Topology Inference of Unknown Networks Based on Robust Virtual Coordinate Systems

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

***Abstract*—[1] Learning and exploring connectivity of unknown networks represent an important problem in practical applications of communication networks and social-media networks. Modeling large scale networks as connected graphs, it is highly desirable to extract their connectivity information among nodes to visualize network topology, disseminate data, and improve routing efficiency. This work investigates a simple measurement model in which a small subset of source nodes collect hop distance information from networked nodes in order to generate a virtual coordinate system (VCS) for networks of unknown topology. We establish a VCS to define logical distance among nodes based on principal component analysis (PCA) and to determine connectivity relationship and effective routing methods. More importantly, we present robust analytical algorithm to derive VCS against practical issues of missing and corrupted measurements. We also develop a connectivity inference method which classifies nodes into layers based on the hop distances and derives partial information on network connectivity.**

***Index Terms*—Network connectivity, error measurement, principal component analysis, hop distance.**

I. Introduction

In network applications, information characterizing network connectivity for purposes such as routing and node localization can often be categorized into two types depending on practical constraints and applications. Specifically, geographical coordinate system (GCS) characterizes network nodes locations by using their physical coordinates whereas virtual coordinate system (VCS) characterizes network connectivity by specifying a binary connectivity (or adjacency) matrix that captures pairwise connections of all network nodes [1], [2], [3], [4], [5], [6]. This work focuses on establishing VCS of a network with unknown topology or physical locations for network applications such as topology exploration, information dissemination, and routing.

The VCS is usually generated by using network measurement techniques that differ in complexity and cost of collecting connection information from network nodes. We distinguish hereafter between two vastly different measurement approaches: one based on hop distance and another one based on connection path measurement. On one hand, path-based methods, such as *traceroute* [7], must record all intermediate nodes along the path connecting a source node with a destination node. This approach has the advantage of collecting detailed path information between nodes to be probed such that a single path measurement can establish the node connectivity along the path. Its drawback is the significant amount of network bandwidth and power resources needed to transmit the detailed measurements to a data collection center. Thus, path-based methods are less attractive in practice when dealing with resource-limited and large scaled networks subject to bandwidth and power constraints. This method is mainly used for Internet map discovery [7] without stringent resource constraints. On the other hand, hop-based measurement methods require much less bandwidth and power resources to collect and they are more applicable to simple network architectures and power-limited nodes such as in wireless sensor networks. Even though hop distance measurement is easier to achieve and report, it can still provide valuable network connectivity information to establish the VCS. In this paper, we focus on exploiting the hop distance between network nodes and several anchor or source nodes that can simply be obtained at low cost through a controlled flooding mechanism [8].

Establishing VCS based on hop measurement offers several advantages such as reliability and practical simplicity. Physical distance measurement critically depends on received signal quality, channel distortions or fading, noise, co-channel interference, and synchronization error. Without relying on physical distance measurement, VCS is less sensitive to localization errors and indoor GPS outage. In addition, hop distances are much easier to implement through a simple controlled flooding [9], [8] of beacon signals from several source or anchor nodes. Therefore, hop measurement consumes little bandwidth to transmit and is much more reliable to estimate since hop measurements are integer-valued measurements.

The problem of establishing VCS for unknown networks using anchor based hop distances does pose several unique challenges in practice. To start, it is always practically desirable to utilize fewer anchor nodes and record fewer measurements in order to reduce cost and complexity. For this reason, the decision with respect to the number and the placement of anchor nodes [10] within a network is an important open problem. If the number of anchors is small or their placement is not sufficiently diverse, the accuracy of VCS and the estimated network connectivity will degrade [11]. When used for routing, the performance will be less than satisfactory. On the other hand, too many anchors will substantially increase the cost and the complexity of the measurement as well as

[1]This material is based on work supported by the National Science Foundation under Grants No. CNS-1824553 and CNS-1443870

the required network resources for collecting the measurement data. For large and dense networks, even the low complexity measurement of hop distances can form a large data set. An efficient approach to reduce the effective dimensionality of the VCS is to utilize principal component analysis (PCA) [1], [12], [3], [13]. Another practical challenge for large networks lies in the risk of temporary malfunction or disruption of certain nodes during measurement, thereby leading to critical loss or corruption of measurement information. In the event of security breach, some nodes may even be maliciously hacked to provide erroneous (corrupted) measurement data. Our proposed VCS must be robust against unexpected loss of measurements and corrupted data at unknown locations without causing severe performance degradation when the VCS is used in practical network applications [12], [14], [3].

There exist several related works that utilized the concept of VCS for applications such as network topology preservation and routing. The authors of [1] and [12] applied eigen-analysis to analyze and simplify a given hop matrix in order to produce a Cartesian coordinate map. The authors suggested that the second and the third principal components of the hop measurement matrix appear to provide 2-dimensional (2-D) physical coordinates of the network map, possibly homomorphic to the network physical topology. Although several examples were presented to suggest its potential, such visualization method is highly speculative and does not provide a quantitative metric regarding the virtual connectivity matrix of the network. Other than some visualized 2-D topology examples, no analytical justification or interpretation was given in, e.g.,[12], to support the drastic practice of neglecting the first principal component in PCA-based VCS. Another related work [2] also aims to construct a 3-dimensional (3-D) visual topology map by proposing to exploit hop measurements from only three selected anchors to define zones in which nodes are assigned with the same (similar) coordinates without a VCS. These existing works do not consider missing measurements or corrupted measurements that can occur in practical networks.

Instead of artificially constructing visualizable 2-D or 3-D network maps, the authors of [13] applied a PCA-based dimension reduction on hop measurement between "landmark" (anchor) nodes and the remaining network nodes to compute logical path distances for routing data traffic among nodes in a network. The protocol proposed in [3] lets each node forward its packets to the neighbor node nearest to the destination node within the VCS. Another connectivity-based routing protocol is proposed in [15] where the authors apply a tree based method in order to avoid locally optimum solutions [15].

We note that the aforementioned works in the literature attempt to generate heuristic network topology maps for routing without fully examining the connectivity of the network in terms of adjacency matrix. More importantly, the practical issue of missing measurements or measurement error due to malfunction, power outage, or hacking of certain measurement nodes has never been robustly addressed.

In this work, we investigate new ways to utilize the hop distance measurement efficiently to generate a more accurate and robust VCS. Initially, instead of constructing 2-D topology maps [12][1] that are artificially restricted to only two principal components of the measurement matrix, we analyze the anchor-based hop distance measurement matrix to generate a network VCS that is more informative about the network connectivity. Our special contribution lies in the establishment of virtual coordinates based on the PCA and the associated pairwise logical distances between nodes. Unlike existing works, we provide a clear explanation as to why neglecting the 1st principal component for some network topologies can be analytically justified. However, such practice is fragile and can be replaced by the step of centralizing the mean of the hop measurement. We further exploit the low rank property of the hop distance matrix to develop a robust VCS that is resilient against missing measurement by recovering the missing entries with an adapted matrix completion algorithm. Another key contribution is the derivation of a robust PCA method to locate and recover corrupted measurements because of measurement errors, link failures, or malicious hacking.

To present our problem formulation and technical contributions, this manuscript is organized as follows. Section II presents the system model and the problem formulation of anchor based hop distance measurement for network exploration. We also describe ways to establish the VCS for the network and define the associated logical distances among network nodes. In Section III, we provide an analysis of the proposed VCS to better understand the impact and sensitivity of different principal component selections. In Section IV, we focus on solving the practical problems of missing measurements and measurement errors through robust PCA. In Section V, we present network applications that utilize the hop distance measurements for connectivity inference or traffic routing. We provide numerical results in Section VI to demonstrate the benefits of our robust PCA methods through several simulation test examples before concluding the paper in Section VII.

Notations: We use lower-case letters, bold lower-case, and bold upper-case letters to denote scalars, vectors, and matrices, respectively. If $\mathbf{A}$ is a matrix, then $\mathbf{A}^T$ and $\mathbf{A}'$, respectively, denote the transpose and conjugate-transpose of $\mathbf{A}$ whereas $A(i,j)$ denotes the entry of $\mathbf{A}$ in the i^{th} row and j^{th} column.

II. Network Analysis through Virtual Coordinate Systems

A. Exploring Network Connectivity

Our problem begins with the need to explore a network of unknown topology consisting of N nodes denoted by the set $\mathcal{N} = \{n_1, n_2, \ldots n_N\}$. We denote the network topology by its $N \times N$ connectivity (adjacency) matrix $\mathbf{A}$ in which $A(i,j) = 1$ denotes the existence of a connection between node n_i and node n_j whereas $A(i,j) = 0$, otherwise. The connectivity matrix $\mathbf{A}$ of the network is unknown. Our objective is to estimate the entries of $\mathbf{A}$ based on simple network measurement known as hop distances (or hop measurement).

More specifically, we designate a subset of network nodes $\mathcal{A}$ as anchor nodes that can transmit probing packets to other nodes within the network. To collect hop distances, each network node receiving the beacon packet would simply record the number of hops it has taken from the originating anchor node. Through such a simple method, the anchors can

measure and collect their hop distances to the rest of the network through controlled network flooding. The measurement consists of transmitting probing packets to flood the whole network [9], [8] before collecting the hop measurements from receiving nodes.

Hop distance measurement for network exploration has the clear advantage of simplicity and low cost deployment in comparison with other measurement methods such as traceroute [7]. In fact, we can determine the hop distance (count) between anchor node n_j and any other node n_i by letting node n_j broadcast a probing packet P_j which contains its address α_j and initial hop count $h_j = 0$ to its neighbors which will continue to forward P_j one hop at a time. During this process, any node receiving the tuple (α_j, h_j) in P_j will increment its hop measurement $h_j = h_j + 1$ before forwarding the updated tuple (α_j, h_j) as part of P_j to its own neighbors. To maintain the correct hop measurement i.e. the shortest path's length, each node must always store the minimum value for each anchor address α_j. The flooding process terminates when the probing packets from the anchors reach all the networked nodes. After the flooding and hop distance measurement, each node n_i needs to report, to each anchor n_j or a network center, its own address α_i and its hop distance $H(i,j) = h_j$ where $\mathbf{H}$ denotes the hop measurement matrix collected from the hop distance reports. $\mathbf{H}$ will be used later to analyze the network connectivity. Notice that computation and processing of matrix $\mathbf{H}$ is done offline after uploading the measurements to a server. Since we are only collecting hop count information, the amount of data that needs to be reported by anchor nodes to the server is also light. We also assume that during offline data processing step, the network topology remains static, i.e., there is no link failure or node movement.

Alternatively, one can directly explore network connectivity by applying a traceroute method. In traceroute, network topology information can be extracted by finding the shortest path connecting a large number of node pairs. To establish and report the shortest trace from n_i to n_j, traceroute needs every node to record and report the addresses of all intermediate nodes forming the shortest path between n_i and n_j. It is clear that tracerouting is much more complex to execute and is more costly to report.

Fig. 1 shows a simple example of a small network consisting of 9 nodes $\mathcal{N} = \{n_1, n_2, \ldots, n_9\}$. If we let $\mathcal{A} = \{n_1, n_6\}$ be the anchor set, we can generate a 9×2 hop matrix $\mathbf{H}$ also given in Fig. 1. To illustrate the hop distance measurement, the propagation of hop distance h_1 relative to anchor node n_1 is labeled on top of each node in the network. After executing a controlled flooding, the minimum hop distance from node n_9 to anchor node n_1 is 4. Therefore n_9 needs to report hop measurement $H(9,1) = 4$ along with the address α_9 back to the anchor node n_1. However, in the traceroute method, the full list of addresses $(\alpha_2, \alpha_7, \alpha_8, \alpha_9)$ must be reported to describe the shortest path between n_1 and n_9 and is obviously much more costly.

B. Virtual Coordinate Analysis Based on Hop Measurement

To analyze network connectivity, we select M of the N network nodes as anchors. Denote the subset of anchors as

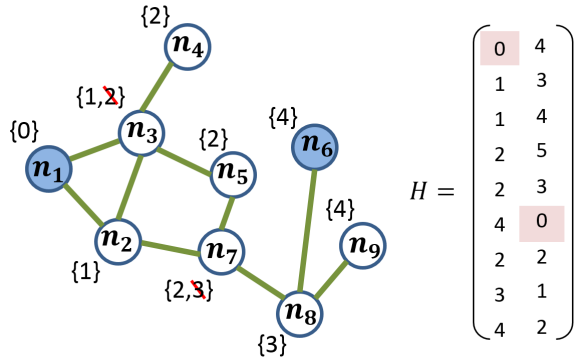

Fig. 1. Graph representing anchored network node and generated anchor measurements

$\mathcal{A} = \{A_1, A_2, \ldots, A_M\} \subset \mathcal{N}$. Once the anchor nodes receive the hop distance reports from all N network nodes, the hop measurements form an $N \times M$ hop matrix $\mathbf{H}$ such that $H(i,j)$ records the shortest hop distance between node n_i and anchor A_j. The hop distance $H(i,j)$ specifies the number of hops through the shortest path linking node n_i to anchor A_j. Since each node n_i collects a hop vector $\mathbf{h}_i$ consisting of hop distances to the M anchors, we can write matrix $\mathbf{H}$ as

$$\mathbf{H} = [\mathbf{h}_1^T, \mathbf{h}_2^T, \ldots, \mathbf{h}_N^T]. \tag{1}$$

We can view $\mathbf{h}_i$ as the *raw* virtual coordinate vector associated with node n_i because it contains all the hop measurements from node n_i to the set of anchors $\mathcal{A}$ without any processing. Using such a raw coordinate system allows us to define the logical distance between any pair of nodes (n_k, n_ℓ) using the l_2-norm as follows

$$\mathrm{d}_h(n_k, n_\ell) = \|\mathbf{h}_k - \mathbf{h}_\ell\|. \tag{2}$$

This logical distance can be used to measure the proximity (such as delay) between any two nodes in the network. If the logical distance between two nodes n_k and n_ℓ is small, then we can conclude that the link distance between n_k and n_ℓ must be small. In fact, we can claim that the number of hops separating n_k and n_ℓ is small. In the most optimistic case, we can even conclude that the two nodes are directly connected, i.e., $A(k,\ell) = 1$.

For large scale and dense networks, we often require many anchor nodes to gather sufficient information for network analysis. Since the dimension of the raw VCS equals M, it is more efficient to process and analyze the collected hop measurement so as to remove the redundantly high dimensionalitiy of the raw VCS. In particular, some preliminary works have applied principal component analysis (PCA) to process the hop measurement matrix for dimension reduction [3], [13] or topological visualization [1], [12].

We aim to analyze the hop matrix to establish reduced-dimension virtual coordinates based on PCA. We shall utilize the logical distance from the VCS to explore the network connectivity and topology. Applying PCA to extract the rank characteristics of the measurement matrix $\mathbf{H}$, we further exploit the low rank property of the measurement matrix to develop robust network analysis algorithms to overcome

missing network measurements because of malfunctions such as power shortage or packet loss. In addition, we also propose a robust PCA algorithm for network connectivity analysis that is resilient against measurement errors caused by practical problems such as hop failures or hackers.

C. PCA and Logical Distance

Applying PCA on $\mathbf{H}$ consists of approximating the column rank of the $N \times M$ measurement matrix $\mathbf{H}$ from M to K ($K < M$). To begin, the first step is to centralize the measurement matrix by removing the mean of matrix $\mathbf{H}$ in order to avoid any translational ambiguity when performing PCA [16]. The mean vector $\boldsymbol{\mu}$ of $\mathbf{H}$ can be defined as:

$$\boldsymbol{\mu} = \frac{1}{N}\sum_{j=1}^{N}\mathbf{h}_j = \frac{1}{N}\mathbf{1}_N^T\,\mathbf{H} \tag{3}$$

where $\mathbf{1}_N$ is the all one vector of dimension $N \times 1$. We denote by $\tilde{\mathbf{H}}$ the zero-mean hop measurement matrix as

$$\tilde{\mathbf{H}} = \mathbf{H} - \mathbf{1}_N\,\boldsymbol{\mu} = \left[\mathbf{I}_N - \frac{\mathbf{1}_N\,\mathbf{1}_N^T}{N}\right]\mathbf{H} = \mathbf{Z}\,\mathbf{H}, \tag{4}$$

where $\mathbf{I}_N$ is the identity matrix of dimension $N \times N$ and $\mathbf{Z} = \mathbf{I}_N - N^{-1}\mathbf{1}_N\mathbf{1}_N^T$. Now we arrive at a centralized measurement matrix $\tilde{\mathbf{H}}$ for PCA.

We perform singular value decomposition (SVD) on $\tilde{\mathbf{H}}$

$$\tilde{\mathbf{H}} = \tilde{\mathbf{U}}\tilde{\mathbf{S}}\tilde{\mathbf{V}}^T = \sum_{i=1}^{M}\tilde{\sigma}_i\tilde{\mathbf{u}}_i\tilde{\mathbf{v}}_i^T, \tag{5}$$

where $\tilde{\mathbf{U}} = [\tilde{\mathbf{u}}_1\ \tilde{\mathbf{u}}_2\ \cdots\tilde{\mathbf{u}}_M]$ and $\tilde{\mathbf{V}} = [\tilde{\mathbf{v}}_1\ \tilde{\mathbf{v}}_2\ \cdots\tilde{\mathbf{v}}_M]$ respectively, are matrices consisting of the left and right singular vectors. The diagonal matrix $\tilde{\mathbf{S}} = \mathrm{diag}(\tilde{\sigma}_1, \cdots, \tilde{\sigma}_M)$ is formed by non-negative singular values $\{\tilde{\sigma}_i\}_{i=1}^{M}$ in descending order.

For dimensionality reduction, we project $\tilde{\mathbf{H}}$ over $[\tilde{\mathbf{v}}_1, \tilde{\mathbf{v}}_2, \ldots, \tilde{\mathbf{v}}_K]$ i.e. the first K columns of $\tilde{\mathbf{V}}$ which correspond to the first K principal components associated with the K strongest singular values. We obtain the $N \times K$ reduced-dimension coordinate matrix

$$\mathbf{G} = \tilde{\mathbf{H}}\,[\tilde{\mathbf{v}}_1, \tilde{\mathbf{v}}_2, \ldots, \tilde{\mathbf{v}}_K]. \tag{6}$$

Since the matrix $\tilde{\mathbf{V}}$ is orthonormal, we can write

$$\mathbf{G} = \tilde{\mathbf{U}}\tilde{\mathbf{S}} = \sum_{i=1}^{K}\tilde{\sigma}_i\tilde{\mathbf{u}}_i\mathbf{e}_i^T, \tag{7}$$

where $\mathbf{e}_i = [0 \ldots 0\ 1\ 0 \ldots 0]$ is the i^{th} vector in the standard basis of $\mathbb{R}^N$.

Note that the rows in $\mathbf{G}$ represent the newly transformed low rank VCS for the network. In other words, if we let

$$\mathbf{G} = [\mathbf{g}_1\ \mathbf{g}_2\ \cdots\ \mathbf{g}_N]^T.$$

Then we have the two virtual coordinate vectors $\mathbf{h}_j$ and $\mathbf{g}_j$ for the j-th node using, respectively, raw VCS and PCA-based VCS as follows:

$$\mathbf{h}_j = \sum_{i=1}^{M}\tilde{\sigma}_i\tilde{u}_i(j)\,\tilde{\mathbf{v}}_i^T, \quad j = 1, 2, \cdots, N \tag{8}$$

$$\mathbf{g}_j = \sum_{i=1}^{K}\tilde{\sigma}_i\tilde{u}_i(j)\,\mathbf{e}_i^T, \quad j = 1, 2, \cdots, N. \tag{9}$$

The logical distance d_h defined by the raw VCS is simply

$$\begin{aligned}
\mathrm{d}_h\,(n_k, n_\ell)^2 &= \|\mathbf{h}_k - \mathbf{h}_\ell\|^2 && (10)\\
&= \left\|\sum_{i=1}^{M}\tilde{\sigma}_i(\tilde{u}_i(k) - \tilde{u}_i(l))\tilde{\mathbf{v}}_i^T\right\|^2 && (11)\\
&= \sum_{i=1}^{M}|\tilde{\sigma}_i|^2\,|\tilde{u}_i(k) - \tilde{u}_i(l)|^2\,\|\tilde{\mathbf{v}}_i^T\|^2 && (12)\\
&= \sum_{i=1}^{M}|\tilde{\sigma}_i|^2\,|\tilde{u}_i(k) - \tilde{u}_i(l)|^2, && (13)
\end{aligned}$$

whereas the logical distance d_g within the PCA-based VCS is given by

$$\begin{aligned}
d_g\,(n_k, n_\ell)^2 &= \|\mathbf{g}_k - \mathbf{g}_\ell\|^2 && (14)\\
&= \left\|\sum_{i=1}^{K}\tilde{\sigma}_i(\tilde{u}_i(k) - \tilde{u}_i(l))\mathbf{e}_i^T\right\|^2 && (15)\\
&= \sum_{i=1}^{K}|\tilde{\sigma}_i|^2\,|\tilde{u}_i(k) - \tilde{u}_i(l)|^2\,\|\mathbf{e}_i^T\|^2 && (16)\\
&= \sum_{i=1}^{K}|\tilde{\sigma}_i|^2\,|\tilde{u}_i(k) - \tilde{u}_i(l)|^2. && (17)
\end{aligned}$$

Therefore, we find that

$$\begin{aligned}
\mathrm{d}_h(n_k, n_\ell)^2 &= d_g(n_k, n_\ell)^2 + \sum_{i=K+1}^{M}|\tilde{\sigma}_i|^2\,|\tilde{u}_i(k) - \tilde{u}_i(\ell)|^2\\
&= d_g(n_k, n_\ell)^2 + R_{k\ell}(K), && (18)
\end{aligned}$$

where $R_{k\ell}(K) = \sum_{i=K+1}^{M}|\tilde{\sigma}_i|^2\,|\tilde{u}_i(k) - \tilde{u}_i(l)|^2$ is a residual term corresponding to less important singular values $\{\tilde{\sigma}_{K+1}, \ldots, \tilde{\sigma}_M\}$. Notice that if $\tilde{\sigma}_i \approx 0,\ \forall i > K$ then $R_{k\ell} \approx 0$ thus $\mathrm{d}_h(n_k, n_\ell) \approx d_g(n_k, n_\ell),\ \forall\, k, \ell \in \{1, 2, \ldots, N\}$. Therefore, the analysis clearly shows that, as long as K is chosen such that $\tilde{\sigma}_i \approx 0,\ \forall i > K$, then the PCA-based low rank VCS can capture the logical distance between nodes with little or no loss of information.

III. Analysis of Different Topology Inference Method

To better understand the proposed VCS and the impact of principal component analysis, we provide several analytical results in this section. We present our analysis by first considering an earlier publication [12] that motivated our work and analysis to better understand the underlying issues.

A. Analytical Comparison with Existing Works on VCS

The authors of [12] proposed an interesting approach to generate a virtual coordinate for network topology by constructing a visualizable 2-D virtual coordinate based on matrix $\mathbf{H}$ and suggested its likely homomorphism with the unknown network topology. In [12], the authors attempted to explore the network topology by projecting matrix $\mathbf{H}$ onto the second and the third principal components. This interesting method naturally generates a 2-D visual graph or map for the unknown network. However, it was unclear whether or

not such a dimension-reduced graph would indeed reveal the true topological information (e.g. homomorphism) about the network connectivity. Here we examine the accuracy of this second and third principal component analysis (STPCA) by neglecting the first principal component and its relationship with our proposed VCS based on PCA.

First, we define the k^{th} principal component (PC) of $\mathbf{H}$ as the k^{th} column of matrix $\mathbf{G}$ defined in (6). We note that the PCA of [12] is performed directly on $\mathbf{H}$ without the step of centralization. In fact, the authors of [12] were unable to explain the reason for ignoring the first PC of $\mathbf{H}$ while choosing the second and third PCs as the 2-D coordinates of the nodes in the derived topology map. In this work, we shall bridge this gap by analyzing the impact of STPCA on $\mathbf{H}$ without mean removal on the PCA analysis and provide a clear connection between the use of STPCA on $\mathbf{H}$ and the PCA on $\tilde{\mathbf{H}} = \mathbf{ZH}$.

Let us examine the first PC of $\mathbf{H}$ which was neglected in [12]. We can write the SVD of $\mathbf{H}$ as follows

$$\mathbf{H} = \mathbf{USV}^T = \sum_{i=1}^{M} \sigma_i \mathbf{u}_i \mathbf{v}_i^T. \tag{19}$$

The first PC of $\mathbf{H}$ is associated with the largest singular value $\sigma_1 = \mathbf{u}_1^T \mathbf{H} \mathbf{v}_1$.

Recall from Eq. (4) that $\mathbf{H} = \tilde{\mathbf{H}} + \mathbf{H}_0$ where $\mathbf{H}_0$ is a rank one matrix

$$\mathbf{H}_0 = N^{-1}\mathbf{1}_N\mathbf{1}_N^T\mathbf{H} = \underbrace{\frac{\|\mathbf{H}^T\mathbf{1}_N\|}{\sqrt{N}}}_{\bar{\sigma}_0} \underbrace{\frac{\mathbf{1}_N}{\sqrt{N}}}_{\bar{\boldsymbol{u}}_0} \cdot \underbrace{\frac{(\mathbf{H}^T\mathbf{1}_N)^T}{\|\mathbf{H}^T\mathbf{1}_N\|}}_{\bar{\boldsymbol{v}}_0^T}$$

$\mathbf{H}_0$ can be seen as a low rank perturbation for matrix $\tilde{\mathbf{H}}$. The work in [17] studied the convergence of the extreme singular values of the perturbed matrix which is in our case $\mathbf{H}$. Theorem 2.9 in [17] states that if $\bar{\sigma}_0$ is greater than some threshold $\bar{\theta}$ then the highest singular value of $\mathbf{H}$ will converge almost surely to some function of $\bar{\sigma}_0$:

$$\sigma_1 \to D_{\mu_H}^{-1}(1/\bar{\sigma}_0^2), \tag{20}$$

where $D_{\mu_H}(.)$ is the D-transform of the singular value distribution of $\mathbf{H}$ denoted by μ_H and $\bar{\theta} = D_{\mu_H}(\tilde{\sigma}_1)^{-1/2}$.

According to (20), we can claim that the mean matrix $\mathbf{H}_0$ directly affects the first PC of the hop measurement matrix $\mathbf{H}$: In fact, it makes the highest singular value σ_1 converge to some function of $\bar{\sigma}_0$ along the PC directions $(\mathbf{u}_1, \mathbf{v}_1)$. Although this partially supports the choice of ignoring the first PC of $\mathbf{H}$ in [12] it does not mean this is the optimal way of obtaining VCS from $\mathbf{H}$. In fact, based on the definition of $\mathbf{Z}$, it is clear that $\mathbf{1}_N\mathbf{1}_N^T\mathbf{Z} = 0$. Thus, we can see that the subspace $\mathcal{R}(\tilde{\mathbf{H}})$ spanned by the columns of $\tilde{\mathbf{H}}$ and the subspace $\mathcal{R}(\mathbf{H}_0)$ spanned by the columns of $\mathbf{H}_0$ are orthogonal:

$$\mathbf{H}_0^T\tilde{\mathbf{H}} = N^{-1}\left(\mathbf{1}_N\mathbf{1}_N^T\mathbf{H}\right)^T\mathbf{ZH} = N^{-1}\mathbf{H}^T\mathbf{1}_N\mathbf{1}_N^T\mathbf{ZH} = 0.$$

However, this orthogonality does not hold for the row subspaces $\mathcal{R}(\tilde{\mathbf{H}}^T)$ and $\mathcal{R}(\mathbf{H}_0^T)$ because $\mathbf{H}_0\tilde{\mathbf{H}}^T \neq 0$. Therefore, we cannot claim that $\bar{\sigma}_0\bar{\mathbf{u}}_0\bar{\mathbf{v}}_0^T = \sigma_1\mathbf{u}_1\mathbf{v}_1^T$. That is why discarding the first PC of $\mathbf{H}$ associated with the singular value σ_1, as in [12], is not the best way to get rid of the mean component associated with the singular value $\bar{\sigma}_0$ because it will engender some connectivity information loss as it is demonstrated in section III-B. Our proposed VCS generation method given by equation (5) consists in performing PCA on $\tilde{\mathbf{H}}$ after centralizing $\mathbf{H}$.

We show in Fig. 2 the variation of the studied singular values for different number of anchors scattered in a randomly generated network composed of $N = 60$ nodes. We notice that $\bar{\sigma}_0$ is always above the threshold $\bar{\theta}$. We also plot the analytical expression of the limit of σ_1 given in (20) and we notice it is close to the numerical value of σ_1. It is also worth noting that $\bar{\sigma}_0$ is close but not exactly equal to σ_1. That is why ignoring the first PC of $\mathbf{H}$ does not always yield the correct VCS especially with random anchor placement as we will demonstrate with numerical examples in the following section.

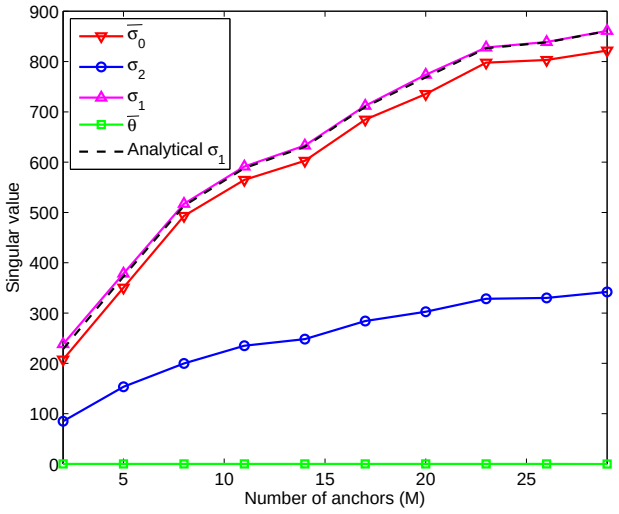

Fig. 2. Variation of the singular values in function of the number of anchors M.

B. PCA Examples

The effect of centralizing $\mathbf{H}$ is further shown through the illustrative examples given in Fig. 3(b) and Fig. 4(b). We also remark from those two figures that the singular values of $\mathbf{H}$ and $\tilde{\mathbf{H}}$ may differ depending on the anchor placement. In particular, when the anchors are more uniformly deployed, as in Fig. 3(a), the second and third PCs of $\mathbf{H}$ are very close to the first and second PCs of $\tilde{\mathbf{H}}$. However, when the anchors are more concentrated in one region of the map, as in Fig. 4(a), the first and second PCs of $\tilde{\mathbf{H}}$ are more significant than the second and third PCs of $\mathbf{H}$. Thus, these pair contain more network's topological information, as clearly seen from the resulting connectivity maps in Fig. 4(c)-(d). In Fig. 4(c), the x and y axes are respectively the second and third PCs of $\mathbf{H}$ and in Fig. 4(d), the x and y axis are respectively the first and second PCs of $\tilde{\mathbf{H}}$.

To quantify the performance of connectivity preservation, we can simply compute the variance of the logical distances between pairwise connected nodes in the network. The benchmark variance is obviously zero since connected nodes are

always 1 hop away from each other (Figs. 3(a),4(a)). Hence, the smaller the variance, the better the topology preservation. Fig. 3(c) and Fig. 3(d) have the same variance (equal to 0.016). However, we observe a much lower variance of 0.059 in Fig. 4(d) when compared with Fig. 4(c) for which the variance is 0.901. Hence, utilizing the first and second PCs of $\tilde{\mathbf{H}}$ as virtual coordinates is more robust to random anchor placement variations than the strategy proposed in [12] which uses the second and third PCs of $\mathbf{H}$. Since the network topology is unknown and the anchor placement tends to be random, instead of directly analyzing the non-negative measurement matrix $\mathbf{H}$, we have shown that analyzing the zero mean matrix $\tilde{\mathbf{H}}$ is not only fully justified for PCA, but also results in topology maps that are more robust to anchor selection.

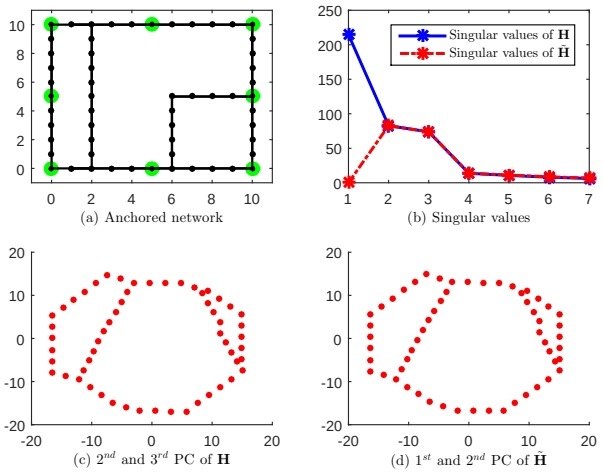

Fig. 3. Hop measurement matrix centralization effect, anchor placement scenario 1

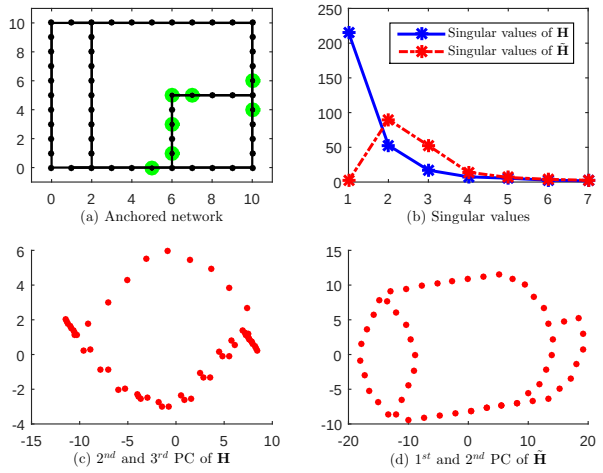

Fig. 4. Hop measurement matrix centralization effect, anchor placement scenario 2

C. K-dimensional logical distance

It is clear that a direct projection onto two singular vectors of $\tilde{\mathbf{H}}$ would neglect other important singular values that play an important role in defining logical distances. In fact, if we use d_2 to denote this benchmark logical distance obtained by projecting $\tilde{\mathbf{H}}$ over $[\tilde{\mathbf{v}}_1\ \tilde{\mathbf{v}}_2]$, we find that

$$\mathrm{d}_h(n_k, n_\ell)^2 = d_2(n_k, n_\ell)^2 + R_{k\ell}(3). \tag{21}$$

A simple comparison shows that the logical distance error between d_h and d_2 is proportional to $R_{k\ell}(3)$, which can be substantial. Therefore, by over reducing the dimensions of $\tilde{\mathbf{H}}$ to 2-D (for visualization purposes), the obtained low-rank graph does not reflect the true logical distance among network nodes. Curiously enough, this method provided apparently homomorphic maps that shared similar shapes with the original network graph for several special pedagogical network shapes chosen from [12]. Here we shall test more generic networks to assess the efficacy of applying logical distances d_g and d_2 to characterize neighboring node relationships.

We construct multiple networks with N randomly deployed nodes in a 2-D coverage area of radius γ. We generate the connectivity edges among the nodes based on physical communication range γ/P. If any two nodes are separated by a distance of γ/P or less, then they will be directly connected. We compare the accuracy of applying the three logical distances to characterize network connectivity or topology. More specifically, we use d_h, d_2 and d_g for different values of K in the following figures by averaging results from 100 Monte Carlo random tests.

In Fig. 5 we use the well known Spearman's rank correlation coefficient to assess the correlation between the hop distance and different logical distances. For each node n_i we form two vectors $\mathbf{v}_h^i$ and $\mathbf{v}_d^i$ of length $N-1$. $\mathbf{v}_h^i$ contains the hop distances between node n_i and all other nodes whereas $\mathbf{v}_d^i$ contains the corresponding logical distances. Spearman's rank correlation method consists in associating rank vectors $\mathbf{r}_h^i$ and $\mathbf{r}_d^i$ with $\mathbf{v}_h^i$ and $\mathbf{v}_d^i$, respectively [18]. The rank coefficient ρ_i associated with node n_i is computed as follows

$$\rho_i = \frac{\sum_{j=1}^{N-1}(r_h^i(j) - \bar{r}_h^i).(r_d^i(j) - \bar{r}_d^i)}{\sqrt{\sum_{j=1}^{N-1}(r_h^i(j) - \bar{r}_h^i)^2 \sum_{j=1}^{N-1}(r_d^i(j) - \bar{r}_d^i)^2}}, \tag{22}$$

where $\bar{r}_h^i$ and $\bar{r}_d^i$ are the mean ranks. Fig. 5 shows the distribution of the correlation coefficients relative to all the network nodes.

In another test, we compute the percentage of nearest neighbor estimation errors denoted as ψ. First, we let $\mathcal{N}_i = \{n_j,\ A(i,j) = 1\}$ define the set of nearest neighbors of node n_i for which a maximum neighbor logical distance radius r_i can be determined

$$r_i = \max_{n_j \in \mathcal{N}_i} d(n_i, n_j) \tag{23}$$

We then compute the number of nodes that have a virtual distance to n_i that is smaller than r_i but are not direct neighbors of n_i:

$$\psi_i = \text{cardinality}\left(\{n_j :\ d(n_i, n_j) < r_i, n_j \notin \mathcal{N}_i\}\right). \tag{24}$$

We can cumulatively obtain the total number of nearest neighbor violations as $\psi_T = \sum_{i=1}^{N} \psi_i$. ψ_T is plotted in Fig. 6 as a function of the number of anchors.

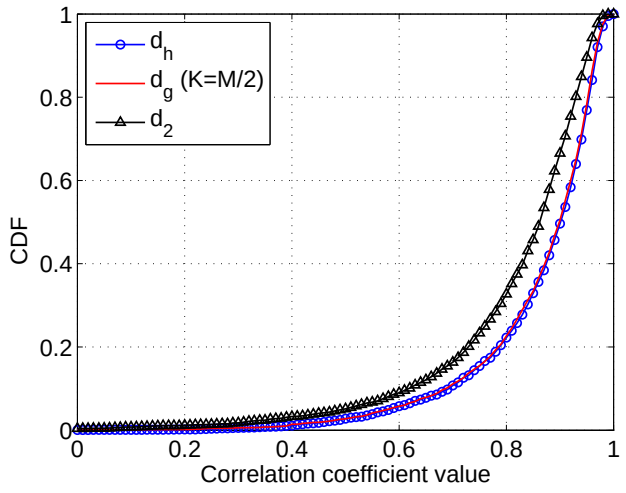

Fig. 5. CDF of correlation coefficient ρ, $N = 50$, $M = 10$, $\gamma = 50$, $P = 4$

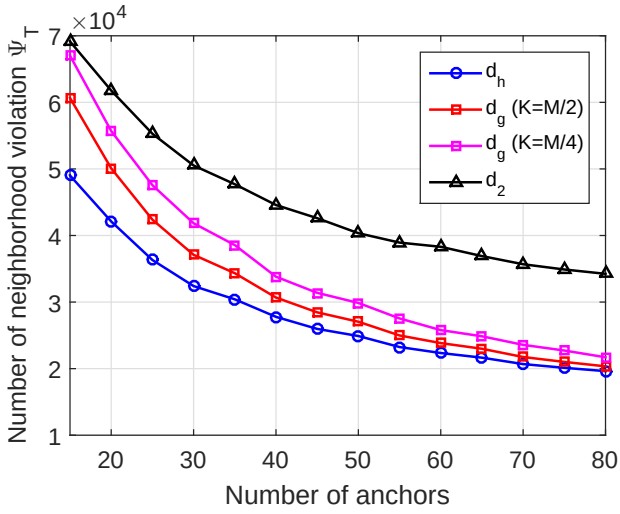

Fig. 6. Logical distance comparison in terms of nearest neighbor violation α, $N = 100$, $\gamma = 100$, $P = 4$

We notice from Fig. 5 and Fig. 6 that d_2 results in higher neighborhood violation and lower correlation coefficient with the hop distance. Therefore, when we perform PCA on the hop matrix $\hat{\mathbf{H}}$ to obtain a VCS, the logical distances based on d_h and d_g provide, as expected, more informative and accurate relationship with respect to nodes and their connectivities. These comparisons shown in Fig. 5 and Fig. 6 illustrate the risk of oversimplification by relying only on two dimensions to construct a VCS.

In summary, PCA captures the low rank property of the hop matrix $\mathbf{H}$ without performance loss. In fact, by considering the first K principal components, we can preserve much of the connectivity information of the full matrix $\mathbf{H}$. More importantly, we shall exploit this low-rank property of $\mathbf{H}$ to derive robust network analysis when faced with missing and corrupted hop measurements in practical scenarios.

IV. Robust Analysis against Incomplete or Erroneous Measurement

Existing works rely on complete measurement matrix for network analysis. In this section, we generalize our network connectivity analysis by modeling two types of practical measurement errors: (a) missing measurement in which some entries of matrix $\mathbf{H}$ are lost; (b) corrupted measurement in which some entries of $\mathbf{H}$ are in error at unknown locations. We now propose two robust algorithms to exploit the low rank property of $\mathbf{H}$ to tackle these two problems.

A. Matrix Completion for Missing Measurements

Recall that hop distances are collected using a technique known as controlled network flooding [8]. Even with such a simple method, there are risks of losing some of the hop measurement entries. The loss of information may be due to the random failure of report channels, node malfunctions, or controlled flooding of limited range. Hence, a small number of entries in $\mathbf{H}$ can be practically absent. Since the missing data in $\mathbf{H}$ will pose challenges to the PCA and the establishment of the VCS, we shall investigate robust means to effectively tackle such problems because of missing data without severely degrading the efficacy of connectivity inference and routing.

Our goal is to generate a virtual coordinate system that can robustly deal with a hop matrix that contains a small fraction of missing entries. Unlike the method proposed in [19] that recovers missing hop measurements using a network-centric imputation technique by dividing routing path into source node, core border and measurement (anchor) node, our fundamental concept is to exploit the low-rank property of the hop matrix $\mathbf{H}$ that has been demonstrated in the previous section. To find the missing data entries, we propose a low rank matrix completion method based on PCA.

As shown in Section II, neglecting those small but non-zero singular values of matrix $\mathbf{H}$ does not seriously affect the logical distance based on PCA. In fact, $\mathbf{H}$ exhibits a strong low-rank property. To model the effect of missing measurements, we propose to decompose the full matrix $\mathbf{H}$ into a sum of a low rank matrix $\mathbf{L}$ and a sparse perturbation matrix $\mathbf{Q}$. In particular, we let

$$\mathbf{H} = \mathbf{L} + \mathbf{Q} \tag{25}$$

where $\mathrm{rank}(\mathbf{L}) < \mathrm{rank}(\mathbf{H})$ and $\mathbf{Q}$ approximates the contribution from the negligible singular values $\{\sigma_i\ ,\ K < i \leq M\}$. To successfully decompose $\mathbf{H}$, the optimization problem can be formulated as follows

$$\begin{aligned} \min_{\mathbf{L},\mathbf{Q}} \quad & \mathrm{rank}(\mathbf{L}) + \lambda \|\mathbf{Q}\|_0 \\ \text{s. t.} \quad & \mathbf{H} = \mathbf{L} + \mathbf{Q}. \end{aligned} \tag{26}$$

We will discuss later in this section how to efficiently solve problem (26). Now, we will denote by δ the perturbation level that forms an upper bound to the Frobenius norm of $\mathbf{Q}$

$$\|\mathbf{Q}\|_F < \delta.$$

To establish the validity of this decomposition based on low rank and sparsity, Fig. 7 shows an example of a random

network with 100 nodes averaging the value of δ over 100 independent Monte Carlo simulation runs. It is clear that the value of the upperbound δ is small and decreases substantially with the number of anchor nodes.

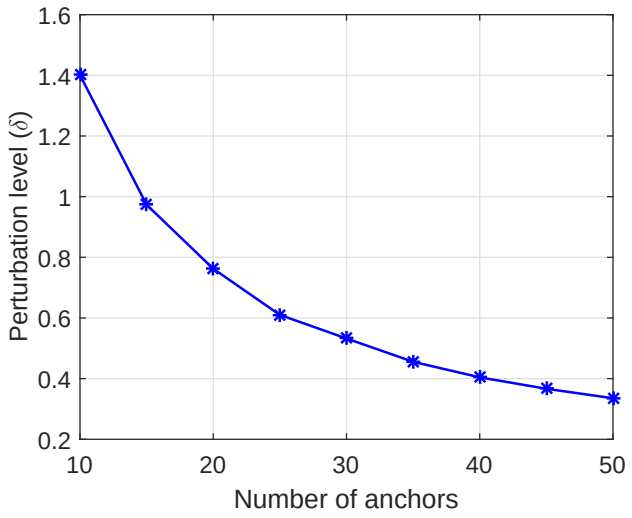

Fig. 7. δ variation with the number of anchors, $N = 100$

Having demonstrated the validity of the low-rank matrix decomposition for $\mathbf{H}$, we now formulate the recovery of missing hop measurements as a noisy low rank matrix completion problem. In order to tackle the missing entries, we let $\Omega = \{(i,j);\ H(i,j) \text{ is observed}\}$ denote the set of indices of $\mathbf{H}$ containing observed hop entries. We would like to recover a low rank matrix $\mathbf{Y}$ that is very close to $\mathbf{H}$ over the set Ω. Hence, we define a linear projection $\mathcal{P}_\Omega$ of a nonnegative $N \times M$ matrix $\mathbf{A}$ such that its (i,j)-th entry is given by

$$\mathcal{P}_\Omega(\mathbf{A})_{i,j} = \begin{cases} A(i,j), & (i,j) \in \Omega \\ 0, & \text{otherwise} \end{cases} \tag{27}$$

We can now formulate the matrix completion problem for hop matrix with missing data entries as follows:

$$\mathbf{P0}: \quad \begin{aligned} \min_{\mathbf{Y}} \quad & \text{rank}(\mathbf{Y}) \\ \text{s. t.} \quad & \|\mathcal{P}_\Omega(\mathbf{Y} - \mathbf{H})\|_F \leq \delta, \end{aligned} \tag{28}$$

where δ is a user-defined value based on experiments similar to those used to generate Fig. 7.

In order to solve the optimization problem (28), we first consider the simple case of a truly low rank matrix $\mathbf{H}$ with missing entries. The problem of matrix completion for a true low rank $\mathbf{H}$ has been studied in [20]. The essence is to search for a low rank matrix $\mathbf{Y}$ that coincides with $\mathbf{H}$ in Ω. This leads to the following optimization problem

$$\begin{aligned} \min_{\mathbf{Y}} \quad & \text{rank}(\mathbf{Y}) \\ \text{s. t.} \quad & \mathcal{P}_\Omega(\mathbf{Y}) = \mathcal{P}_\Omega(\mathbf{H}). \end{aligned} \tag{29}$$

The solution of this problem requires additional conditions. In some cases, such as when $\mathbf{H}$ is the all-zero matrix except for one row (or column), it is clearly impossible to recover $\mathbf{Y}$ where Ω provides so few measurements.

To avoid such pathological examples, we introduce the notion of incoherence with respect to sparse matrices which ensures that a low rank matrix does not have too spiky singular vectors and is not too sparse.

Definition 1: (Matrix incoherence with respect to sparse matrices) A rank-r matrix $\mathbf{X}$ of dimension $N \times M$ is said to be ν-incoherent with respect to the set of sparse matrices if

$$\|\mathbf{u}_i\|_\infty \leq \sqrt{\nu/N}$$
$$\|\mathbf{v}_j\|_\infty \leq \sqrt{\nu/M}$$

where $\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^T$ is the SVD of $\mathbf{X}$, and $\mathbf{u}_i$, and $\mathbf{v}_i$ are the i^{th} row of $\mathbf{U}$ and j^{th} row of $\mathbf{V}$, respectively.

In addition, the problem (29) is generally NP-hard. However, under certain conditions as defined by [20] the problem (29) can be relaxed as follows

$$\begin{aligned} \min_{\mathbf{Y}} \quad & \|\mathbf{Y}\|_* \\ \text{s. t.} \quad & \mathcal{P}_\Omega(\mathbf{Y}) = \mathcal{P}_\Omega(\mathbf{H}), \end{aligned} \tag{30}$$

where $\|\mathbf{Y}\|_*$ is the nuclear norm of $\mathbf{Y}$ (i.e. the sum of all singular values of $\mathbf{Y}$). We now recite this result from [20] as *Theorem 1* here (the proof is found in [20]):

Theorem 1: Let $\mathbf{X}$ be a low rank matrix of rank r and dimensions $N \times M$ where $M < N$. Assume that we observe m entries from $\mathbf{X}$ with locations sampled uniformly at random and that $\mathbf{X}$ satisfies the incoherence assumptions stated in *Definition 1*. Then there exists a positive constant c such as if

$$m \geq c\nu^4 N \log^2(N),$$

then, with probability $1 - N^{-3}$, the exact matrix $\mathbf{X}$ is the solution of the nuclear norm relaxation problem (30).

We further note that our original problem formulation (28) involves a measurement matrix $\mathbf{H}$ that is not strictly low-rank. In fact, $\mathbf{H}$ consists of a sparse perturbation matrix $\mathbf{Q}$ in addition to a dominant low rank component matrix $\mathbf{L}$.

Therefore, we relax problem P0 as follows

$$\mathbf{P1}: \quad \begin{aligned} \min_{\mathbf{Y}} \quad & \|\mathbf{Y}\|_* \\ \text{s. t.} \quad & \|\mathcal{P}_\Omega(\mathbf{Y} - \mathbf{H})\|_F \leq \delta, \end{aligned} \tag{31}$$

and we resort to another result *Theorem 2* whose proof is found in [21].

Theorem 2: Under the assumptions of *Theorem 1*, let $\hat{\mathbf{Y}}$ be the solution of problem (31). Then $\hat{\mathbf{Y}}$ obeys

$$\|\mathbf{H} - \hat{\mathbf{Y}}\|_F \leq 4\sqrt{(2p^{-1} + 1)M}\delta + 2\delta, \tag{32}$$

where $p = m/(NM)$ is the fraction of observed entries of $\mathbf{H}$.

Specifically, this result states that, by solving problem (31), the Frobenius norm of the recovery error is proportional to the perturbation level δ and can be upper-bounded by (32).

We propose to solve problem (31) more efficiently by considering a regularized version which is formulated as follows

$$\mathbf{P2}: \quad \min_{\mathbf{Y}} \quad \frac{1}{2}\|\mathcal{P}_\Omega(\mathbf{Y} - \mathbf{H})\|_F^2 + \tau\|\mathbf{Y}\|_*. \tag{33}$$

Problem $\mathbf{P2}$ of (33) can efficiently be solved using the proximal gradient descent method [16]. It has also been shown that, for a sufficiently large τ [21], the solution of Problem $\mathbf{P2}$ in (33) converges to that of Problem $\mathbf{P1}$ of (31), allowing us to tackle the problem of missing measurement data.

B. Robust PCA against Corrupted Measurement

In addition to missing measurement data, practical networks may also face occasional hacking or temporary hop outage. Such problems can lead to corrupted measurement data at sparse but unknown locations.

We let $\mathbf{H}_c$ denote a sparsely corrupted hop matrix. As a result, we receive the corrupted hop matrix

$$\mathbf{H}_c = \mathbf{H} + \mathbf{E}, \tag{34}$$

where $\mathbf{E}$ is the sparse matrix of errors affecting the accurate but unaccessible hop matrix $\mathbf{H}$. We assume that the elements of $\mathbf{H}_c$ remain integers such that the sparse elements of $\mathbf{E}$ belong to $\mathbb{Z}$. The hop measurement error can arise from:

1) Random node or link failure. In this case, the collected hop distance is no longer the shortest hop path.
2) False hop reports from "*corrupted*" nodes denoted by the set $\mathcal{N}_c \subset \mathcal{N}$.

In both cases, we assume that the locations of the erroneous measurements are uniformly distributed at random. Consequently, the locations of the non-zero, sparse entries of $\mathbf{E}$ are uniformly distributed at random.

Unlike the case of missing observations, treated in section IV-A, the corrupted entries locations denoted as $\Delta = \{(i,j);\ E(i,j) \neq 0\}$ are unknown. This requires us to locate the corrupted entries before correcting them. We propose to combat network measurement errors by adopting the technique of *robust PCA* [16]. We now discuss two different approaches.

1) Iterative re-weighted least square (IRLS): One approach to recover $\mathbf{H}$ from the corrupted hop matrix $\mathbf{H}_c$ is apply the iterative re-weighted least squares (IRLS) algorithm. The idea of IRLS is to apply PCA iteratively by assigning different weights to the observations according to the residual error value.

Specifically, the weight w_{ij} assigned to entry $H(i,j)$ is updated in each iteration via[16]

$$w_{ij} = \theta(e_{i,j})/e_{i,j}^2 \tag{35}$$

where $\theta(.)$ is a loss function and $e_{i,j} = H(i,j) - \mathbf{g}_i^T \nu_j$ in which ν_j is the j^{th} row of matrix $\mathbf{V}$. The advantage of IRLS is simplicity of its implementation. However, it is not guaranteed to converge. For this reason, we propose a better method with guaranteed convergence.

2) Sparse matrix extraction: By combining equation (34) and the system model adopted in (26). The corrupted hop matrix $\mathbf{H}_c$ can be decomposed into

$$\mathbf{H}_c = \mathbf{H} + \mathbf{E} = \mathbf{L} + (\mathbf{Q} + \mathbf{E}), \tag{36}$$

where $\mathbf{L}$ and $\mathbf{Q}$ are the same matrices defined in the system model presented in Section IV-A. We propose to find the corrupted measurements by decomposing $\mathbf{H}_c$ into a low rank matrix $\mathbf{L}_0$ and a sparse matrix $\mathbf{E}_0$ which leads to the following optimization problem

$$\mathbf{P3}: \quad \begin{aligned} \min_{\mathbf{L}_0, \mathbf{E}_0} \quad & \text{rank}(\mathbf{L}_0) + \lambda \|\mathbf{E}_0\|_0 \\ \text{s. t.} \quad & \mathbf{H}_c = \mathbf{L}_0 + \mathbf{E}_0 \end{aligned} \tag{37}$$

Similarly to the case of missing measurements, we need to impose incoherence assumptions (*Definition 1*) to matrix $\mathbf{L}_0$ to prevent it from being too sparse and to avoid ambiguous solutions. We also require that the locations of the non-zero elements of matrix $\mathbf{E}_0$ do not form a conspicuous pattern and are uniformly distributed at random, as stated earlier in this section.

Problem $\mathbf{P3}$ is another difficult nonconvex optimization problem. To find a numerically efficient solution, we formulate a convex relaxation of this problem as

$$\mathbf{P4}: \quad \begin{aligned} \min_{\mathbf{L}_0, \mathbf{E}_0} \quad & \|\mathbf{L}_0\|_* + \lambda \|\mathbf{E}_0\|_1 \\ \text{s. t.} \quad & \mathbf{H}_c = \mathbf{L}_0 + \mathbf{E}_0 \end{aligned} \tag{38}$$

To understand the convergence of problem $\mathbf{P4}$, we recite an important result of [22] here as *Theorem 3* (see [22] for its proof).

Theorem 3: Let $\mathbf{X} = \mathbf{L}_0 + \mathbf{E}_0$ of be an $N \times M$ matrix. Suppose $\mathbf{L}_0$ has low rank and is ν-incoherent with respect to the set of sparse matrices according to *Definition 1*. Assume further that the non-zero elements of the sparse matrix $\mathbf{E}_0$ are uniformly distributed at random (without forming a pattern). If

$$\text{rank}(\mathbf{L}_0) \leq \frac{\rho_d M}{\nu^2 \log^2(N)} \text{ and } \|\mathbf{E}_0\|_0 \leq \rho_s NM$$

for some positive constants ρ_d and ρ_s, then there exists a constant $c > 0$ such that, with probability of at least $1 - cN^{-10}$, the solution of problem (38) for $\lambda = 1/\sqrt{N}$ is exact.

Problem (38) is also known as principal component pursuit and it can be efficiently solved using alternating direction method of multipliers (ADMM) [23]. The solution $\mathbf{E}_0$ obtained by solving problem (38) contains both error corruptions as well as the perturbation components as modeled in equation (36). However, we can easily distinguish the elements of $\mathbf{E}$ from those of $\mathbf{Q}$ because we know that the non-zero elements of $\mathbf{E}$ must belong to the integer set $\mathbb{Z}$. Furthermore, the perturbation level δ introduced by $\mathbf{Q}$ decreases with growing number of anchors (Fig. 7). We can detect and denote the recovered error matrix as $\hat{\mathbf{E}}$.

V. Network connectivity inference and routing applications

Thus far, we have focused on the recovery of the hop matrix against missing measurement or measurement corruptions. Once the recovery of $\mathbf{H}$ and the robust VCS characterizing the logical distances are completed, there are at least two practical applications of interest to consider. The first application uses the hop matrix or hop-based logical distances to extract information about node connectivity. The second application uses the logical distance to perform packet routing within the network.

A. Network connectivity inference

In this section, we propose a method to extract information about the adjacency matrix $\mathbf{A}$ given the (recovered) hop matrix. The proposed method is applicable to any connected graph and is based on a search strategy after organizing the nodes into different layers. The basic principle is presented in our preliminary work [14] to infer network topology of large scale networks from hop distances.

Algorithm 1 describes our layered search algorithm. We begin by assigning a single anchor node A_m to the the first layer labeled $L_{m,0}$. Next, based on the hop measurements collected by anchor A_m i.e. the m^{th} column of matrix $\mathbf{H}$, we place all the nodes that are k hops away from A_m in the k^{th} layer. Denote these sets as $L_{m,k} = \{n_i;\ H(i,m) = k\}$. Finally, we follow the rules of Algorithm 1 by drawing edges between the nodes. Specifically, Algorithm 1 applies the following observations to specify the connectivity between nodes n_i and n_j:

- n_i and n_j cannot be connected if their hop distances with respect to a common anchor A_m, differ by more than one hop.
- n_i and n_j are connected if their hop distances, with respect to a same anchor A_m, differ by exactly one hop, and there is no other node that has the same hop number from A_m, i.e., at least n_i or n_j is the only node in its layer.
- For all other links, an immediate determination cannot be made. Binary hypothesis to detect whether two nodes have direct connectivity can rely on the logical distances [14].

The output of Algorithm 1 is a partial network topology that is given by the incomplete adjacency matrix $\mathbf{A}_c$ in which

$$A_c(i,j) = \begin{cases} 1, & \text{if } n_i \text{ and } n_j \text{ are connected} \\ 0, & \text{if } n_i \text{ and } n_j \text{ are not connected} \\ x, & \text{if connectivity between } n_i \text{ and } n_j \\ & \text{cannot be immediately determined} \end{cases} \tag{39}$$

Data: $\mathbf{H}$
Result: $\mathbf{A}_c$
for *each anchor* A_m **do**
  Generate layers $\{L_{m,k}\}_{k\geq 0}$ by assigning the node A_m to the first layer $L_{m,0}$ and the remaining nodes to lower layers $\{L_{m,k}\}_{k\geq 1}$ 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**
      $A_c(i,j) = 0$
    else if $|h_{n_i,A_m} - h_{n_j,A_m}| = 1$ & $Card\Big(L_{m,min\{h_{n_i,A_m},h_{n_j,A_m}\}}\Big) = 1$ **then**
      $A_c(i,j) = 1$
    else
      $A_c(i,j) = x$
    end
  end
end
Merge the obtained adjacency matrix from each anchor A_m;

Algorithm 1: Proposed connectivity inference algorithm

As an illustrative example, we consider the simple network composed of $N = 9$ nodes shown in Fig. 1. We assume we only have one anchor node $\mathcal{A} = \{n_1\}$. Thus, the generated hop matrix $\mathbf{H}$ is given by

$$\mathbf{H} = \begin{bmatrix} 0 & 1 & 1 & 2 & 2 & 4 & 2 & 3 & 4 \end{bmatrix}^T.$$

Using Algorithm 1 we start by grouping nodes into layers then inferring the connectivity relationships between nodes as shown in Fig. 8 where

- solid edge between n_i and n_j if $A_c(i,j) = 1$
- no edge between n_i and n_j if $A_c(i,j) = 0$
- dotted edge between n_i and n_j if $A_c(i,j)$ is unknown

The obtained connectivity constraints allow us to define the incomplete adjacency matrix $\mathbf{A}_c$ as shown in Fig. 8. It is clear that by using the measurements from multiple anchors at the same time we can reduce the number of unknown adjacency relationships furthermore.

In addition, we can leverage the logical distance between two nodes to detect whether they are connected. In one simple detection algorithm, we can compute the average logical distance $\bar{d}$ of node pairs with known connectivity as

$$\bar{d} = \frac{1}{|\mathcal{K}|} \sum_{(n_i,n_j)\in\mathcal{K}} d_g(n_i,n_j) \tag{40}$$

where $\mathcal{K} = \{(n_i,n_j),\ A_c(i,j) = 1\}$. This average logical distance $\bar{d}$ can serve as a parameter to develop detection thresholds $\delta_0(\bar{d})$ and $\delta_1(\bar{d})$ such that $\delta_0(\bar{d}) < \delta_1(\bar{d})$. If the logical distance between two nodes of unknown connectivity falls below $\delta_0(\bar{d})$, then we can decide that they are directly connected. If the logical distance is higher than $\delta_1(\bar{d})$, then we decide that they are disconnected. Following such a detection algorithm, the number of unknown elements in the connectivity matrix $\mathbf{A}_c$ can be substantially reduced.

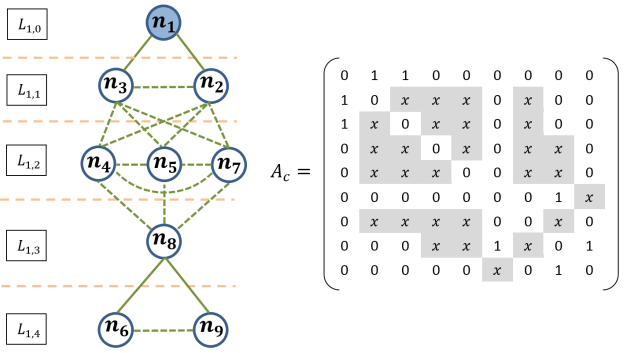

Fig. 8. Graphical representation the output of Algorithm 1

The adjacency inference method discussed in this section allows us to evaluate and demonstrate the advantage of the proposed robust VCS.

B. Routing using virtual coordinates

After recovering the missing and corrupted measurements, we can establish the logical distance between any two nodes. The logical distance is also commonly used in routing applications to optimize the path for packet forwarding.

In prior works such as [13], the authors defined the logical distance using the K most important principal components

for routing. In other words, routing can be based on d_g introduced in II-C. Logical distance d_g derived from PCA not only preserves the logical distance d_h but it also adds more resilience to degenerate anchor nodes set. We shall apply our robust VCS algorithms to estimate and recover the optimum hop matrix from missing and erroneous data before utilizing the routing protocol of [13] based on the logical distance d_g.

The routing protocol [13] simply lets each node forward its packets to its neighbor node that is closest to the packet destination node in terms of logical distance. More specifically, consider a source node S with a reduced dimension virtual coordinate vector $\mathbf{g}_s = [s_1, s_2, \ldots, s_K]^T$ in a K-dimensional space after PCA. Similarly let T be the destination node with coordinate vector $\mathbf{g}_t = [t_1, t_2, \ldots, t_K]^T$. Based on logical distance, node S searches among its neighbors to find the next intermediate hop that has minimum logical distance to destination node T. In case such an intermediate node is not unique, a *fallback* mechanism is activated. It consists in incrementally reducing the dimension of the virtual coordinates space until an intermediate node with a unique minimum logical distance to T is found.

VI. Simulation Results

We generate random wireless networks in our simulations to test our proposed methods for virtual coordinate system generation. We start by randomly deploying N nodes in a circular 2-D coverage area of radius γ. Edges between the nodes are generated based on the communication range $R = \gamma/P$ of the wireless nodes, i.e., any node pair separated by physical distance below R will be connected with an edge. After generating such a random network with unknown topology, we randomly select M anchors and collect the hop measurements by applying the controlled flooding scheme described earlier in Section II. All simulation results are averaged over 100 independent Monte Carlo runs with the network nodes and the anchors locations independently generated for each run according to a uniform distribution.

A. Measurement Recovery

In this part, we test the performance of measurement recovery of our proposed algorithms against missing or corrupted hop entries.

First, we consider the scenario of missing measurements. In Fig. 9 we compare the performance of our proposed matrix completion algorithm **P2** with the benchmark imputation algorithm of [19]. We define the recovery error e_r as the norm of the difference between the recovered and the exact hop matrix. For numerical illustration, we calculate the recovery error

$$e_r = \|\hat{\mathbf{H}} - \mathbf{H}\|_F \tag{41}$$

as a function of the percentage of missing measurements.

From the simulation results in Fig. 9, one can clearly see the performance advantage of our proposed convex optimization algorithm over the network-centric imputation technique proposed in [19].

Next, we consider the scenario of corrupted measurements. We set the number of nodes at $N = 80$ and let the percentage

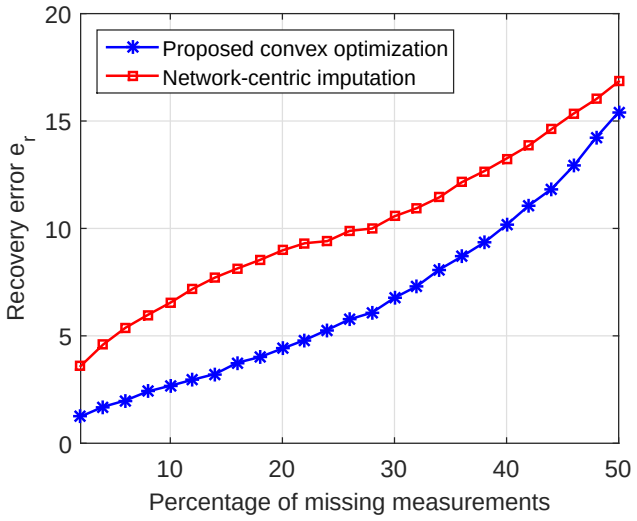

Fig. 9. Comparison of matrix completion methods, $N = 100$, $\gamma = 100$, $P = 4$, $M = N/4$

of corrupted measurements vary between 0% and 10% of the total number (NM) of measurement entries. We define the detection error e_d as the norm of the difference between the true randomly generated error corruption matrix $\mathbf{E}$ and the estimated error matrix $\hat{\mathbf{E}}$ based on optimization **P4** in Section IV-B. Specifically, let

$$e_d = \|\hat{\mathbf{E}} - \mathbf{E}\|_F. \tag{42}$$

We determine the resulting e_d for different numbers of randomly selected anchors and different percentages of corrupted entries. As shown in Fig 10, the detection error e_d decreases with the number of anchors for all percentages of corrupted measurements. Such outcome is fully expected, particular in light of the numerical results from Fig. 7 which shows that the amount of perturbation δ decreases with growing number of anchors.

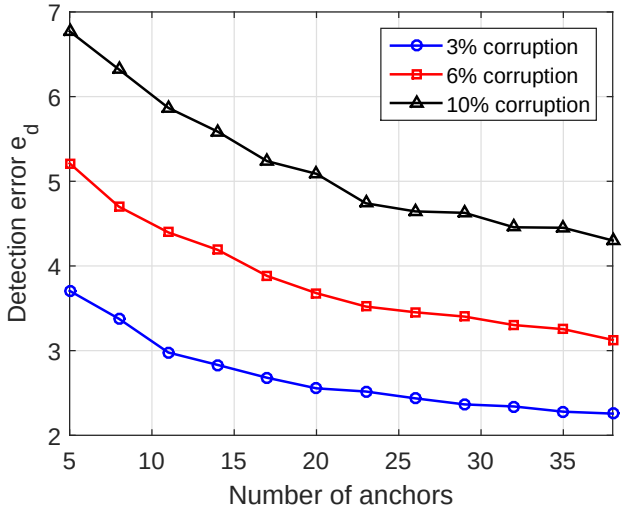

Fig. 10. Detection of Corrupted measurements, $N = 80$, $\gamma = 100$, $P = 4$

B. Connectivity Inference

We test the performance of connectivity inference based on recovered measurement matrix in the presence of missing or corrupted measurements.

For connectivity inference, let e_c denote the fraction of remaining unknown and erroneous entries in the adjacency matrix $\mathbf{A}_c$ after executing Algorithm 1. This error e_c allows us to estimate how much information the hop measurements directly provide about the connectivity relationships between the networked nodes.

In Fig. 11, we first consider the effect of anchor selection on our convex relaxation algorithm against missing measurements. For the network with $N = 100$ nodes, we randomly drop 30% of the hop measurements as missing. We vary the number of anchors from 5% to 50% of the total nodes. We plot e_c for different numbers of anchors and compare the performance based on three virtual coordinate systems:

(a) "Full VCS" based on full measurement matrix;
(b) "Partial VCS" based on measurement matrix with missing entries;
(c) "Robust VCS" based on $\hat{\mathbf{H}}$ through matrix completion of missing entries.

We also show in Fig. 11 the performance from the thresholding method described in Section V-A. We choose $\delta_0(\bar{d}) = 0.9\,\bar{d}$ and $\delta_1(\bar{d}) = 1.1\,\bar{d}$. We notice that in all the cases the percentage of unknown connectivities decreases with the number of anchors because the proposed Algorithm 1 allows us to know more about the adjacency matrix by considering additional anchor nodes. In addition, our robust approach ensures a better connectivity inference which is close to the full VCS with complete hop measurements.

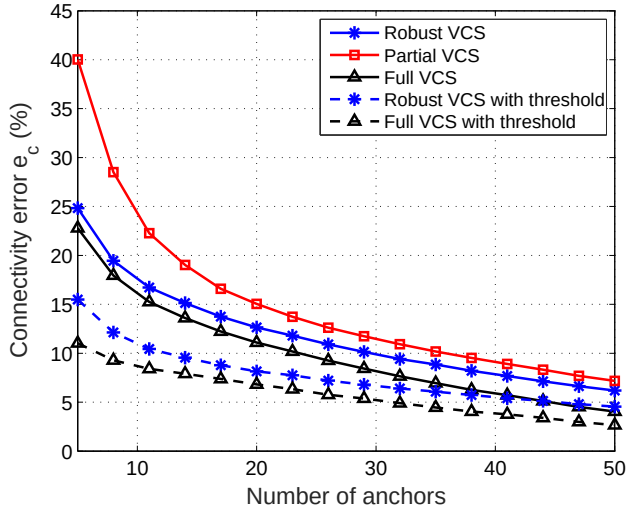

Fig. 11. Performance of adjacency inference (Algorithm 1) in presence of missing hop measurements, $N = 100$, $\gamma = 100$, $P = 4$, $K = M/2$, percentage of missing measurements 30%

Another practical case of missing hop measurement is from applying limited controlled flooding. In some cases such as large scale or resource limited networks, reducing resource usage by limiting the controlled flooding can help prolong the network lifetime and adapt the topology inference method to limited resource applications. To do so, we fix a hop distance upper limit h_m so that anchors can only probe nodes which are at most h_m hops away. The parameter h_m can be seen as the exploration range in hop units. Thus, the hop measurements relative to nodes which are more than h_m hop away from the anchor are considered unknown or missing, for which we can apply the robust VCS algorithm. Fig. 12 shows that our method is able to recover the missing measurements for different values of h_m.

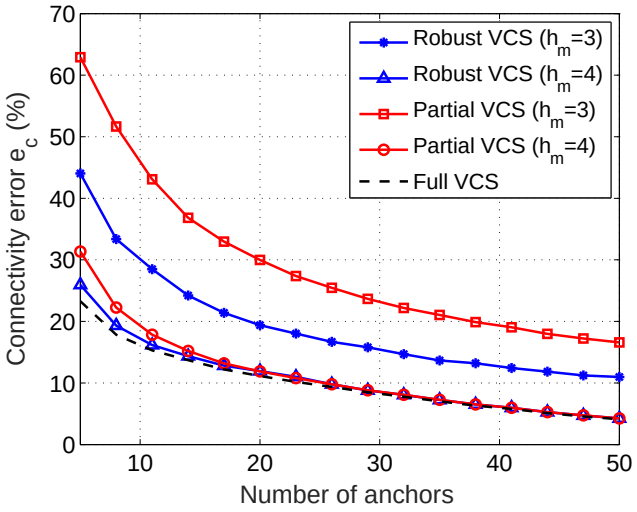

Fig. 12. Performance of adjacency inference (Algorithm 1) in the case of limited flooding, $N = 100$, $\gamma = 100$, $P = 4$, $K = M/2$

Next, we test the proposed methods with larger scale networks. We increase the number of nodes to 400. Fig. 13 shows that even with a large number of nodes we can keep a performance similar to Fig. 11.

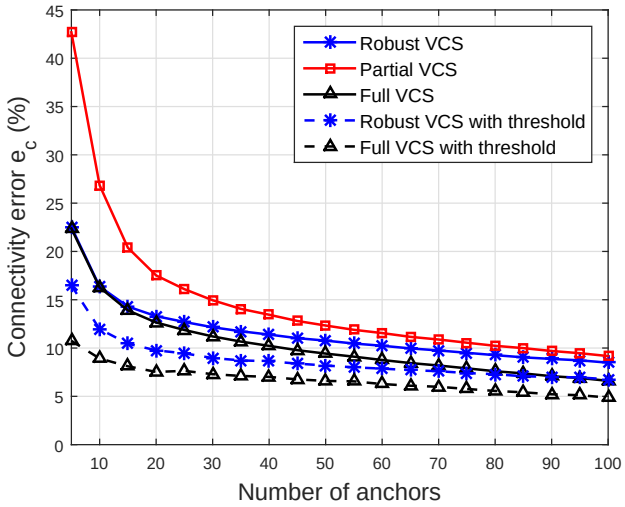

Fig. 13. Performance of adjacency inference (Algorithm 1) in presence of missing hop measurements, $N = 400$, $\gamma = 100$, $P = 5$, $K = M/2$, percentage of missing measurements 30%

Similarly to Fig. 11, Fig. 14 demonstrates the results of

connectivity in the presence of measurement corruption. We let 5% of the hop measurements be randomly corrupted with a sparse error matrix $\mathbf{E}$. We also test three virtual coordinate systems:

(a) "Full VCS" relying on zero measurement corruption;
(b) "Corrupted VCS" based on measurement matrix with corrupted entries;
(c) "Robust VCS" based on robust PCA detection and recovery of corrupted entries.

Comparing the detection errors of the three different VCS, we see a considerable improvement in the accuracy of connectivity inference by the proposed robust VCS method. In fact the robust VCS method achieves nearly the same connectivity inference performance as that of the full VCS without measurement errors. This result demonstrates the strength and the robustness of the proposed robust PCA algorithm for establishing a more robust and practical VCS.

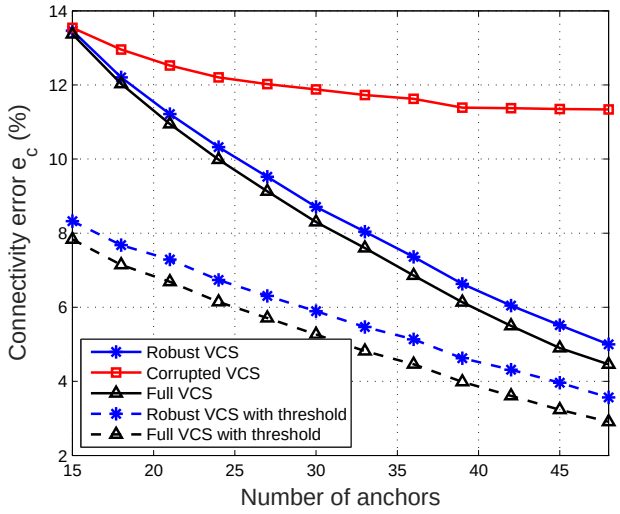

Fig. 14. Performance of adjacency inference (Algorithm 1) in presence of corrupted hop measurements, $N = 100$, $\gamma = 100$, $P = 4$, $K = M/2$, percentage of corrupted measurements 5%

C. Traffic Routing Successes

For traffic routing, we also generate random networks with random anchor positions. We randomly choose the starting and destination nodes for traffic routing.

For comparison purposes, we include a geographical-coordinate-based routing method called greedy perimeter stateless routing (GPSR) [24] as a benchmark performance. This method is more efficient and obviously independent of the number of anchors but it requires accurate geographical coordinates. Such requirement is costly and leads to high hardware and software complexity which is not practical for networks with limited resources.

Fig. 15 shows the resulting packet delivery rates (percentage of successfully delivered packets) under different numbers of anchors. We suppose that 10% of the hop measurements are randomly missing. Similar to the connectivity inference tests in Section VI-B, we test the performance of three different systems: "full VCS", "partial VCS", and "robust VCS". The results clearly demonstrate that our robust VCS can lead to much higher successful delivery rate over "partial VCS" that does not recover the missing measurements. In fact, robust VCS achieves performance that is nearly identical to the full VCS without missing measurements.

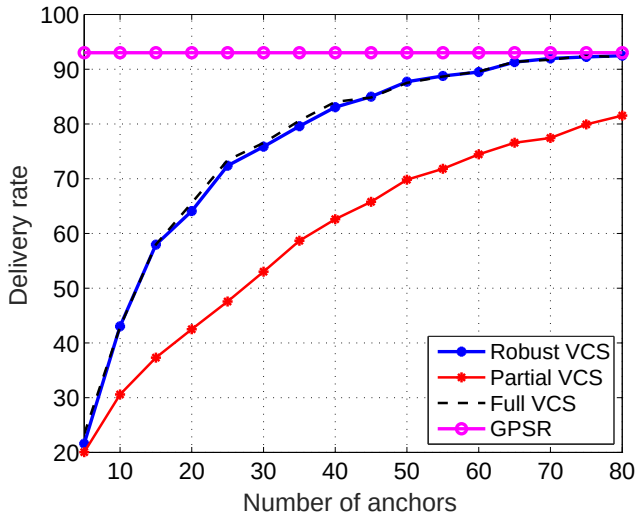

Fig. 15. Performance of routing in presence of missing hop measurements, $N = 100$, $\gamma = 100$, $P = 4$, $K = M/2$, percentage of missing measurements 10%

We now consider that 5% of the hop measurements are randomly corrupted with errors. We test the three cases of VCS: "full VCS", "Corrupted VCS" and our proposed "Robust VCS". Fig. 16 illustrates the resulting packet delivery rates as a function of the number of anchors. Once again, the proposed robust VCS can generate much higher successful delivery rate over "corrupted VCS" that neither locates nor recovers the 5% error measurement. In fact, robust VCS achieves performance that is nearly identical to the full VCS without any measurement errors.

VII. Conclusion

In this work, we proposed to utilize logical distance metric according to a PCA based virtual coordinate system to infer unknown network topology by relying only on simple hop distance measurements. This approach allowed us to capture the low rank property of the measurement matrix. To ensure resilience of our VCS against practical issues of missing and corrupted measurement, we presented two different convex optimization algorithms to recover missing or corrupted measurements. Our proposed topology inference approaches are based on the powerful tools of low-rank matrix completion and sparse signal processing for restoring the imperfect hop matrix. Our numerical simulation results demonstrate the robustness of the proposed VCS for recovering missing measurements and for locating and correcting measurement errors. Our tests also demonstrated substantial performance improvement offered by our new VCS algorithms in applications involving network connectivity inference and traffic routing. One of our future

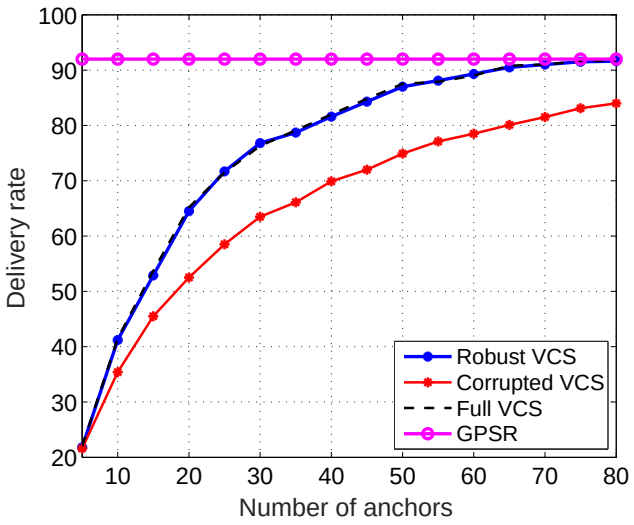

Fig. 16. Performance of routing in presence of corrupted hop measurements, $N = 100$, $\gamma = 100$, $P = 4$, $K = M/2$, percentage of corrupted measurements 5%

works is to investigate new method for network connectivity inference under simultaneous missing and corrupted measurements.

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] A. Rao, S. Ratnasamy, C. Papadimitriou, S. Shenker, and I. Stoica, "Geographic routing without location information," in *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking*, MobiCom '03, pp. 96–108, 2003.

[6] 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," in *IEEE 26th International Conference on Computer Communications*, pp. 2234–2242, May 2007.

[7] R. Govindan and H. Tangmunarunkit, "Heuristics for internet map discovery," in *Proceedings of IEEE INFOCOM*, vol. 3, pp. 1371–1380 vol.3, Mar 2000.

[8] O. Liang, Y. A. Sekercioglu, and N. Mani, "A low-cost flooding algorithm for wireless sensor networks," in *IEEE Wireless Communications and Networking Conference*, pp. 3495–3500, March 2007.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] T. Bouchoucha, N. Chuah, and Z. Ding, "Finding link topology of large scale networks from anchored hop count reports," in *IEEE Global Telecommunications Conference*, 2017.

[15] 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.

[16] Y. Ma, S. Sastry, and R. Vidal, *Generalized Principal Component Analysis*. Interdisciplinary Applied Mathematics, Springer New York, 2015.

[17] F. Benaych-Georges and R. R. Nadakuditi, "The singular values and vectors of low rank perturbations of large rectangular random matrices," *J. Multivar. Anal.*, vol. 111, pp. 120–135, Oct. 2012.

[18] Y. Dodge, *The Concise Encyclopedia of Statistics*. Springer New York, 2008.

[19] B. Eriksson, P. Barford, R. Nowak, and M. Crovella, "Learning network structure from passive measurements," in *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement*, IMC '07, (New York, NY, USA), pp. 209–214, ACM, 2007.

[20] E. J. Candès and B. Recht, "Exact matrix completion via convex optimization," *CoRR*, vol. abs/0805.4471, 2008.

[21] E. J. Candes and Y. Plan, "Matrix completion with noise," *Proceedings of the IEEE*, vol. 98, pp. 925–936, June 2010.

[22] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?," *CoRR*, vol. abs/0912.3599, 2009.

[23] Z. Lin, R. Liu, and Z. Su, "Linearized alternating direction method with adaptive penalty for low-rank representation," *Advances in Neural Information Processing Systems 24*, pp. 612–620, 2011.

[24] 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.