associated with applications. Nodes, initially oblivious to network topology and their position within the network, gradually infer information required to evaluate their own Virtual Coordinates (VCs). A Singular Value Decomposition based transformation allows each node to convert the VCs of nodes, gleaned from the source or destination address field of packets, to corresponding Topological Coordinates (TCs). Eventually each node generates a topology map of the network, thus becoming aware of its own location and those of other nodes. Effectiveness of self-learning scheme, in terms of convergence of different stages, and gradual development of network awareness at nodes are illustrated. While many applications of network awareness within nodes can be foreseen, we illustrate how performance of routing can improve dramatically over time as network awareness develops within a node.**

Keywords-Network-awareness, Routing, Sensor Networks, Virtual Coordinates,Topology Preserving maps

I. INTRODUCTION

We envision future sensor networks that continually evolve over time becoming smarter and better at what they do by learning and inferring information about the network and sensed phenomena based on information gleaned from on-going packet transmissions. Upon initial deployment, the nodes would be quite oblivious of their environment, their position in the entire network, sensing phenomena and the nature of the network that they belong to. However, as time evolves the nodes will gain knowledge and awareness of its network as well as its physical environments. While most proposed and existing sensor networking strategies use a dedicated set-up phase to organize the network, we contemplate nodes that do their assigned sensing tasks based on their current capability while listening to on-going activities to develop 'awareness' about its network and environment over time. We use the term "network/topology awareness" to indicate a node's knowledge of the topology of the network including the shape and voids, and its place in that network. State information will be gained gradually, and as nodes attain different levels of network awareness, they will transition to more efficient protocols and strategies, resulting in the network itself becoming more intelligent over time. This paper is a step toward realizing such a vision.

This research is supported in part by NSF Grant CNS-0720889

Recent advances in Wireless Sensor Networks (WSN) point to a vast array of potential applications including target tracking, surveillance, environment and habitat monitoring, subsurface plume tracking [12], and monitoring aquatic environments [14]. Information discovery, data fusion and dissemination play crucial roles in such applications. When nodes are randomly deployed, dropped from a helicopter for example, the only information a node initially knows about the network or has access to, is how many neighbors it has and their IDs. With this information a node starts the mission of sensing, data acquisition, and data dissemination. Content based and addressed based schemes are the two main streams of data dissemination [6]. Former routing schemes are used in locating unknown destinations [5]. Such schemes use information contained in the packet to define and discover the destination set while the latter, address based schemes, use some kind of structural address, physical positions for instant, to describe destination. Rumor routing [5] is an example of Random Routing (RR) /content based protocols that do not need a setup phase. In RR, a "packet" is forwarded to a randomly selected neighbor. This packet can be a query looking for information of a destination or a sink. Traditional address based schemes have a setup phase in which control packets are exchanged/flooded to organize the network. After the setup phase, which usually involves significant overhead, the nodes gain sufficient information for subsequent operation. Even in cases where the network is structured by an initial setup phase, a RR phase is still required to discover the destination or sink which has the required information.

This paper presents and evaluates a comprehensive self-learning scheme that exploits routine RR messages used for information discovery and dissemination, to allow nodes infer knowledge about the network and become aware of the network. Network-awareness is achieved without the need for localization based on error prone analog measurements such as RSSI or time delay. To our knowledge, this is the first such learning strategy. Achieving network awareness also allows the network; transition to more advanced algorithms such as address based routing strategies to enhance performance over time.

Consider a newly deployed, thus unstructured network, which uses RR. The nodes develop a Virtual Coordinate (VC)

specification based on information gleaned from queries and agents of RR. The Virtual Coordinate System (VCS) characterizes each node by a coordinate vector consisting of the shortest hop distances to each of a set of M anchors. Furthermore, each node keeps on learning about the network by making use of the source and destination addresses of the packets that it can hear. These coordinates are then converted to topological coordinates (TCs) using a Singular-Value Decomposition based transformation. This allows the nodes to construct their own map of the network and thus become network aware. TCs also allow a node to identify where it is, e.g., whether it is a boundary or interior node, or where it is with respect to sensed phenomena. Consider plume tracking [12] as an example; a node can now locally identify whether it is on the boundary of the plume, and the network can track which way the plume is moving, etc., without any additional localization cost incurred. The rates at which nodes acquire VCs and then generate topological coordinates are investigated. The effects of network parameters such as number of nodes, number of packets that traversed in the network and Time-To-Live (TTL) of the packets on performance are evaluated using a 2D sensor network.

Reinforcement learning, supervised schemes, neural network approaches, etc., have been used to implement learning in WSNs. But their main disadvantage is the cost penalty paid during the model training period. Moreover, as the destination varies, the model needs to be trained again. In contrast, the proposed scheme lays a foundation for nodes to infer their positions, shape of network boundaries, etc. by moving beyond the traditional ways of learning.

Section II discusses related work. Then Section III explains generation of VCs based on RR, and proposes a learning based strategy for topological coordinate generation from VCs. Section IV evaluates the performance. Section V uses routing as an example to demonstrate improvement in performance achievable with increased topology awareness, while Section V concludes our work.

II. Related Work

Learning techniques used for resolving sensor-networking challenges are briefly reviewed next

An algorithm for distributed classification in sensor networks based on different classification types is presented in [11], which describes the sensed values as a Gaussian Mixture, and uses a machine learning approach for classification. A probabilistic semi-supervised learning approach proposed in [15] reduces the calibration effort and increases the tracking accuracy. This method is based on semi-supervised conditional random fields, which develops the learned model from a set of training data. A scheme for allocation of limited resources such as energy and bandwidth in sensor networks, Self-Organizing Resource Allocation (SORA), is presented in [13]. SORA defines a virtual market in which nodes sell goods, such as sensor readings or data aggregates, in response to costs that are pre-programmed. Nodes take actions to optimize their profit, subject to energy budget constraints. Using reinforcement strategies, nodes adapt their operation over time in response to feedback from payments. An incremental multi-classification support vector machine (SVM) technique for action classification based on real-time multi-video collected by homogeneous sites is proposed in [2]. The technique is based on an adaptation of least square SVM (LS-SVM) formulation. There is an initial offline supervised learning phase followed by online learning phases. A cluster head performs an ensemble of model aggregations based on the sensor inputs. A pattern recognition based event detection scheme is proposed in [4], in which the, individual sensory measurements of sensor nodes are integrated into high-level event patterns to recover the state of the monitored environment.

III. Network-awareness From Random Routing

The proposed scheme for achieving network awareness by individual nodes, based on the packet transmissions that they are able to hear, is presented next. It consists of three phases: (a) VC generation, (b) generation of a transform to infer topological coordinates from VCs, and (c) development of a map of the network within each node by gleaning information about network nodes from on-going packet transmissions. The first stage is aimed at developing a set of VCs to characterize the nodes without flooding it. The RR scheme used during this stage is described next followed by the three phases.

A. Random routing

Many RR protocols have been described in literature [1][5]. In this paper we use rumor routing [5], although any other random scheme will work. In rumor routing, a node selects a random neighbor as the next hop. Two types of randomly routed messages called event agents and queries exist. When a group of nodes experiences an event, each node probabilistically decides to send out a message, informing rest of the nodes in the network. This message is called an "event agent" or simply an "agent". On the other hand, a node looking for specific information of a physical phenomenon sends out a packet called a query, requesting information. Both agents and queries are forwarded to randomly selected neighbors. To prevent a packet from circulating indefinitely, a packet is associated with a Time-To-Live (TTL) timer initialized at the source node with a maximum number of hops (H) the packet is allowed to traverse in the network.

B. Virtual coordinate generation via self learning

Consider a network with N nodes in which a subset of M nodes are designated as anchors. A node that is h_{A_j} hops away from the $A_j{}^{th}$ anchor will have h_{A_j} as the jth ordinate. Thus ith node is characterized by the vector, $P_{(i)} = [h_{iA_1} \dots h_{iA_j} \dots h_{iA_M}]$ of cardinality M. Traditional approach is to use a setup phase, in which each anchor floods the network so that each node can evaluate the shortest distance, in terms of number hops, to each of the anchors. The anchors may be selected randomly, or by an anchor placement mechanism.

In the proposed scheme, a set of nodes is pre-assigned to become anchors. Alternatively, a node can become an anchor

with a predefined probability $p < 0.06$. As the network is unstructured initially, the intended application has to use RR for data dissemination. Thus, the nodes send out agents/queries on random routes as shown in Fig. 1. When an anchor node A_i receives such a message or if it generates one, A_i will append a tuple { A_i ID, $\hat{h}_{A_i}$} containing its self ID and a hop count field to the packet, and forwards it to a randomly selected neighbor. Moreover if a packet containing hop count to A_i, passes through A_j then A_j will append its ID and hop distance to the packet as { A_i ID, $\hat{h}_{A_i}$, A_j ID, $\hat{h}_{A_j}$} to begin generating ordinate with respect to A_j. This is illustrated in Fig. 1.

When a node receives a message with an anchor ID and hop count pair for the first time, it will store the anchor ID and corresponding hop count received incremented by 1 hop. If the node already has a hop count corresponding to the particular anchor, the new value is stored only if the new value is less than the existing value for the hop count. Then the node adds one hop to the hop count(s) in the packet and forwards it to a neighbor. Initially the ordinates estimate by nodes, $[\hat{h}_{iA_1} \ldots \hat{h}_{iA_j} \ldots \hat{h}_{iA_M}]$, are not the shortest path distances to the anchors but as packets disseminate in the network and nodes keep updating the ordinates, ultimately the values converge to the shortest path distances to anchors $[h_{iA_1} \ldots h_{iA_j} \ldots h_{iA_M}]$. How a node figures out whether it has VCs with sufficient accuracy is discussed later.

This is not another flooding scheme, rather it makes use of random routed message to structure the network via VC learning. For instance consider the sample random path shown in Fig. 1 with randomly selected anchors A_1, …, A_7. When A_1 receives the packet it appends the tuple $\{A_1,0\}$ and forwards to a neighbor. When this packet reaches anchor A_4, it stores the distance to anchor A_1, i.e. { A_1,2+1}. Then it updates the distance to A_1 as { A_1,3}, appends { A_4,0} and forwards it to a neighbor per RR. Any of the neighbors that overhear the packet as they were passively listening to the channel, the green nodes in Fig. 1, also update their coordinates but do not forward the packet. This speeds-up the learning process. A node starts using the VCs after it has forwarded or overheard consecutive L_p number of packets without having any updates in its VCs.

C. *Distributed self generation of topological coordinates from virtual coordinates*

Consider a WSN with N nodes, with each node characterized by a vector of VCs, with distance to each of the M anchors ($N >> M$). Let P be the $N \times M$ VC matrix of the network. As explained in [6], if a node has a subset of VCs of nodes (L_N), i.e., a matrix Q containing a subset of rows of P, topological coordinate (TC) of the node N_i can be acquired by using Singular Value Decomposition as follows:

$$Q = USV^T$$

$$[X^T, Y^T]_{(i)} = P_{(i)}[V^{(2)}, V^{(3)}] \quad (1)$$

U and V are unitary matrices of size $L_N \times L_N$ and $M \times M$ respectively where S is a diagonal matrix with singular values as diagonal entries. As explained in [9] if L_N is around 2% of the network nodes if selected uniformly at random.

Once the network nodes have acquired a VCS, or a close approximation thereto, routing is performed using a VC based routing schemes for example as in [6][8]. As a node forward the packets, it can store L_N number of different addresses, appearing in destination and source address fields, to form the Q matrix. Note that the V matrix is not the same at all the nodes. But if the Q matrix is at the node is a good representations of the entire VCS, $V_{(i)}$ of N_i is a rotated version of $V_{(j)}$ of node N_j. Simply, each node is capable of generating a representative TPM of the network, even though that at one node may be different in orientation from that another. This does not cause problems for WSN algorithms such as routing as relative node locations are preserved in each of the maps.

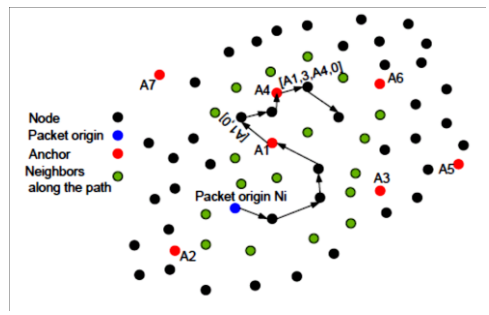

Figure 1: An example network indicating the virtual coordinate propagation in the VC developing stage. Network has 7 anchors.

D. *Network topology map generation within a node*

As a node forwards or overhears packets, it acquires the VCs of their source and destination nodes. The node translates each of these coordinates to the corresponding TCs, and adds those nodes to a topological map that it generates and maintains. With time, as the node acquires VCs of more and more nodes, the map continues to expand, ultimately converging to a map of almost entire network. No explicit transmission cost or energy is spent for topology map generation. Computational complexity of SVD can be significantly reduced, by using eigen-value decomposition instead for $[V^{(2)}, V^{(3)},]_{(i)}$ calculation for 2D networks [6] at node N_i. We can consider full V matrix estimation using eigen-value decomposition as an upper bound on computation complexity, and is given by $(4M^2L_N + 8M^3)$ [9], while that for memory usage is $(2 \times M) + (L_N \times M)$, where L_N is the number of node addresses used for V matrix estimation. Note that we require just 2nd and 3rd columns of V.

IV. Performance Analysis of Network-Aware Sensor Network

The performance of the proposed scheme is evaluated next. MATLAB® 2009b was used for the computations. The circular network with three voids with 496 nodes (see Fig. 2), one of the CSU Sensor-Net benchmarks [7] that has been used for evaluating different WSN algorithms [8],[9],[10], is used to illustrate the effectiveness of the learning scheme.

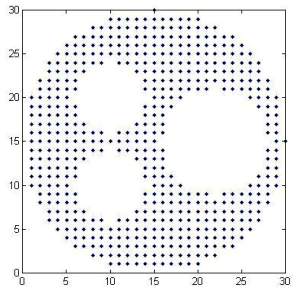

Figure 2: Circular network with 496 nodes. Same shaped networks with 1081 and 2048 nodes are also used in simulations

In addition, we also considered a similar shape with 1081 and 2048 nodes to evaluate the scalability of the scheme. Communication range of a node is unity. Corresponding to a probability 0.02 at which a node decides to become an anchor, 10, 20 and 40 randomly selected nodes served as anchors for the networks with 496, 1081 and 2048 nodes respectively.

The convergence and parameter tuning involved with VCS learning is investigated next, followed by the evolution of TPM in Subsection B.

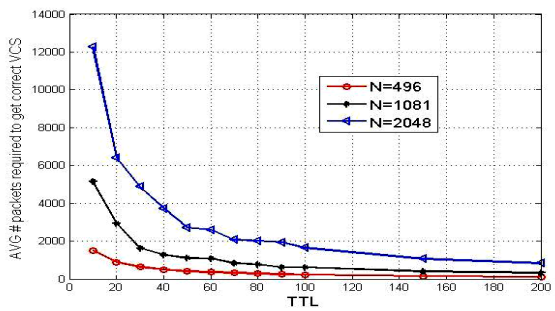

Figure 3: Number of packets required for all the nodes to generate error free VCS vs. TTL (averaged over 10 different configurations). Network has 10, 20 and 40 anchors respectively for N=496 , 1081 and 2048.

A. *Self-learning based VC generation*

Randomly selected source nodes start disseminating packets in the network searching for information. Once such packets pass through anchors, the nodes use them to estimate the VCs as explained above. The time it takes for the nodes to learn the VCs and the accuracy of such VCs are of interest. As an evaluation metric, the error in evolved VCs ($\hat{h}_{iA_j}$) compared to correct VCs (h_{iA_j}), is defined as:

$$E_V = \frac{1}{MN}\sum_{i=1}^{N}\left(\sum_{j=0}^{M}|h_{iA_j} - \hat{h}_{iA_j}|\right) \quad (2)$$

Figure 3 gives the number of packets required to achieve error free VCS, i.e., $E_V = 0$, under different TTLs. It demonstrates the convergence VCs as a result of network wide learning. As the network size increases the TTL required to achieve correct VCS, under a fixed number of rumors, is higher. Based on these results, for the rest of the evaluations the TTL for networks of nodes, 496, 1081 and 2048 are set to 100, 150 and 200 respectively.

The next challenge is how a node can determine on its own whether it has correct VCs, given that the proposed network learning scheme is distributed. When the number of consecutive packets a node has heard but did not cause any updates of its estimated VCs exceeds a threshold L_p, the node assumes that it has the correct VCs. In order to decide a value for L_p, variation of L_p vs. the error in generated VCs is evaluated in Fig. 4. It can be seen from Fig. 4 that there exists a L_p where error in VCS (E_V) goes to zero; but the drawback is the diminishing return as the number of packet gets higher. When a node has determined that it has accurate VCs, it is considered to be in the VC mode. Now the node can move beyond RR to use more organized VC routing schemes [6][8]. How the percentage of nodes in the VC mode increases with the number of packets transmitted in the network is illustrated in Fig. 5 for networks of sizes 496, 1081 and 2048 and $L_p =$ 10, 20 and 30 respectively. For $N = 496$, for example, when L_p changes from 10 to 30, number of packets required for all the nodes to get in to VC mode is more or less the same. Nevertheless, from Fig. 4 for larger L_p, error in VCS gets smaller. The number of packets to achieve 100% VC mode nodes for a network of size 2048 is approximately thrice that for a 496-node network, and twice that for a 1081-node network. L_p is set to 30 in the rest of simulations as at L_p of 30 the error of the generated VCS (from Fig. 4), is 0.0001, 0.002 and 0.004 respectively in 496, 1081 and 2048- node networks.

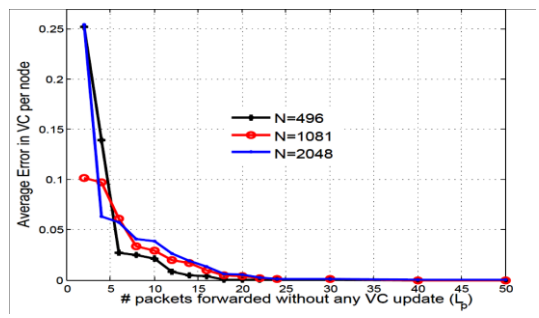

Figure 4: Error in VCS vs. number of packets forwarded by each node without any VC updates L_p, averaged over 10 anchor placements.

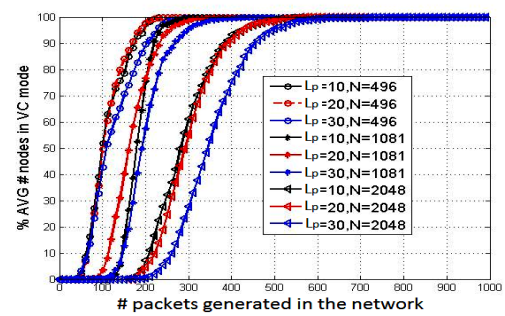

Figure 5: Percentage of nodes in VC mode vs. number of packet that traversed the network, averaged over 10 random anchor placements. TTL is 100.

B. *Evolution of topology awareness*

When a node decides that its VCs have converged, it enters the VC mode. A node in this mode identifies itself using its VCs, and thus packets generated by this node contain its VCs. Each node that hears the packet transmission thus can store it locally to create the matrix Q required for topology coordinate calculation. A node collects L_N unique virtual coordinates from the packets that it forwards or hears. Results presented below use $L_N = 10$ for $N = 496$ network. Q matrix can now be evaluated and used to convert VC specification of any

node including its own to the corresponding topology coordinates.

Every node continues listening to on-going packet transmissions to gather VCs of nodes from source and destination address fields. Then it will calculate the corresponding topology coordinates of collected nodes and add them to the topology map of the network. As time evolves, the node finds out about more and more nodes, and the topology map completes. Fig. 6 (a)-(d) illustrates the evolving view of the topology map at one of the nodes in the network, which is representative of what happens at each node. Packet origin and destination is selected uniformly at random. Fig. 6 (e)-(h) illustrate the TPM at the same node at the end of 15000 packet disseminations as the number of anchors vary from 5 to 20.

Initially no node had any idea of its position, the neighbors' positions or the network topology. Only information it knew about the network was how many neighbors it had and the neighbor IDs. But with the proposed scheme, nodes now gain awareness of their position in the overall network, the shape of the network and its topology. All this has been achieved without a dedicated setup phase requiring its own packet exchange, but by listening to on-going transmissions and inferring information about the nodes and the network.

V. Topology Awareness – Benefits And Applications

Topology awareness provides nodes the capability of selecting appropriate algorithms, for self-organization, routing, topology management, boundary detection, etc., based on the level of the awareness it has. Routing is considered below due to the crucial role it plays in sensor network applications. As a node gains network awareness, it can change the routing scheme, e.g., starting with Random Routing when the nodes have no awareness at all, to coordinate based routing as awareness is gained. In fact, once a node becomes aware of a significant part of the topology map, it is possible for it to map the path to the destination, thus being able to achieve almost full routability. Even partial topology maps, e.g., when a node does not have information about all the other nodes, the technique can still be beneficial. For instance, RR schemes can be enhanced to improve the probability of discovering destination by dispatching queries and agents more intelligently. They can be dispatched for example, to dense or central regions (market places) of a network. Alternatively, they may be dispatched in straight lines or some geometric pattern to maximize the rendezvous probability.

Next we discusses how routing performance of an unstructured network improves as nodes move from RR to VCS based routing first, and then to TPM based routing. In the proposed self evolving network, after developing VCS with error E_V, it will start using VCs for routing till topological coordinates are generated. Any VC based routing proposed in literature is possible to use. When a packet is routed from N_i to N_d, nodes which are in VCS mode will use VCS based routing scheme while nodes in Topological mode will use a routing scheme called Geo-Logical routing (GLR) [10].

TC mode that uses topology based coordinates for distance evaluation; VC mode, which uses VCs and a VC based distance, and AM (Anchor Mode) which routes packets toward the anchor A_c that is closest to the destination. The source node initiates routing in TC mode. Packet continues to get routed in this mode until it reaches a local minima in the topology space, at which time the mode changes to VC mode. If it encounters a local minima in this mode, the packet is

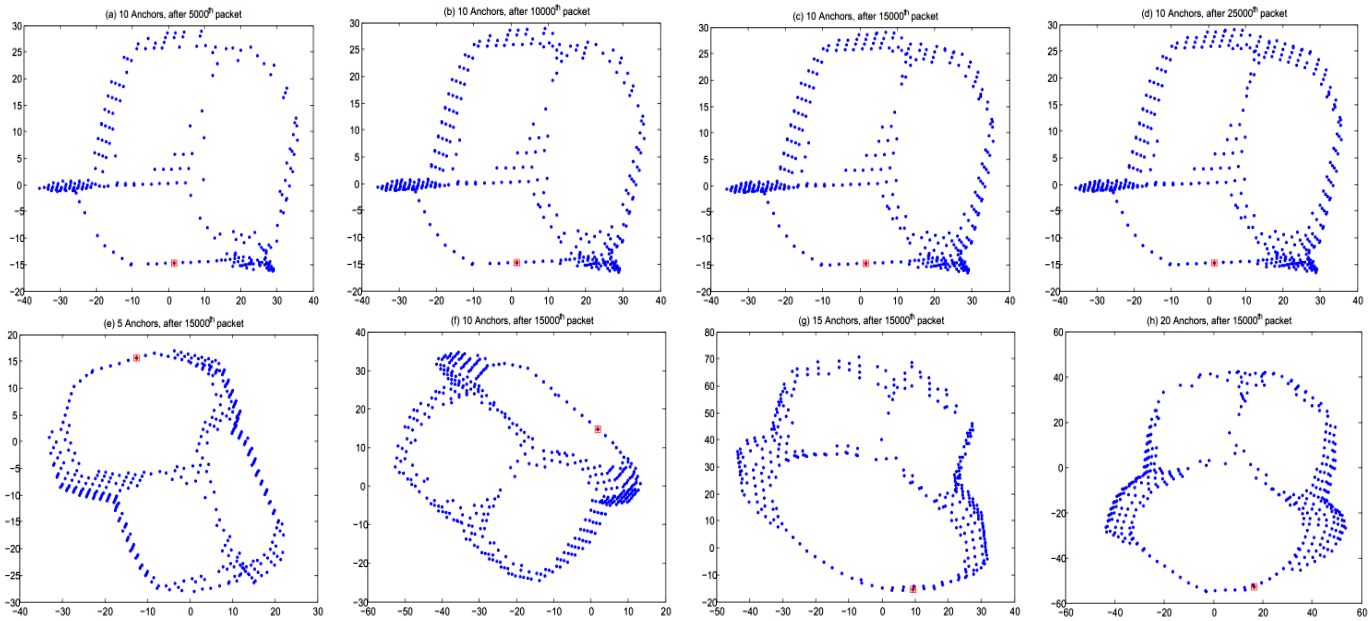

Figure 6: (a)-(d) TPM growth at a sample node (identified by color red) when the number of anchors is 10. (e)-(f) TPM after 15000 packets dessiminate in the network when the numer of anchors vary as 5,10,15 and 20 respectively. TTL of a packet is 100.

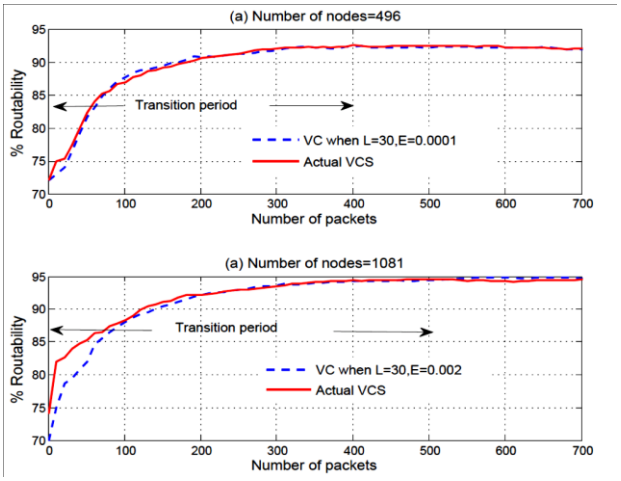

Figure 7: Variation of % number of nodes in the topological domain as the number of packets generated in the network varies.

routed using AM mode. Once the anchor is reached, it goes switches to the VC mode and procedure repeats until the packet has reached the destination or the TTL has expired.

Figure 8 illustrates the change in routability as the nodes achieve VCs first, and then the topology awareness. % Routability of the network is defined as,

$$\frac{\text{Total \# of packet that reached the destination}}{\text{Total number of packet generated}} \quad (3)$$

Routability performance in two network sizes N=496 and 1081 are examined in Fig 7 (a) and (b) respectively. At the beginning, i.e. when the number of packets disseminated in the network is zero (See Fig. 7), the entire network is in the VC mode. Dashed curve is corresponding to VCS which is generated using the proposed learning scheme thus E_V is 0.0001 and 0.002 in Fig 7(a) and (b). Performance with an error free VCS is shown in continuous red curves. Initially, entire network's routing is purely based on VC based greedy forwarding. As explained earlier, when the packets disseminate in the network nodes start collecting unique source and destination VCs and generate TCs locally. Those nodes that have TCs start using GLR for next node selection while rest of the nodes use VC based greedy selection for next node selection. As it can be seen, during the transition period in Fig 8, fraction of the nodes in the network uses GLR while the rest rely on VC based greedy selection. End of the transition period, entire network start using GLR (see Fig. 8). It can be clearly seen in Fig. 8 that, with topological coordinates and VCs based hybrid routing schemes provide about 20% improvement is routability in the network. Moreover, as the results in Fig 7 indicate, network may bare up to a certain % of error in VCS without affecting its performance.

VI. Conclusion and Future Work

A novel scheme by which nodes can achieve network awareness using routine packet transmissions was presented. The scheme does not need node localization using unreliable analog measurements or a costly training phase. The nodes simply infer information for a virtual coordinate system, and generate a topology map of the network on their own via an intelligent distributed algorithm.

The scheme as described requires self-elected or pre-identified anchor nodes to insert a field to each packet. A simple modification that allows only one anchor to modify a packet would keep packet length constant, to save energy for example, yet will increase the learning period. We also demonstrated that a network is able to operate in VC mode even with a small error in VCs.

Many existing applications can benefit from network awareness, as illustrated using routing as an example. When a near complete topology map is learned by each node, it is possible to identify source to destination path completely and thus achieve 100% routability. One can also envision many novel applications not only in ubiquitous sensor networks but also in emerging nanonetworks that will become possible with network awareness, especially due to the fact that it can now be achieved with very little cost.

References

[1] J.N. Al-Karaki, and A.E. Kamal, "Routing techniques in wireless sensor networks: a survey," IEEE Wireless Communications, Dec. 2004, pp. 6-28.

[2] M. Awad, X. Jiang, and Y. Motai, "Incremental support vector machine framework for visual sensor networks," EURASIP Journal on Advances in Signal Processing ,2007.

[3] J. Bachrach, and C. Taylor, "Localization in sensor networks," Ch. 9, Handbook of Sensor Networks. Stojmenovic (Editor), John Wiley 2005.

[4] M. Baqer, and A.I. Khan, "Energy-efficient pattern recognition approach for wireless sensor networks," 3rd Int. Conf. on Intelligent Sensors, Sensor Networks and Information, Dec. 2007., pp. 509 – 514.

[5] D. Braginsky, and D. Estrin, "Roumor routing algorithm for sensor networks," Proc. 1st Workshop on Sensor Networks and Applications (WSNA), 2002.

[6] Q. Cao and T. Abdelzaher, "Scalable logical coordinates framework for routing in wireless sensor networks," ACM Transactions on Sensor Networks, Nov. 2006, pp. 557-593.

[7] CSU Sensor-Net Benchmarks, Available: http://www.cnrl.colostate.edu/Projects/VCS/

[8] D. C. Dhanapala and A. P. Jayasumana, "CSR: Convex subspace routing protocol for WSNs," Proc. 34th IEEE Conf. on Local Computer Networks, Oct. 2009.

[9] 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, Oct. 2010.

[10] D.C. Dhanapala and A.P. Jayasumana, "Geo-logical routing in wireless sensor networks," Proc. 8th IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON), June 2011.

[11] I. Eyal, I. Keidar, and R. Rom, "Distributed data classification in sensor networks," Proc. 29th ACM SIGACT-SIGOPS , Jul 2010, pp. 151-160.

[12] Q. Han, A. P. Jayasumana, T. Illangasekare & T. Sakaki, "A Wireless sensor network based closed-loop system for subsurface contaminant plume monitoring," Proc. 22nd IEEE Int. Symp. on Parallel and Distributed Processing, April, 2008.

[13] G. Mainland, D.C. Parkes, and M. Welsh, "Decentralized, adaptive resource allocation for sensor networks," Proc. 2nd Symp. on Networked Systems Design & Implementation, 2005, pp. 315-328.

[14] G.V. Merret, and Y.K. Tan, "Wireless sensor networks: application centric design," InTech. December, 2010.

[15] R. Pan, J. Zhao, V.W. Zheng, J..J. Pan, D. Shen, S.J. Pan, and Q. Yang, "Domain-constrained semi-supervised mining of tracking models in sensor networks," Proc. 13th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, 2007, pp. 1023-1027.