Finding Link Topology of Large Scale Networks from Anchored Hop Count Reports

Taha Bouchoucha, Chen-Nee Chuah, Zhi Ding
Deptment of Electrical and Computer Engineering
University of California, Davis, California, 95616
Email:{tbouchoucha,chuah,zding}@ucdavis.edu

Abstract—[1] **Learning network topology from partial knowledge of its connectivity is an important objective in practical scenarios of communication networks and social-media networks. Representing such networks as connected graphs, exploring and recovering connectivity information between network nodes can help visualize the network topology and improve network utility. This work considers the use of simple hop distance measurement obtained from a fraction of anchor/source nodes to reconstruct the node connectivity relationship for large scale networks of unknown connection topology. Our proposed approach consists of two steps. We first develop a tree-based search strategy to determine constraints on unknown network edges based on the hop count measurements. We then derive the logical distance between nodes based on principal component analysis (PCA) of the measurement matrix and propose a binary hypothesis test for each unknown edge. The proposed algorithm can effectively improve both the accuracy of connectivity detection and the successful delivery rate in data routing applications.**

Index Terms—**Graph, Connectivity, hop distance, PCA**

I. Introduction

Information characterizing network topology for purposes such as routing and node localization can often be categorized into two types depending on practical constraints and applications: geographical coordinate systems (GCS) where node locations are characterized by physical coordinates based on various signal or delay measurements, and virtual coordinate systems (VCS) [1], [2], [3], [4] where the network topology is characterized by the binary inter-node connectivity information (also known as adjacency matrix). Often, virtual node coordinates can be determined from simple measurements such as hop counts from each network node with respect to a set of source (anchor) nodes. Unlike GCS, VCS is less sensitive to path estimation errors since hop distances (counts) are easier to measure according to a simple controlled flooding [5], and hop measurement is much more reliable than physical distance measurement that critically depends on received signal quality, channel disturbances such as noise, multipath interference, co-channel interference, fading or shadowing, and clock synchronization.

Nevertheless, network topology in terms of VCS also poses several unique challenges in practice. In particular, it is always desirable to utilize fewer anchor nodes and measurements for practical purposes. Hence, the selection with respect to the number and the placement of anchor nodes [6] within a network is a critically important open problem. On one hand, if the number of anchors is small or their placement is not diverse enough, the reconstruction of network topology will degrade, thereby leading to poor routing performance. Also, by deploying only a small number of anchor nodes, temporary malfunction or outage of certain nodes can lead to critical loss of measurement and performance. On the other hand, using many anchors and measurements will substantially increase the cost, the complexity, and the network traffic load in practical applications.

Several published works exist on the use of VCS for applications such as network topology preservation and routing. In [1] and [7], virtual coordinates are analyzed through principal component analysis (PCA) in order to produce a Cartesian coordinate map that is homomorphic to the network physical configuration while preserving information about the physical layout and the voids of the network. The authors suggested that the second and the third principal components of the hop count measurement matrix appear to provide a 2-dimonsional (2-D) coordinates of the topology preserving map. Examining the resulting visual 2-D graphs, various examples indicate this approach appears to be topology preserving. Nevertheless, such visually based approach does not provide quantitative metrics regarding the successful recovery of the node connectivity matrix of the network. A similar work [2] also aims to construct a virtual topology instead of trying to approximate the physical coordinates. The author of [2] proposed to exploit hop counts from three selected anchors to define zones of nodes assigned with the same (similar) coordinates without assigning virtual coordinates to each node. Applying more quantitative outputs instead of relying on visual 2-D display of network maps, the authors of [8] suggested a PCA-based dimension reduction in VCS to perform routing in networks with landmark nodes (i.e., anchor nodes) which provide hop count measurements with respect to the remaining network nodes. They normalized the first two principal components of the hop count measurement matrix, similarly to the reduced dimension virtual coordinates [7], in order to construct a routing algorithm [8]. Several other routing algorithms also rely on VCS [3], [9], [10]. The protocol developed in [3] lets each node forward its packets to the neighbor node nearest to

[1]This material is based on work supported by the National Science Foundation under grants CNS1443870 and CNS1702752

the packet destination node in VCS. Loop and void avoidance techniques can further increase the percentage of successful delivery. One practical challenge lies in local minima during routing. The idea of [9] is to dynamically select anchors such that the distance function between source and destination is convex. This approach favors network edge (boundary) anchors over interior ones. Another connectivity-based routing protocol proposed in [10] to apply a tree based recovery method in order to avoid local minimum solutions. We note that all previous routing methods do not provide any global information about the adjacency matrix of the network.

Other recent works on network topology inference utilized various measurements different from hop counts. In [11], graph signal processing is used to identify networks topology by exploiting the spectral characteristics of graphs as well as graph filter dynamics in order to model network diffusion processes. In [12], nonlinear structural equations are proposed to model signal propagation and diffusion processes in sparse networks in order to infer unknown network topologies. The authors of [13] exploited Traceroute operations to probe paths within the network to infer the network topology.

In this work, our objective is to utilize the hop-count measurement more efficiently for network topology inference. Instead of utilizing visual graphs and 2-D homomorphic topology maps [7] [1] that do not provide any information about the adjacency matrix, we use the anchor hop measurements to reconstruct the adjacency matrix of the networked nodes. The reconstructed adjacency matrix can provide a full connectivity description of the network topology that reveals the relationship between nodes and facilitates packet routing. More specifically, our new approach shall adopt the VCS and improve the utility of PCA results to infer the unknown topology of connected networks based only on hop count measurements at a subset of anchor nodes. We first reduce the number of unknown edges based on pre-processing the hop count matrix before applying PCA to generate the virtual nodal coordinates. The proposed pre-processing is a low complexity tree-based search method that establishes the maximum number of node connectivities that can be directly determined from the available hop count measurements. These detected edges are then analyzed to determine a threshold ϵ that defines a binary hypothesis test for each unknown edge between two nodes based on the logical distance. Our novel approach presents two key advantages: (a) reduction of problem size by lowering the number of unknown edges for detection; (b) determination of an empirically derived threshold ϵ for the binary hypothesis test.

The rest of the paper is organized as follows. In Section II, we describe the system model and present the basics PCA for analyzing the hop count matrix. Next, we present our proposed topology inference method in Section III. We discuss traffic routing based on the recovered topology in Section IV. Finally, we provide some test results in Section V before conclusions.

Notations: Lower-case letters, bold lower-case, and bold upper-case letters, respectively, designate scalars, vectors, and matrices. If $\mathbf{A}$ is a matrix, then $\mathbf{A}^T$ denotes the transpose of $\mathbf{A}$. $A(i,j)$ denotes the entry in the i^{th} row and j^{th} column of matrix $\mathbf{A}$.

II. VIRTUAL DISTANCE AND TOPOLOGY INFERENCE

We consider a wireless network consisting of N nodes denoted as $\{n_1, n_2, \dots n_N\}$ which form a connected graph and from which we pick up a subset of M anchors denoted as $\{A_1, A_2, \dots, A_M\}$ where M typically is much smaller than N. To explore the unknown topology of the network, we collect the hop distance of each node n_i to the M anchor nodes. The results form an $N \times M$ hop count matrix $\mathbf{P}$ such that $P(i,j) = h_{n_i,A_j}$ where h_{n_i,A_j} is the hop distance between node n_i and anchor A_j. h_{n_i,A_j} specifies the minimum number of hops through the shortest path linking node n_i to anchor A_j. Each node n_i generates a vector $\mathbf{p}_i$ of size M which contains the hop distances from node n_i to the set of M anchors, given by

$$\mathbf{p}_i = [h_{n_i,A_1}, h_{n_i,A_2}, \dots, h_{n_i,A_M}]. \quad (1)$$

$\mathbf{p}_i$ represents the i^{th} row of the hop count matrix $\mathbf{P}$ which can be written as

$$\mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_N]^T. \quad (2)$$

The redundancy of the virtual coordinate vectors could be reduced from M to k by performing PCA and taking the k first most significant principal components. The PCA-based method starts with a singular value decomposition (SVD) of the hop count matrix $\mathbf{P}$ as follows

$$\mathbf{P} = \mathbf{U}\mathbf{S}\mathbf{V}^T = \mathbf{Q}\mathbf{V}^T, \quad (3)$$

where $\mathbf{S} = \text{Diag}(s_1, s_2, \dots, s_M)$ is a diagonal matrix containing the singular values of matrix $\mathbf{P}$ sorted in a decreasing order and matrix $\mathbf{Q}$ is given by

$$\mathbf{Q} = \mathbf{P}\mathbf{V} = [\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_M], \quad (4)$$

where $\mathbf{q}_i$ is the i^{th} principal component of the matrix $\mathbf{P}$. The final step in the PCA is to keep from the matrix $\mathbf{Q}$ only the k first most significant components which gives

$$\mathbf{Q}^k = [\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_k] = [\mathbf{p}_1^k, \mathbf{p}_2^k, \dots, \mathbf{p}_N^k]^T, \quad (5)$$

where the i^{th} row $\mathbf{p}_i^k$ of the matrix $\mathbf{Q}^k$ is the vector of virtual coordinates of the node n_i using only k principal components. $\mathbf{Q}^k$ can be seen as the projection of the hop count matrix $\mathbf{P}$ into the space spanned by its first k most significant eigenvectors.

The Euclidean distance between the virtual coordinates of the nodes obtained from the PCA is then used to estimate the logical distance between each pair of nodes (n_i, n_j) as follows

$$d(n_i, n_j) = \|\mathbf{p}_i^k - \mathbf{p}_j^k\|_2. \quad (6)$$

Notice that by over reducing the dimensions of $\mathbf{P}$ (for visualizations purposes), the authors in [7] degraded the performance of the logical distance because they neglected important singular values. Curiously enough, their method provided a homomorphic map that has a similar shape to the original network.

In routing applications [3], [4], [9], this logical distance d is directly used to identify the neighbor nearest to a given destination node for the next link in traffic routing. We propose a new use of the distance d to estimate the connectivity between network nodes.

Let us examine an exemplary distribution of the logical distances between each possible pair of nodes of a randomly generated network composed of 400 nodes in Fig. 1. From this example, we can clearly see the distinction of distance distributions between connected nodes (denoted by $D_1 = \{d(n_i, n_j);\ E(i,j) = 1\}$) and non connected nodes (denoted by $D_0 = \{d(n_i, n_j);\ E(i,j) = 0\}$). This observation motivates our design of a binary hypothesis test based on the logical distance $d(n_i, n_j)$ between nodes. Thus, we propose PCA binary hypothesis testing (PCA-BHT) that compares the logical distance against a threshold ϵ to decide whether the nodes n_i and n_j are connected. For the symmetric $N \times N$ adjacency matrix $\mathbf{E}$ that we want to recover, our hypothesis testing should be

$$E(i,j) = \begin{cases} 1, & \text{if } d(n_i, n_j) \leq \epsilon \\ 0, & \text{if } d(n_i, n_j) > \epsilon \end{cases} \tag{7}$$

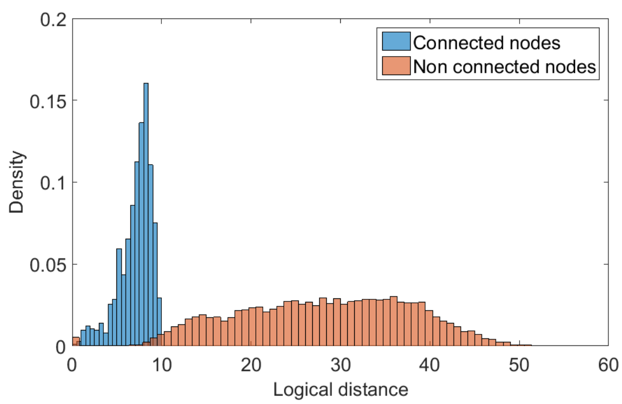

Fig. 1. Distribution of logical distance

III. Proposed network topology inference method

A. Tree-based search method using hop count measurements

In this section, we aim to extract information about the connectivity between the networked nodes using the hop count matrix $\mathbf{P}$ in order to formulate constraints on the values of the unknown connectivity relationships. The proposed method is applicable to any connected graph and it is based on generation of a tree for each anchor node. We begin by assigning a given anchor node A_m to the root of a tree T_m. Next, based on the hop measurements provided by anchor A_m we construct T_m by placing all the nodes which are k hops away from anchor node A_m in the k^{th} layer of the tree T_m which is denoted as $L_{m,k}$ i.e. $L_{m,k} = \{n_i;\ h_{n_i,A_m} = k\}$. Finally, we draw the edges between the placed nodes by respecting the rules stated in Algorithm 1 which utilizes the following constraints to specify the connectivity between nodes n_i and n_j:

- n_i and n_j are disconnected if their hop count, with respect to a same anchor A_m, differs by more than one hop.
- n_i and n_j are connected if their hop count, with respect to a same anchor A_m, differs by exactly one hop say for example $h_{n_j,A_m} = h_{n_i,A_m} + 1$ and the cardinality of $L_{m,h_{n_i,A_m}}$ is equal to 1 i.e. contains only one node.

The output of Algorithm 1 is a set of constraints on the connectivity information that can be seen as an incomplete adjacency matrix $\mathbf{E}'$ where $E'(i,j)$ is equal to '1' if n_i and n_j are connected, '0' if not and '?' if the connectivity cannot be determined given the set of anchor nodes.

Data: $\mathbf{P}$
Result: $\mathbf{E}'$
for *each anchor* A_m **do**
    Generate a tree T_m by assigning the node A_m to the root and the remaining nodes to lower layers based on their hop distance to A_m ;
    for *each pair of nodes* n_i *and* n_j **do**
        Denote $L_{m,h_{n_i,A_m}}$ and $L_{m,h_{n_j,A_m}}$ the respective layers of n_i and n_j ;
        if $|h_{n_i,A_m} - h_{n_j,A_m}| > 1$ **then**
            $E'(i,j) = 0$
        else if $|h_{n_i,A_m} - h_{n_j,A_m}| = 1$ *&* *Card*($L_{m,min\{h_{n_i,A_m}, h_{n_j,A_m}\}}$) $= 1$ **then**
            $E'(i,j) = 1$
        else
            $E'(i,j) =$ '?'
        end
    end
end
Merge the obtained adjacency matrix from each anchor A_m;

Algorithm 1: Proposed tree based search algorithm

As an illustrative example, we consider a simple network in Fig. 2 with one anchor (node n_1) and the generated hop count matrix $\mathbf{P}$ corresponding to that anchor. The tree generated for anchor n_1 is shown in Fig. 3 where

- solid edge between n_i and n_j means $E'(i,j) = 1$
- absence of edge between n_i and n_j means $E'(i,j) = 0$
- dotted edge between n_i and n_j means unknown $E'(i,j)$

Using these constraints, we obtain the incomplete adjacency matrix in Fig. 3. It is clear that by using the measurements from multiple anchors at the same time we can reduce the number of unknown edges furthermore. Therefore, the obtained incomplete adjacency matrix $\mathbf{E}'$ represents a set of constraints on the connectivity relationships between the networked nodes. Next, we will exploit the obtained constraints in order to improve the PCA-BHT and provide the complete adjacency matrix $\mathbf{E}$ and thus recover the topology of the network.

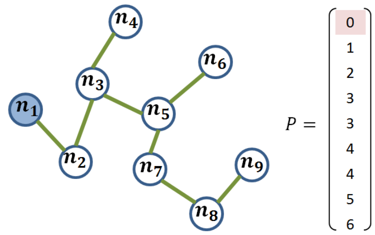

Fig. 2. Graph representing anchored network node and anchor measurements

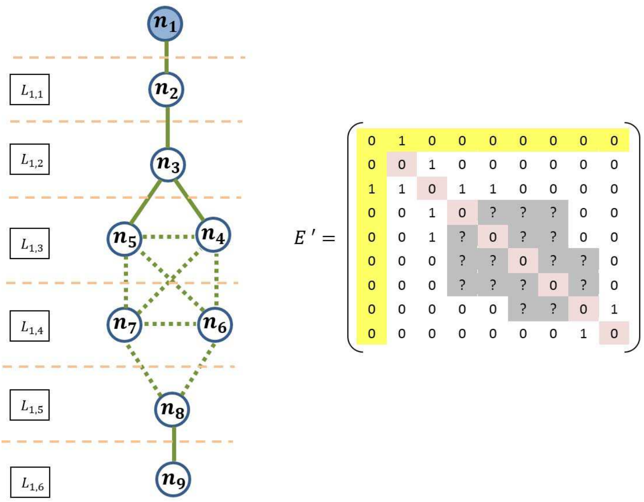

Fig. 3. Tree generation and incomplete adjacency matrix

B. Modified PCA-based binary hypothesis testing (MPCA-BHT)

We show hereafter that the pre-processing step, detailed in Section III-A, can be used to achieve two goals. First, we reduce the number of unknowns for PCA-BHT. Second, we provide a reasonable value of ϵ to use in (7) in order to recover the complete network topology.

We showed in the previous section that the simple information of hop measurements provided by a set of anchor nodes contains some constraints about the edges that we can extract using Algorithm 1. However, the output of Algorithm 1 is an incomplete adjacency matrix since it may contain unknown entries. An unknown entry in the (i, j) position of the incomplete adjacency matrix $\mathbf{E}'$ means that Algorithm 1 is not able to decide whether node n_i and node n_j are connected. To remove this ambiguity, we propose to use the PCA-based method presented in Section II to generate virtual coordinates of the networked nodes then determine the connectivity relationships by applying the binary hypothesis test (7).

We start by reducing the number of unknowns involved in the PCA by extracting fully determined nodes from the incomplete adjacency matrix $\mathbf{E}'$. We define a node n_i to be fully determined if the connectivity relationships between n_i and the remaining networked nodes are known. In other words, n_i is fully determined if the i^{th} row and i^{th} column of the incomplete adjacency matrix $\mathbf{E}'$ do not contain any unknown entries. For example, Fig. 4 shows the incomplete adjacency matrix of the network used in the example of Fig. 2. We notice that nodes $\{n_1, n_2, n_3, n_9\}$ (green diagonal elements) are fully determined since their connectivity with the remaining nodes of the network is known. However, nodes $\{n_4, n_5, n_6, n_7, n_8\}$ (red diagonal elements) are clearly not fully determined.

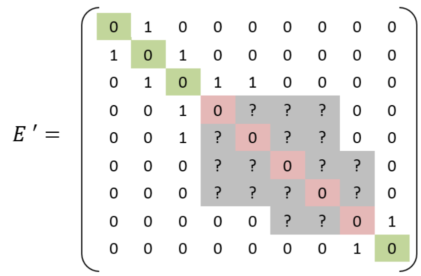

Fig. 4. Example of incomplete adjacency matrix

Since the fully determined nodes have known adjacency relationship with all the nodes in the network, it is no longer necessary to include them as unknowns in the PCA. Suppose we have a network of N nodes and M anchors. If we obtain N' fully determined nodes from Algorithm 1, then the hop count matrix $\mathbf{P}$ used in PCA will be of dimension $(N - N') \times M$ instead of $N \times M$. Consequently, this will further reduce the complexity of PCA and PCA-BHT.

The second advantage of the pre-processing step is to provide a value for the threshold ϵ which is used in the binary hypothesis test (7). After applying Algorithm 1 and obtaining the incomplete adjacency matrix $\mathbf{E}'$, we can define D_ϵ as the set of pair of nodes that are whether connected or have unknown connectivity relationship

$$D_\epsilon = \{(i, j); E'(i, j) = 1 \text{ or } E'(i, j) = \text{'?'}\}. \quad (8)$$

Next we define ϵ as

$$\epsilon = \max_{(i,j) \in D_\epsilon} \{d(n_i, n_j)\}. \quad (9)$$

Note that the number of connected pairs of nodes that are obtained from the incomplete adjacency matrix $\mathbf{E}'$ could not be enough to estimate the value of ϵ. Hence, it is reasonable to add the unknown edges to the set D_ϵ.

IV. Traffic Routing in the Network

One of the most important applications of network topology inference is to execute traffic routing operations optimally. There exist a few routing strategies that were previously developed for connected networks. We distinguish between algorithms that are based on simple hop counts [3] versus algorithms that are based on known geographical coordinates [14]. We consider in this section two existing methods known as the Logical coordinate routing (LCR) [3] and Greedy Perimeter Stateless Routing (GPSR) [14] for benchmark comparison. LCR method uses the hop count measurements provided by a set of anchors. LCR simply forwards packets to the neighbor that is closest to their destination nodes in terms of logical distance. The shortcomings of LCR include the risk of infinite loop and local minima due to the ambiguity of virtual coordinates. On the other hand, GPSR is a more efficient routing algorithm but it requires accurate geographical node locations. Such requirement is costly and less practical in networks with limited resource. Nevertheless, because of its accuracy, we include the GPSR method in our performance comparisons as a benchmark.

The proposed network topology inference can be used to perform routing in networks. In fact, the adjacency matrix $\mathbf{E}$ obtained from the MPCA-BHT can be used to find the shortest path for traffic routing from a source node n_s to a destination node n_d. When the recovered adjacency matrix is obtained without edge error the shortest path is guaranteed and can be found by using a simple algorithm such as Dijkstra. Hence, the only source of delivery failure in our case can be attributed to an error in the recovered adjacency matrix. Such routing failure becomes less likely as the number of anchor increases, to be seen in the next numerical simulation section.

Regarding communication overhead, we note that both LCR and the proposed MPCA-BHT must account for the cost of communication associated with obtaining the hop count matrix $\mathbf{P}$ although they differ in ways of utilzing the hop count information. LCR iteratively explores the neighbors until a packet reaches its destination which may lead to not only a suboptimal path and a possibly infinite loop, but also additional communication overhead arising from the local neighbor exploration step. On the other hand, the proposed MPCA-BHT method does not require such additional communication overhead because the obtained adjacency matrix provides a global information with respect to the topology of the network. Hence, it allows us to apply Dijkstra's shortest path algorithm on the obtained adjacency matrix with a computational complexity of order $O(M \log(N))$.

V. Simulations

We now test the efficiency of our proposed MPCA-BHT to recover the network topology based only on knowledge of hop count measurements between the networked nodes and a set of anchor nodes. Consider a wireless network. We randomly deploy N nodes in a coverage area. Edges between the nodes are generated based on the communication range R of the nodes. Thus, for any node located at the position (x, y) in the 2-dimensional plane, we allow edges to exist from that node to all other nodes located within a radius R. This leads to a network with random unknown topology.

First, we run the proposed tree-based search (Algorithm 1) to determine the incomplete adjacency matrix. Second, we test the PCA-BHT method given in Section II. Finally, we apply the MPCA-BHT method developed in Section III-B. For each method, we calculate and plot the percentage of edge reconstruction error as a function of the number of anchors and the number of nodes in the network. We average over 500 Monte Carlo runs in all the figures otherwise stated. The test results are given in Figs. 5 and 6. It is worth noting that an edge reconstruction error in the PCA-based methods is a flipped 1 or 0 in the reconstructed adjacency matrix of the network. However, in the proposed tree based search (Algorithm 1), because it is not possible to commit an error by flipping a 0 or 1 in the adjacency matrix, we only show the percentage of unknown edges rather than the percentage of edge reconstruction error.

The simulation results show that the tree-based search alone is not enough to recover the full connectivity information of the network as there still is a high percentage of unknown edges especially for low number of anchors. However, after applying our tree search as preprocessing to estimate the suitable value of threshold ϵ before using the PCA-BHT method, the error percentage is substantially reduced. Clearly the MPCA-BHT algorithm outperforms the original PCA-BHT method. In Fig. 5, we notice that the performance of connectivity recovery of every tested method improves as the number of anchors grows. In fact, by increasing anchors, we can form more constraints on the connectivity relationships using Algorithm 1. Thus, we have a better estimate of the threshold ϵ, thereby enabling us to utilize the PCA-based method more efficiently. Fig. 6, shows that the performance of connectivity recovery degrades when the total number of networked nodes N becomes large, since it results in more unknowns.

For traffic routing applications, we generate random networks with random anchor locations. We also randomly choose the starting and destination nodes for traffic routing. Applying our proposed topology inference algorithm to routing (Section IV), Fig. 7 shows the resulting packet delivery rates (percentage of successfully delivered packets) of different routing methods. Our proposed method outperforms LCR and approaches the benchmark GPSR performance as the number of anchors increases. Higher number of anchors allows more efficient recovery of the network edges, as shown in Fig. 5. Thus the shortest path between any two nodes can be found with higher probability. In Fig. 8, we are interested in examining the routing path length. We let ρ be the ratio between the routing path length and the shortest path length given successful packet delivery. As expected, given successful delivery, our proposed method always succeeds in finding the shortest path ($\rho = 1$). However, LCR often results in longer routing paths ($\rho \geq 1$) and therefore requires more resource usage.

VI. Conclusion

Our investigation studies the recovery of unknown network connectivity without prior topology information. Our specific goal is to recover the adjacency matrix of networked nodes by adopting a very simple measurement of hop distance between network nodes and designated anchor nodes. We developed a tree-based search algorithm to establish certain connectivity constraints between network nodes. Applying principal component analysis, we formulated each unknown edge decision as a binary hypothesis problem based on logical euclidean

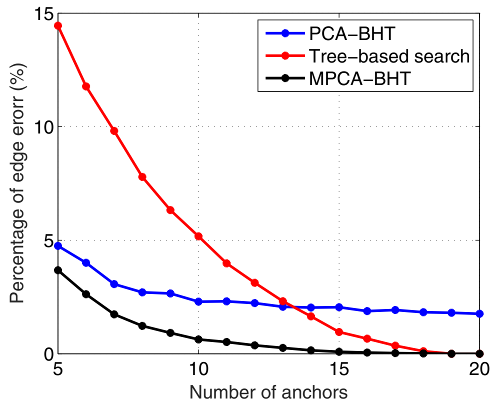

Fig. 5. Effect of number of anchors M on the connectivity recovering, N=20, Number of PC = 5, R = 30, Area: 100×100

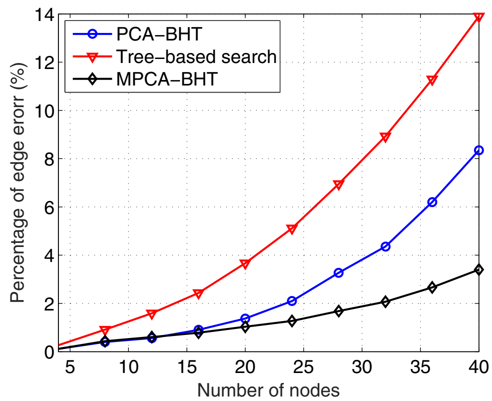

Fig. 6. Effect of number of networked nodes N on the connectivity recovering, M = N/4, R = 30, Area: 100×100

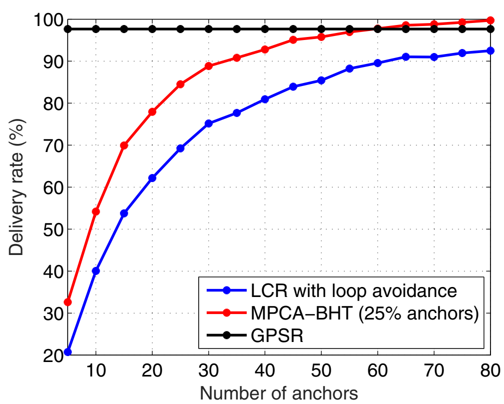

Fig. 7. Routing performance, N=100, Number of PC = 5, Area: 100×100

distance between two nodes. We developed a modified PCA-based method with improved performance. We successfully applied the proposed network topology inference method for traffic routing through the networks. Our planned future research works include the use of robust PCA techniques to deal with missing and corrupted hop counts.

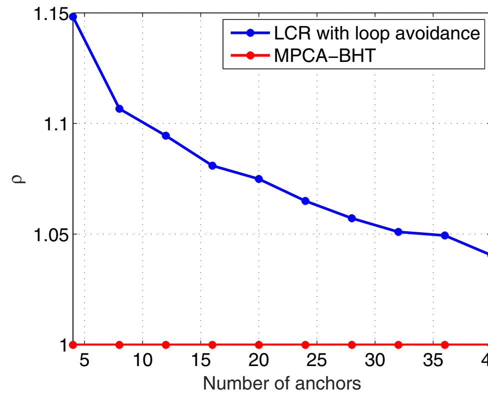

Fig. 8. Routing shortest path performance, N=80, Number of PC = 5, Area: 100×100

REFERENCES

[1] C. D. Dulanjalie and P. J. Anura, "Topology preserving maps from virtual coordinates for wireless sensor networks," in *IEEE 35th Conference on Local Computer Networks (LCN)*, pp. 136–143, Oct 2010.

[2] A. Caruso, S. Chessa, S. De, and A. Urpi, "GPS free coordinate assignment and routing in wireless sensor networks," in *IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 1, pp. 150–160 vol. 1, March 2005.

[3] Q. Cao and T. Abdelzaher, "A scalable logical coordinates framework for routing in wireless sensor networks," in *IEEE 25th International Real-Time Systems Symposium*, pp. 349–358, Dec 2004.

[4] K. Liu and N. Abu-Ghazaleh, "Aligned virtual coordinates for greedy routing in WSNs," in *IEEE International Conference on Mobile Ad Hoc and Sensor Systems*, pp. 377–386, Oct 2006.

[5] B. Wang, C. Fu, and H. B. Lim, "Layered diffusion based coverage control in wireless sensor networks," in *IEEE 32nd Conference on Local Computer Networks*, pp. 504–511, Oct 2007.

[6] D. C. Dhanapala and A. P. Jayasumana, "Anchor selection and topology preserving maps in WSNs; a directional virtual coordinate based approach," in *IEEE 36th Conference on Local Computer Networks (LCN)*, pp. 571–579, Oct 2011.

[7] D. C. Dhanapala and A. P. Jayasumana, "Topology preserving maps; extracting layout maps of wireless sensor networks from virtual coordinates," *IEEE/ACM Transactions on Networking*, vol. 22, pp. 784–797, June 2014.

[8] T. Shao, A. L. Ananda, and M. C. Chan, "Practical connectivity-based routing in wireless sensor networks using dimension reduction," in *IEEE 6th Annual Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, pp. 1–9, June 2009.

[9] D. C. Dhanapala and A. P. Jayasumana, "CSR: Convex subspace routing protocol for wireless sensor networks," in *IEEE 34th Conference on Local Computer Networks*, pp. 101–108, Oct 2009.

[10] S. Tao, A. L. Ananda, and M. C. Chan, "Greedy hop distance routing using tree recovery on wireless ad hoc and sensor networks," in *IEEE International Conference on Communications*, pp. 2712–2716, May 2008.

[11] S. Segarra, A. G. Marques, G. Mateos, and A. Ribeiro, "Network topology identification from spectral templates," in *IEEE Statistical Signal Processing Workshop (SSP)*, pp. 1–5, June 2016.

[12] Y. Shen, B. Baingana, and G. B. Giannakis, "Nonlinear structural equation models for network topology inference," in *Annual Conference on Information Science and Systems (CISS)*, pp. 163–168, March 2016.

[13] B. Holbert, S. Tati, S. Silvestri, T. F. L. Porta, and A. Swami, "Network topology inference with partial information," *IEEE Transactions on Network and Service Management*, vol. 12, pp. 406–419, Sept 2015.

[14] B. Karp and H. T. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks," in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, (New York, NY, USA), pp. 243–254, ACM, 2000.