Estimation of Message Source and Destination from Link Intercepts

Derek Justice and Alfred Hero

Department of Electrical Engineering and Computer Science

University of Michigan

Ann Arbor, MI 48109

Abstract

We consider the problem of estimating the endpoints (source and destination) of a transmission in a network based on partial measurement of the transmission path. Sensors placed at various points within the network provide the basis for endpoint estimation by indicating that a specific transmission has been intercepted at their assigned locations. During a training phase, test transmissions are made between various pairs of endpoints in the network and the sensors they activate are noted. From these possibly noisy measurements, we develop necessary constraints that any feasible network topology must satisfy. Randomized rounding of the solution to a semidefinite programming relaxation generated from the constraints is used to produce samples of network topologies defined over the feasible set. When a subset of the deployed sensors are activated, corresponding to the occurrence of a transmission with unknown endpoints, a monte carlo approximation of the posterior distribution of source/destination pairs given the activated sensors is computed by averaging over the topology samples and used in maximum a posterior estimation of the endpoints. We illustrate the method using simulations of power-law random topologies.

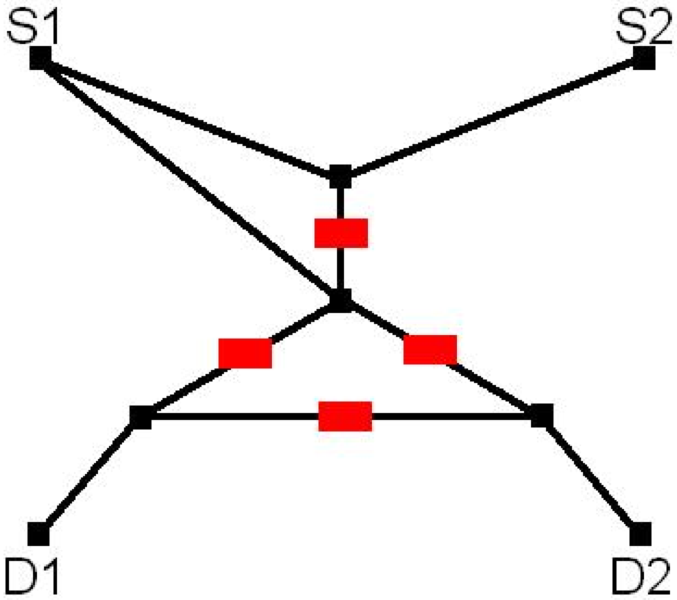

Fig. 1. Diagram of the measurement appartus on a sample network. Probing endpoints are labeled (s1, d1) and (s2, d2). A box on a link represents a sensor that indicates when a transmission path intercepts that link.

I. Introduction

We present a method to estimate the endpoints (source and destination) of a data transmission in a network whose logical topology is unknown. We assume there are a number of asynchronous sensors placed on some subset of the links in a network. A sensor is activated, and its activation recorded, whenever the path of a data transmission is intercepted on the link where the sensor is situated. If multiple sensors are activated by a single transmission, they are not capable of providing the order in which they were activated. During a preliminary training phase, the network is probed by transmitting data packets between various pairs of external nodes $\{(s,d)_i\}$, and the link sensors p activated by each transmission are recorded. The measurement apparatus is illustrated on a sample network in Fig. 1.

The resulting data $\{(s,d,p)_i\}$ is processed by the system shown in Fig. 2 so that when a sensor configuration p_x with unknown endpoints (referred to as the *suspect transmission*) is noted, estimates of its source and destination (s_x, d_x) may be provided.

Some information about the network topology is necessary in order to estimate the endpoints of the suspect transmission, however the topology of our network is unknown. We use the data

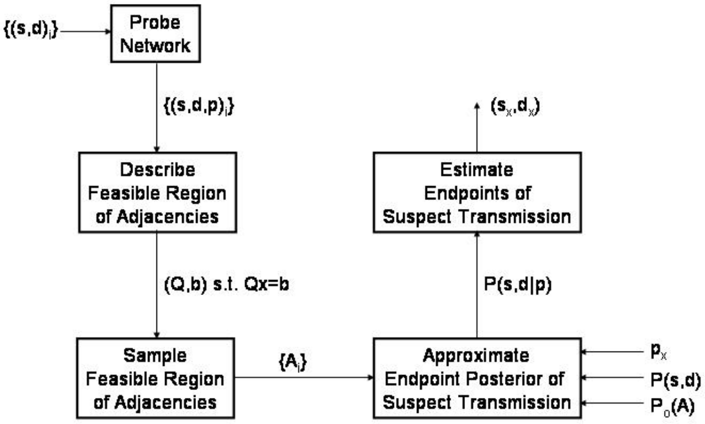
{(s,d)_i}
Probe Network
{(s,d,p)_i}
Describe Feasible Region of Adjacencies
(Q,b) s.t. Qx=b
Sample Feasible Region of Adjacencies
{A_i}
Approximate Endpoint Posterior of Suspect Transmission
p_x
P(s,d)
P_0(A)
P(s,d|p)
Estimate Endpoints of Suspect Transmission
(s_x,d_x)

Fig. 2. Diagram of the transmission endpoint estimation system, assuming internal link sensors have already been deployed.

obtained in the probing phase to precisely describe the space of feasible topologies by translating the data into linear constraints that a topology's adjacency matrix representation must satisfy. The constraints are in the form $Qx = b$ where Q is a 0-1 matrix, b is an integer vector, and x is a vectorized version of the 0-1 adjacency matrix. The definition of Q and b naturally depends upon whether the network of interest is directed or undirected; both cases are considered. Given Q and b the computation of feasible solutions to the linear constraint equation is no small task, in fact it is known to be an NP-Complete problem [1]. We consider the associated minimum norm problem $min \left\| Qx - b \right\|_W^2$ where $x \in \{0,1\}^n$ and $\left\| \cdot \right\|_W$ is a quadratic norm with respect to the positive definite matrix W. It is known that combinatorial optimization problems of this type may be successfully approximated by 'lifting' them into a higher dimensional matrix space where $X_{ij} = x_i x_j$ and $X \in \{0,1\}^{n \times n}$ [2].

With the advent of polynomial time interior point methods for linear programming that can be extended to semidefinite programming [3], [4], it is convenient to consider a semidefinite programming (SDP) relaxation of the higher dimensional problem. Indeed, SDP relaxations have proven to be powerful tools for approximating hard combinatorial problems [5], [6], [7], [8]. The SDP, however, is solved over a continuous domain so it is necessary to retrieve a 0-1 solution from the possibly fractional SDP solution. One possibility is a branch and bound scheme whereby certain variables are fixed and the SDP is repeated until a discrete solution is found [1], [8]. The branch and bound algorithm can take an exponential amount of time, depending on how tight the desired bound is. A randomized rounding scheme was developed in [6] for SDP relaxations of the MAXCUT and MAX2SAT problems. This scheme is shown to produce solutions of expected value at least 0.878 times the optimal value in [6]. We develop an SDP relaxation of the 0-1 minimum norm problem and apply the randomized rounding method to produce a number of network topology adjacency matrices $\{A_i\}$ that approximately satisfy the linear constraints $Qx = b$. We derive an expression for the expected value of the squared error $E\left[\|Qx - b\|_W^2\right]$ of samples produced in this way. This expression depends on the solution of the SDP relaxation, but an upper bound on the error independent of the SDP solution is also given.

The network topology samples are used in conjunction with prior distributions on endpoints $P(s, d)$ and topologies $P_o(A)$ to compute a Monte Carlo approximation of the posterior distribution of endpoints given the suspect transmission $P(s, d|p_x)$ via Bayes rule. Bayes formula for this problem essentially reduces to the expected value of a functional of the topology A; our approximation of the endpoint posterior thus becomes a weighted averaged of the values of this functional at each sample topology A_i where the weights are determined by the prior distribution $P_o(A)$. It is readily apparent that this functional requires the conditionals $P(p_x|s, d, A)$ for all s, d and A (the path likelihood functions). We propose a likelihood model for which longer paths between a specific source/destination are no more likely than shorter paths between the same pair. Furthermore, instead of normalizing the conditionals over all feasible paths p, we normalize over the k-shortest paths, which can be computed in polynomial time using an algorithm described in [9]. With the posterior distribution $P(s, d|p_x)$ in hand, we can immediately give the MAP estimate of (s_x, d_x) or an a posteriori confidence region of probable source/destination pairs.

The related area of network tomography has recently been a subject of substantial research. It refers to the use of traffic measurements over parts of a network to infer characteristics of the complete network. Some characteristics of interest include the following: source/destination traffic rates [10], [11], link-level packet delay distributions [12], [13], link loss [14], and link topology [15], [16]. For an overview of relevant tomography problems for the Internet see [17]. In many applications, the tomography problem is ill posed since data is insufficient to determine a unique topology or delay distribution.

Our work is related to the internally sensed network tomography application described in [18], [19]. These works propose a methodology for estimating the topology of a telephone network using the measurement apparatus illustrated in Fig. 1. The data transmissions are of course telephone calls and the asynchronous sensors are located on trunk lines. A simple argument in [19] demonstrates that the number of topologies consistent with the data measured during the probing phase $\{(s,d,p)_i\}$ is exponential in the number of link sensors. Indeed the problem is ill-posed as the data required to provide a reasonable estimate of the topology will never be available in practice. We sidestep the difficulties of developing a single topology estimate by averaging over many feasible topologies in computing the endpoint posterior distribution of a suspect transmission.

The solution approach we develop is very general, and we suspect it might have application in all sorts of networks: including telephone networks as described in [18], ad hoc networks, social networks, or biological networks [20]. As a provocative example, consider a social network formed for the purposes of covertly distributing some product (such as weapons technology or a controlled substance). Here, suppliers of the product would play the role of source nodes and consumers the role of destination nodes. The link sensors would consist of middlemen willing to indicate a particular request was processed, but nothing more. The network might be probed by initiating requests to a specific supplier in the vicinity of a likely consumer. Based on information from the middlemen, it would be possible to localize likely suppliers and consumers of a particular transaction using these methods.

The approach described here might also find utility in systems conveniently modeled by graphs, such as finite state automata. The problem of machine identification is a classic problem in the theory of automata testing [21], [22]. Here, we are given a black box with an automaton inside whose transition function is unknown. Based on the response of the system to certain input sequences, we wish to reconstruct the transition function. The link to the network topology recovery aspect of our problem is clear, since a graph provides a convenient representation for the transition function of interest. The external nodes chosen in the probing phase of our problem is analagous to the input sequences to the black box automaton. Similarly, link sensors correspond to events in the automaton's observable event set. An exhaustive algorithm for solving this problem is given in [21] and shown to have exponential run time. Our methods might be adapted to provide a polynomial time approximation algorithm.

The outline of this paper is as follows. We review the problem, describe in detail each component of the endpoint estimation system (Fig. 2), and analyze its complexity in Section II. In Section III, we provide some simulations of small power-law random graphs. These are random graphs whose vertex degrees follow a power law distribution. Such graphs are observed to occur frequently in natural and synthetic systems [20], [23]. We generate them according to the configuration model described in [20], [24]. In Section IV we provide some reasonable extensions of this problem utilizing feedback with the graph edit distance as a metric [25] to suggest an adaptive system and finally offer some concluding remarks.

II. Source-Destination Estimation and System Model

Let $G(V, E, f)$ be a simple graph defined by the vertex set V, edge set E, and incidence relation $f : E \to V \times V$. The adjacency matrix A associated with G is given by

$$A_{ij} = \begin{cases} 1 & \text{if } \exists e \in E \text{ and } v_i, v_j \in V \text{ such that } f(e) = (v_i, v_j) \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

We allow G to be either directed or undirected; however, it should be known a priori which is the case. It follows easily that if G is undirected, then A is symmetric. In our application, E defines the set of links in the network topology, V defines the routers or switches connected by these links, and f determines the pair of routers/switches connected by each link.

A path between vertices $u \in V$ and $v \in V$ is given by $\tilde{p}_{uv} \subseteq E$, where $\tilde{p}_{uv}$ contains the edges passed in the path from u to v. Because the sensors are asynchronous, they are not capable of providing ordering information. Thus we assume the paths are unordered sets. Let $E_l \subseteq E$ be the set of edges on which sensors are placed. Let $T \subseteq V$ be the set of external vertices, that is nodes that can send and/or receive data transmissions.

The purpose of our system is to estimate the source and destination of an activated sensor set p_x corresponding to a transmission whose endpoints are unknown (i.e. suspect transmission). We utilize a Bayesian framework to produce suitable approximations of the endpoint posterior distribution:

$$P(s, d|p_x) = \sum_A \frac{P(p_x|s, d, A)P(s, d)}{\sum_{s,d} P(p_x|s, d, A)P(s, d)} P_o(A) \tag{2}$$

by assuming the availability of appropriate prior distributions on communication $P(s, d)$ and network topology $P_o(A)$ and introducing a model for the conditional path probabilities $P(p_x|s, d, A)$ based on shortest path routing. The posterior in Eq. (2) is approximated by summing over the argument evaluated at a number of topology samples $\{A_i\}$. The solution to a semidefinite programming relaxation is randomly rounded to produce topology samples that approximately satisfy linear constraints derived from the measurements obtained in preliminary probing of the network. With the approximate endpoint posterior distribution in hand, we can provide MAP estimates of the endpoints (s_x, d_x) of the suspect transmission p_x and compute appropriate error measures.

A. Probing the Network

Along the lines of the network tomography paradigm, the probing phase consists of swapping data transmissions between pairs of external nodes in T and observing which link sensors are activated in response. Let $T_s \subseteq T$ be the set of source nodes from which data transmissions originate and $T_d \subseteq T$ be the set of destination nodes at which data transmissions terminate. The user supplies a set of source and destination pairs $\{(s, d)_i\}$ where $s \in T_s$ and $d \in T_d$. The probing mechanism passes a data transmission from s_i to d_i for each i and notes the sensors activated by each transmission $p_{sd} \subseteq E_l$. Note that since a sensor may not be on every link in the network (i.e. $E_l \subset E$), p_{sd} is related to $\tilde{p}_{sd}$ (the path in the true network) by the following

$$p_{sd} = \tilde{p}_{sd} \cap E_l \tag{3}$$

The probing phase provides the measurements $\{(s,d,p_{sd})_i\}$ that may be used to define the feasible region of network topologies.

We also allow for errors in the sensor measurements. Suppose that each link sensor $e \in E_l$ has an associated miss probability $\alpha_m(e) = P(e \notin p_{sd} | e \in \tilde{p}_{sd})$ and false alarm probability $\alpha_f(e) = P(e \in p_{sd} | e \notin \tilde{p}_{sd})$. In this case, the probing mechanism repeats the data transmission from s_i to d_i N times for each i. These N measurements are used to construct a maximum likelihood estimate $\hat{p}_{s_i d_i}$ of each path $p_{s_i d_i}$ according to the following model. Along the lines of a generalized likelihood approach, the probing mechanism passes along the maximum likelihood path estimates for each $(s,d)_i$, given by $\{(s,d,\hat{p}_{sd})_i\}$, for use in defining the feasible region of network topologies.

Define the path indicator vector ν^{sd} whose elements are given by $\nu^{sd}(j) = I_{p_{sd}}(e_j)$ for all $j = 1, 2, \ldots |E_l|$ where $I_A : A \to \{0,1\}$ is the usual indicator function. If we assume sensor errors are independent across paths and measurements, then the joint probability mass function of the N observed path vectors for a given source/destination pair ν_i^{sd} is

$$\begin{aligned} p_{sd}\left(\nu_1^{sd}, \nu_2^{sd}, \ldots \nu_N^{sd} | \tilde{\nu}^{sd}\right) = \textstyle\prod_{k=1}^{N} \prod_{j=1}^{|E_l|} \alpha_m(e_j)^{\left(1-\nu_k^{sd}(j)\right)\tilde{\nu}^{sd}(j)} \beta_m(e_j)^{\nu_k^{sd}(j)\tilde{\nu}^{sd}(j)} \ldots \\ \alpha_f(e_j)^{\nu_k^{sd}(j)\left(1-\tilde{\nu}^{sd}(j)\right)} \beta_f(e_j)^{\left(1-\nu_k^{sd}(j)\right)\left(1-\tilde{\nu}^{sd}(j)\right)} \end{aligned} \tag{4}$$

where $\beta_m(e) = 1 - \alpha_m(e)$ and $\beta_f(e) = 1 - \alpha_f(e)$. If we define the likelihood function $L(\tilde{\nu}^{sd})$ as the logarithm of the expression in Eq. (4), then it may be written explicitly as

$$\begin{aligned} L(\tilde{\nu}^{sd}) = \textstyle\sum_{j=1}^{|E_l|} \left(N \log \beta_f(e_j) + \sum_{k=1}^{N} \nu_k^{sd}(e_j) \log \frac{\alpha_f(e_j)}{\beta_f(e_j)}\right) + \ldots \\ \textstyle\sum_{j=1}^{|E_l|} \left(N \log \frac{\alpha_m(e_j)}{\beta_f(e_j)} + \sum_{k=1}^{N} \nu_k^{sd}(e_j) \log \frac{\beta_m(e_j)\beta_f(e_j)}{\alpha_m(e_j)\alpha_f(e_j)}\right) \tilde{\nu}^{sd}(e_j) \end{aligned} \tag{5}$$

Since only the second term in Eq. (5) depends on $\tilde{\nu}^{sd}$ and $\tilde{\nu}^{sd} \in \{0,1\}^{|E_l|}$, the maximum likelihood path estimate may be written quite compactly as

$$\hat{p}_{sd} = \left\{ e_j \in E_l \;\middle|\; N \log \frac{\alpha_m(e_j)}{\beta_f(e_j)} + \sum_{k=1}^{N} \nu_k^{sd}(j) \log \frac{\beta_m(e_j)\beta_f(e_j)}{\alpha_m(e_j)\alpha_f(e_j)} \geq 0 \right\} \tag{6}$$

With these in hand, we proceed to describe the feasible region of topologies.

B. Describing the Feasible Region of Adjacencies

In order to estimate the endpoints of a suspect transmission, it is necessary to have some idea of the logical topology of the network. Instead of considering the logical adjacencies implied by the actual network $G(V, E, f)$, we are concerned with adjacency relationships corresponding to only those elements utilized in the probing phase. For example, we cannot hope to pinpoint the position of a link e in the original network that is not monitored by a sensor (i.e. $e \in E - E_l$), nor can we locate a link whose sensor was not activated by any data transmission in the probing phase. To capture this notion, we define the set of *identifiable edges* E_I as the set of edges whose sensors are activated by at least one data transmission during probing:

$$E_I = \{e \in E_l \ \mid \ e \in p_{s_i d_i} \text{ for some } i\} \tag{7}$$

Note that in Eq. (7) and throughout, if sensor errors are an issue, replace p_{sd} with its maximum likelihood estimate $\hat{p}_{sd}$ as described in the previous section. The particular topology we wish to describe is then given by $G_A(V_A, E_A)$ where $V_A = E_I \cup T$ and $E_A \subseteq V_A \times V_A$. G_A may be directed or undirected, depending upon the nature of G.

We assume non-identifiable edges are essentially ’collapsed’ in the original network G. This is done by recursively assigning v_i the value of v_j for all $(v_i, v_j) \in V$ such that $(v_i, v_j) = f(e)$ for some $e \in E - E_I$ and $v_i \notin f(e)$ for any $e \in E_I$. The idea here is to assure two elements are logically adjacent in G even if they are physically separated by a link (or subgraph of links) that is not identifiable. Under this assumption, we now define what the adjacency relationships in G_A mean with regard to the original network G.

Consider first the case when G is undirected. Two edges are adjacent if they share a common endpoint vertex, that is $e_1 \in E_I \cap V_A$ is adjacent to $e_2 \in E_I \cap V_A$ if $f(e_1) \cap f(e_2) \neq \phi$. An edge and an external vertex are adjacent if the external vertex serves as one endpoint of the edge, that is $e \in E_I \cap V_A$ is adjacent to $v \in T \cap V_A$ if $v \in f(e)$. Two external vertices may not be adjacent since there must be at least one link between them.

Consider now the case when G is directed. Here we must be careful about order: x *is adjacent to* y means x can reach y when traversing the graph in the allowed direction. Edge e_1 is adjacent to edge e_2 if the incoming endpoint of e_1 is the outgoing endpoint of e_2, that is $e_1 \in E_I \cap V_A$ is adjacent to $e_2 \in E_I \cap V_A$ if $f(e_1)_2 = f(e_2)_1$. Edge e is adjacent to external vertex v if v is the incoming endpoint of e, that is $e \in E_I \cap V_A$ is adjacent to $v \in T \cap V_A$ if $v = f(e)_2$. External vertex v is adjacent to edge e if v is the outgoing endpoint of e, that is $v \in T \cap V_A$ is adjacent to $e \in E_I \cap V_A$ if $v = f(e)_1$. As before, two external vertices may not be adjacent.

We explicitly constrain the adjacency matrix A of the graph G_A based on the probing data $\{(s, d, p_{sd})_i\}$. In addition to the implications of the above discussion, there are zeros along the diagonal of A; this leads to at most $|E_I|^2 + |E_I|(2|T| - 1)$ unknown 0-1 variables to be determined (this quantity is cut in half for undirected graphs thanks to symmetry). In developing the constraints, we assume that no cycles occur in any of the measured paths. If the paths were ordered, it would be a straightforward exercise to write down adjacency relationships among elements in the path under this assumption. Because the measured paths p_{sd} are unordered, we cannot say precisely which elements are adjacent; we can only say that each element in the path must be adjacent to some other element(s) in the path.

Consider first the undirected case. Under the no cycle assumption, each measured path p_{sd} implies the following: each $e \in p_{sd}$ must be adjacent to exactly two elements from the set $\{p_{sd} - e \cup \{s, d\}\}$, $s \in T_s$ must be adjacent to one element from p_{sd}, and $d \in T_d$ must be adjacent to one element from p_{sd}. These are restated as linear constraints on the adjacency matrix A in Eq. (8).

$$\begin{aligned} &\sum_{\{j|v_j \in p_{sd} - e_i \cup \{s,d\}\}} A_{ij} = 2 \text{ for all } e_i \in p_{sd} \\ &\sum_{\{j|v_j \in p_{sd}\}} A_{i_s j} = 1 \\ &\sum_{\{j|v_j \in p_{sd}\}} A_{i_d j} = 1 \end{aligned} \tag{8}$$

An undirected graph also has a symmetric adjacency matrix, i.e. $A_{ij} = A_{ji}$ for all i, j. Thus we can solve for the upper half of the adjacency matrix only and the lower half is automatically determined. This reduces the number of variables to $\frac{1}{2}\left(|E_I|^2 + |E_I|(2|T| - 1)\right)$. If v_j in any of the index sets in Eq. (8) has $j > i$ then we simply replace that v_j with v_i where i is the corresponding element in the upper half of A.

The constraints on the adjacency matrix of a directed graph follow similarly from the no cycle assumption: one element from $\{p_{sd} - e \cup \{s\}\}$ must be adjacent to each $e \in p_{sd}$, each $e \in p_{sd}$ must be adjacent to one element from $\{p_{sd} - e \cup \{d\}\}$, $s \in T_s$ must be adjacent to one element from p_{sd}, and one element from p_{sd} must be adjacent to $d \in T_d$. These are given in Eq. (9) as constraints on the directed adjacency matrix A.

$$\begin{aligned}
&\sum_{\{i|v_i \in p_{sd} - e_j \cup \{s\}\}} A_{ij} = 1 \text{ for all } e_j \in p_{sd} \\
&\sum_{\{j|v_j \in p_{sd} - e_i \cup \{d\}\}} A_{ij} = 1 \text{ for all } e_i \in p_{sd} \\
&\sum_{\{j|v_j \in p_{sd}\}} A_{i_s j} = 1 \\
&\sum_{\{i|v_i \in p_{sd}\}} A_{i j_d} = 1
\end{aligned} \tag{9}$$

We therefore have $|p_{sd}| + 2$ linear constraints on an undirected adjacency matrix or $2|p_{sd}| + 2$ linear constraints on a directed adjacency matrix implied by each (s, d, p_{sd}) measurement. All constraints may be collected into a single system, so that the feasible region of network topologies G_A is given by $\{x|Qx = b\}$ where x is a vectorized version of the adjacency matrix of G_A and Q, b are defined by the appropriate constraints.

C. Sampling the Feasible Region of Adjacencies

It is necessary to sample adjacency matrices from the feasible region defined in the previous section. This amounts to finding several solutions to the problem

$$\begin{aligned}
&\text{find } x \in \{0,1\}^n \\
&\text{such that } Qx = b
\end{aligned} \tag{10}$$

Unfortunately, the problem in Eq. (10) is NP-complete [26]. We consider an equivalent restatement of Eq. (10)

$$\begin{aligned}
&\text{minimize } (Qx - b)^T W (Qx - b) \\
&\text{such that } x \in \{0,1\}^n
\end{aligned} \tag{11}$$

where W is a (symmetric) positive definite matrix that may be chosen to emphasize the relative importance of the different constraints. Obviously any optimal solution of the problem in Eq.

(11) with zero value solves the feasibility problem in Eq. (10). The problem in Eq. (11) is no easier than the original statement, however, it has been shown that problems of this type (0-1 quadratic programs) can be approximated quite well using a semidefinite relaxation [7].

We now proceed to derive the SDP relaxation of Eq. (11). Our relaxation is similar to the one derived in [6] for MAX2SAT. First note that the optimization in Eq. (11) is equivalent to

$$\begin{aligned}&\text{minimize } x^T D x - 2d^T x\\&\text{such that } x \in \{0,1\}^n\end{aligned} \tag{12}$$

where $D = Q^T W Q$ and $d = Q^T W b$. This is easily seen by expanding the objective in Eq. (11) and dropping the constant term. Now note that $x_i^2 = x_i$ since $x_i \in \{0,1\}$; this fact this allows Eq. (12) to be re-expressed as

$$\begin{aligned}&\text{minimize } \textstyle\sum_{i,j} D_{ij} x_i x_j - 2\sum_j d_j x_j^2\\&\text{such that } x \in \{0,1\}^n\end{aligned} \tag{13}$$

We now introduce variables $y_i \in \{-1,1\}$ for each $x_i \in \{0,1\}$ for $i = 1 \ldots N$ along with an additional $y_{n+1} \in \{-1,1\}$ so that the change of variables is given by

$$x_i = \frac{1}{2}\left(1 + y_{n+1} y_i\right) \tag{14}$$

The identities in Eq. (15) follow from this change of variables.

$$\begin{aligned}x_i x_j &= \tfrac{1}{4}\left[(1 + y_i y_j) + (1 + y_{n+1} y_i) + (1 + y_{n+1} y_j) - 2\right]\\-x_i x_j &= \tfrac{1}{4}\left[(1 - y_i y_j) + (1 - y_{n+1} y_i) + (1 - y_{n+1} y_j) - 4\right]\end{aligned} \tag{15}$$

If we introduce a negative sign in the objective, then the optimization in Eq. (13) becomes

$$\begin{aligned}&\text{maximize } \tfrac{1}{4}\textstyle\sum_{i,j}\left[B_{ij}(1 + y_i y_j) + C_{ij}(1 - y_i y_j)\right] - e^T D e\\&\text{such that } y \in \{-1,1\}^{n+1}\end{aligned} \tag{16}$$

where e is a vector of ones and matrices B, C are given by

$$\begin{aligned} B &= \begin{pmatrix} 0 & 2d \\ 2d^T & 0 \end{pmatrix} \\ C &= \begin{pmatrix} D & De \\ (De)^T & 0 \end{pmatrix} \end{aligned} \tag{17}$$

In order to obtain a semidefinite program, define the matrix $Y = yy^T$. It is simple to show that $Y = yy^T$ for some vector y if and only if $Y \succeq 0$ (i.e. Y is positive semidefinite) and $rank(Y) = 1$. We drop the nonconvex rank-1 constraint to obtain the SDP relaxation

$$\begin{aligned} &\text{maximize } Tr\left[(B - C)Y\right] \\ &\text{such that } \begin{array}{l} diag(Y) = e \\ Y \succeq 0 \end{array} \end{aligned} \tag{18}$$

where $Tr[\cdot]$ indicates the trace operation and the constraint $diag(Y) = e$ is added to enforce $y_i^2 = 1$. The equivalence of the objective functions in Eq. (18) and Eq. (16) can be seen easily by replacing $y_i y_j$ with Y_{ij} and dropping constant terms. The SDP in Eq. (18) may be solved in polynomial time using a primal-dual path following algorithm [4]. The result of this optimization Y^* will in general be a non-integer symmetric positive semidefinite matrix. In [6], a randomized rounding methodology is proposed to recover a -1,1 vector y from the SDP solution Y^*. The strategy is to first perform the Cholesky factorization $Y^* = \tilde{V}^T \tilde{V}$, then choose a random hyperplane through the origin with normal vector r. The value of y_i is then determined by whether the corresponding column $\tilde{v}_i$ of $\tilde{V}$ lies above or below the hyperplane, i.e. $y_i = 1$ if $v_i^T r \geq 0$ and $y_i = -1$ if $v_i^T r < 0$.

A direct application of the method in [6] provides a means for generating M approximate samples from the feasible region of network topologies. Simply generate M vectors $\{r^k\}_{k=1}^M$ from the uniform distribution on the set $S_n = \{x \in \mathbf{R}^{n+1} | x^T x = 1\}$. The i^{th} element of the k^{th}

vectorized adjacency sample $\hat{x}$ is then given by

$$\hat{x}_i^k = \begin{cases} 1 & \text{if } sign(v_i^T r^k) = sign(v_{n+1}^T r^k) \\ 0 & \text{if } sign(v_i^T r^k) \neq sign(v_{n+1}^T r^k) \end{cases} \tag{19}$$

This result can be seen by applying the rounding method and then using the change of variable formula given in Eq. (14).

We now proceed to derive the mean squared error $E\left[\|Q\hat{x} - b\|_W^2\right]$ of the sample adjacency in Eq. (19). First note that the rounding scheme used implies the following identities.

$$\begin{aligned} E[1 + y_i y_j] &= 2P\left(sign(v_i^T r) = sign(v_j^T r)\right) \\ E[1 - y_i y_j] &= 2P\left(sign(v_i^T r) \neq sign(v_j^T r)\right) \end{aligned} \tag{20}$$

where r is a random vector from the uniform distribution on $S_n = \{x \in \mathbf{R}^{n+1} | x^T x = 1\}$. We may evaluate the probabilities in Eq. (20) quite easily via the observation in [6]. Note that symmetry of the distribution implies $P\left(sign(v_i^T r) \neq sign(v_j^T r)\right) = 2P\left(v_i^T r \geq 0, v_j^T r < 0\right)$. And if $\theta = \arccos(v_i^T v_j)$ is the angle between the vectors v_i and v_j then it follows $P\left(v_i^T r \geq 0, v_j^T r < 0\right) = \frac{\theta}{2\pi}$ since the distribution of r is uniform on S_n. A similar argument applies to the case of matching sign. The results are summarized below.

$$\begin{aligned} P\left(sign(v_i^T r) = sign(v_j^T r)\right) &= 1 - \tfrac{1}{\pi}\arccos(v_i^T v_j) \\ P\left(sign(v_i^T r) \neq sign(v_j^T r)\right) &= \tfrac{1}{\pi}\arccos(v_i^T v_j) \end{aligned} \tag{21}$$

If we define the matrix Z such that $Z_{ij} = \arccos(Y_{ij}^*)$ where Y^* is the solution of the SDP relaxation in Eq. (18) and note that the objective function in Eq. (16) is exactly equal to $b^T W b - \|Q\hat{x} - b\|_W^2$, then we may take the expectation of the objective in Eq. (16) and apply the identities in Eqs. (20) and (21) to obtain the mean squared error as

$$E\left[\|Q\hat{x} - b\|_W^2\right] = \|Qe - b\|_W^2 - \frac{1}{2\pi} Tr\left[(C - B)Z\right] \tag{22}$$

where e is a vector of ones.

We may obtain a bound on the expected value of the squared error in Eq. (22) independent of the solution to the SDP. As in [6], define the constant α

$$\alpha = \min_{z\in[0,\pi]} \ \frac{2}{\pi}\frac{z}{1-\cos z} \tag{23}$$

From this definition of α, the following identities follow immediately

$$\begin{aligned} &\tfrac{1}{2}\alpha(1+\cos z) \leq 1 - \tfrac{1}{\pi}z \\ &\tfrac{1}{2}\alpha(1-\cos z) \leq \tfrac{1}{\pi}z \end{aligned} \tag{24}$$

We take the expected value of the objective function in Eq. (16) and apply the identities in Eq. (24) with $Z_{ij} = \arccos(Y^*_{ij})$ to give

$$b^TWb - E\left[\|Q\hat{x} - b\|^2_W\right] \geq \alpha\frac{1}{4}\left(\sum_{i,j}\left[B_{ij} + C_{ij}\right] + Tr\left[(B-C)Y^*\right]\right) - e^TDe \tag{25}$$

Now suppose the equation $Qx = b$ has at least one feasible solution x^0. Let y^0 be the corresponding -1,1 vector and $Y^0 = y^0(y^0)^T$. We then have

$$0 = \left\|Qx^0 - b\right\|^2_W = e^TDe + b^TWb - \frac{1}{4}\left(\sum_{i,j}\left[B_{ij} + C_{ij}\right] + Tr\left[(B-C)Y^0\right]\right) \tag{26}$$

But since Y^* solves the SDP in Eq. (18), it follows

$$Tr\left[(B-C)Y^*\right] \geq Tr\left[(B-C)Y^0\right] = 4e^TDe + 4b^TWb - \sum_{i,j}\left[B_{ij} + C_{ij}\right] \tag{27}$$

We may now combine the inequalities in Eqs. (25) and (27) and rearrange to obtain a bound on the expected value of the squared error that is independent of the SDP solution

$$E\left[\|Q\hat{x} - b\|^2_W\right] \leq (1-\alpha)\left(e^TDe + b^TWb\right) \tag{28}$$

In practice, the bound in Eq. (28) tends to exceed the true expected value in Eq. (22) by a large amount. However, it is of theoretical interest; since $D = Q^TWQ$, this bound indicates that the total weighted constraint violation of topology samples will tend to increase with the average path length and number of constraints. If the weighted sample error is undesirably large, a naive fix is to simply subsample (i.e. take, say, every ten adjacency samples and discard the rest).

We will return to this issue later and suggest some other fixes, since large errors in the SDP generated adjacency samples is a major obstacle to applying this method to ever larger networks. With the adjacency matrix samples in hand, we proceed to approximate the endpoint posterior of a suspect transmission.

D. Approximating the Endpoint Posterior of a Suspect Transmission

We use the network topology adjacency samples obtained in the previous section to derive an approximate endpoint posterior distribution of a suspect transmission (that is a data transmission whose source and destination are unknown). Let $p_x \subseteq E_I$ be the set of identifiable link sensors activated by the suspect transmission. Let $P_o(A)$ be a prior distribution on the adjacency matrices of the derived topology G_A. Recall from Section II-B, that G_A is the logical topology describing the connections among identifiable edges and external vertices in the original network G. Indeed, it is no small task to determine a prior distribution on the derived topology G_A given a prior on the original network topology G, since an infinite number of topologies G might correspond to the same G_A. Let $P(s,d)$ be a prior distribution that indicates the probability a particular source node will communicate with a particular destination node. We assume these probabilities are independent of the network topology A, thus $P(s,d|A) = P(s,d)$. Under this assumption, Bayes rule may be used to write the endpoint posterior $P(s,d|p_x)$ as

$$P(s,d|p_x) = \sum_A \frac{P(p_x|s,d,A)P(s,d)}{\sum_{s,d} P(p_x|s,d,A)P(s,d)} P_o(A) \tag{29}$$

Define $f_{p_x,sd}(A)$ as in Eq. (30).

$$f_{p_x,sd}(A) = \frac{P(p_x|s,d,A)P(s,d)}{\sum_{s,d} P(p_x|s,d,A)P(s,d)} \tag{30}$$

It is then clear that the endpoint posterior in Eq. (29) may be re-expressed as

$$P(s,d|p_x) = E\left[f_{p_x,sd}(A)\right] \tag{31}$$

Thus the strong law of large numbers suggests a Monte Carlo estimate of the endpoint posterior using the topology adjacency matrix samples $\{A_i\}_{i=1}^M$ given by

$$\hat{P}(s,d|p_x) = \frac{1}{\sum_{i=1}^M P_o(A_i)} \sum_{i=1}^M f_{p_x,sd}(A_i) P_o(A_i) \tag{32}$$

The estimate in Eq. (32) is reasonable provided we can compute the value of the function in Eq. (30) for each topology sample A_i. We require a model for the conditional path probabilities $P(p|s,d,A)$ for this computation. Clearly, the routing mechanism used in the network should figure prominently into any such model. We propose a model based on shortest path routing.

Let $P_{sd|A}$ be the set of all paths (of finite length) from source s to destination d in the topology A. Let $w : \mathbf{R}_+ \to \mathbf{R}_+$ be a nonincreasing function on the positive reals. We then propose $P(p|s,d,A)$ as

$$P(p|s,d,A) = \frac{1}{\gamma_{sd|A}^K} w\left(\frac{|p|}{\min\limits_{\tilde{p}\in P_{sd|A}} |\tilde{p}|}\right) I_{P_{sd|A}}(p) \tag{33}$$

where $I_A : A \to \{0,1\}$ is the indicator function and $\gamma_{sd|A}^K$ is a normalization constant. The formula in Eq. (33) essentially ensures longer paths are no more probable than shorter paths and that invalid paths have probability zero. Note that the measured paths p are unordered, so we say $p \in P_{sd|A}$ if there exists an ordering of p given by p_o such that $p_o \in P_{sd|A}$.

In order to compute the normalization constant $\gamma_{sd|A}^K$ in Eq. (33), we must sum $w\left(\frac{|p|}{\min\limits_{\tilde{p}\in P_{sd|A}} |\tilde{p}|}\right)$ over all $p \in P_{sd|A}$, which generally requires too much computational effort. Instead, we normalize just over the set $P_{sd|A}^K \subseteq P_{sd|A}$ consisting of the K shortest loopless paths between s and d in the network topology A. This set can be computed in $O(Kn^3)$ time for a network with n nodes using an algorithm described in [9]. The normalization is thus

$$\gamma_{sd|A}^K = \sum_{p\in P_{sd|A}^K} w\left(\frac{|p|}{\min\limits_{\tilde{p}\in P_{sd|A}} |\tilde{p}|}\right) \tag{34}$$

We now have all of the necessary ingredients to compute the endpoint posterior distribution estimate given in Eq. (32).

E. Estimating the Endpoints of a Suspect Transmission

We may give maximum a posteriori (MAP) estimates of the endpoints (s_x, d_x) of a suspect transmission p_x after computing the posterior distribution estimate in Eq. (32). Indeed, the MAP estimate is simply given by

$$(\hat{s}_x, \hat{d}_x) = \arg\max_{(s,d)} \hat{P}(s, d|p_x) \tag{35}$$

MAP estimates of s_x or d_x individually are given by maximizing the appropriate marginal $P(s|p_x)$ or $P(d|p_x)$ respectively.

We use as an error measure the ratio $\Lambda_{sd}(p_x)$ below for the estimated endpoints (s_x, d_x).

$$\Lambda_{sd}(p_x) = \frac{\max\limits_{(s,d)} \hat{P}(s, d|p_x)}{\max\limits_{(s,d)} \hat{P}(s, d|p_x) + \max\limits_{(s,d)\neq(\hat{s}_x,\hat{d}_x)} \hat{P}(s, d|p_x)} \tag{36}$$

It is also useful to compute the corresponding ratios associated with the marginalized distributions $\Lambda_s(p_x)$ and $\Lambda_d(p_x)$, as it may be the case that either the source or destination of a suspect transmission is more accurately determined individually than are both collectively. These are given by

$$\Lambda_s(p_x) = \frac{\max\limits_{s} \hat{P}(s|p_x)}{\max\limits_{s} \hat{P}(s|p_x) + \max\limits_{s\neq\hat{s}_x} \hat{P}(s|p_x)} \tag{37}$$

$$\Lambda_d(p_x) = \frac{\max\limits_{d} \hat{P}(d|p_x)}{\max\limits_{d} \hat{P}(d|p_x) + \max\limits_{d\neq\hat{d}_x} \hat{P}(d|p_x)} \tag{38}$$

It is clear that the ratios in Eqs. (36), (37), and (38) must lie in the interval $[\frac{1}{2}, 1]$. Larger values of these ratios in a sense indicate more 'confidence' in the associated MAP estimates since a value of 1 is achieved only when all of the mass of the estimated posterior distribtution is concentrated at the MAP estimate.

F. Algorithm Complexity

We now analyze the complexity of the source/destination estimation scheme developed here and show that producing the topology samples is the most computationally demanding step.

Complexity results are given in terms of the number of (s, d) pairs used in the probing phase–let n denote this number. We assume the number of hops h required for a message to reach its destination starting from the source remains constant with increased problem size. Note this is a reasonable assumption for real networks due to the well known *small world* effect [27], [20].

First consider the size of a problem with n (s, d) pairs used in probing. Since the number of external nodes $|T|$ satisfies $|T| \leq 2n$ and the number of identifiable edges $|E_I|$ satisfies $|E_I| \leq hn$ we have both of these values are $O(n)$. Thus, the number of 0-1 variables associated with the adjacency matrix of $G_A(E_I \cup T, E_A)$ is $O(n^2)$ (recall the number of such variables is proportional to $|E_I|^2 + |E_I|(2|T| - 1)$).

We use the SDP relaxation in Eq. (18) to produce sample topologies that are approximately consistent with the probing measurements. Typically interior point methods are used to solve SDP's to within ϵ of the optimal solution. These are based on Newton's method; therefore at each iteration it is necessary to solve a linear system of equations for the Newton directions ($O(m^3)$ for a system of size m). An algorithm given in [28] is shown to take $O(|\log \epsilon|\sqrt{m})$ iterations for a problem of size m–this performance is typical for all interior point algorithms. Our problem has dimension $O(n^2)$, thus solving the SDP takes $O((n^2)^{3.5})$ or $O(n^7)$ time. A Cholesky factorization is then performed on the SDP solution, which takes $O((n^2)^3)$ or $O(n^6)$ time. The topology samples are then produced by generating M random vectors and taking inner products. The time required for each inner product is linear in the size of the problem; it follows that this step takes $O(n^2)$ time.

In order to compute the approximate posterior distribution in Eq. (32), we must evaluate the functional $f_{p_x,sd}$ in Eq. (30) at each topology sample A_i. The only costly step here is computing the shortest path(s) needed for the path likelihood model in Eq. (33). We are computing shortest paths in $G_A(E_I \cup T, E_A)$, which has $O(n)$ nodes; thus it takes $O(n^3)$ time to compute $P(p_x|s, d, A)$ for each topology sample A_i using the algorithm in Eq. ([9]). Computation of $f_{p_s,sd}$ requires $P(p_x|s, d, A)$ for all n (s, d) pairs. We then have $O(n^4)$ time required for computing the approximate endpoint posteriors once the samples $\{A_i\}$ are provided. Note that this can be reduced to $O(n)$ time if the function w in Eq. (33) is taken as a constant since the normalization

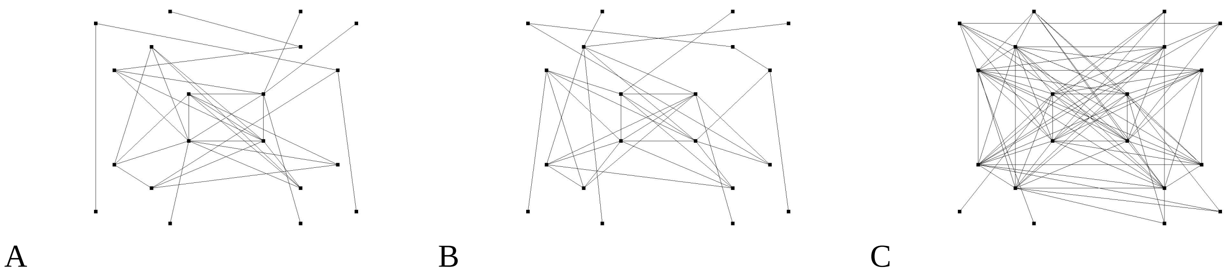

Fig. 3. Example 20-node power law random graphs with parameter $a = 1.3$ (as in Eq. (39). The method is illustrated by simulating on topologies of this type. We assume in one case that sensors are placed on 100% of the links (100% sensor coverage) and in another that sensors are placed on not fewer than 75% of the links (75% sensor coverage). 12 of the 20 nodes are selected as external nodes T: 6 of these are taken as sources T_s and 6 are taken as destinations T_d. 18 of the 36 distinct (s, d) pairs are randomly selected for use in the probing phase, denoted L. The remaining 18 pairs are denoted L^c. During the test phase suspect transmissions are passed between all 36 possible (s, d) pairs. Shortest path routing is used to determine the transmission path.

factors $\gamma_{sd|A}^{K}$ cancel out in Eq. (30) and therefore do not need to be computed.

Our algorithm would benefit greatly from speedy SDP algorithms as solving the SDP relaxation takes the most time $O(n^7)$. A parallel implementation of an interior point algorithm for SDP's might reduce the time requirements if multiple processors are available [29].

III. Simulations

We performed some numerical simulations to demonstrate the utility of the method described in this paper. We used 20-node power law random graphs for the networks to be monitored. Each 20-node graph is generated by choosing a degree value k_i for each node from the power law distribution given by

$$P(k) = \begin{cases} \frac{k^{-a}}{\zeta(a)} & \text{if } k \geq 1 \\ 0 & \text{otherwise} \end{cases} \tag{39}$$

with parameter a selected to be 1.3 and ζ is the Riemann zeta function. The i^{th} vertex is given k_i partial edges, and then pairs of partial edges are selected at random to connect and form an edge. This method for generating a random graph with specified vertex degree distribution is referred to as the configuration model [20]. We slightly modified the method to prohibit multiple connections between a single pair of vertices. Also, we rejected disconnected graphs. Some example 20-node power law random graphs with parameter $a = 1.3$ are given in Fig. 3.

After generating a sample network, we randomly selected 6 source nodes T_s and 6 destination nodes T_d. 18 of the 36 distinct source/destination pairs were then chosen at random from $T_s \times T_d$ for use in the probing phase; this set of pairs is denoted by $L \subset T_s \times T_d$. The remaining 18 pairs in $T_s \times T_d$ are denoted $L^c = (T_s \times T_d) - L$. We used shortest path routing to determine which sensors were activated by each of the 18 data transmissions in L during the probing simulation. Sensors were assumed to be perfectly accurate. We considered two situations for sensor placement. First, we assumed sensors were present on every edge in the graph (i.e. 100% sensor coverage); then we placed sensors on a random subset containing not fewer than 75% of the edges (i.e. 75 % sensor coverage). The number of identifiable edges $|E_I|$ was noted for each graph.

We described the feasible region and formulated the semidefinite programming relaxation in Eq. (18) with the weight matrix W taken as the identity. The relaxation was solved with a predictor-corrector path following algorithm given in [4]. A publicly available C implementation of this algorithm was used [30]. The randomized rounding method was applied to the solution of the SDP relaxation to produce 100 sample adjacencies. The average squared sample error $\frac{1}{M}\sum_k \left\|Q\hat{x}^k - b\right\|^2$ and theoretical mean squared error $E\left[\|Q\hat{x} - b\|^2\right]$ of Eq. (22) were noted for each graph.

Suspect transmissions were then generated using each of the 36 (s,d) pairs from $T_s \times T_d$ as endpoints by noting the identifiable edges passed in the shortest path routes. Note that shortest path routing is consistent with the likelihood model given in Eq. (33). These transmissions were identified as being between (s,d) pairs in either L or L^c. Paths that did not intercept any identifiable sensors were excluded; thus we denote by $L_I \subseteq L$ the set of (s,d) pairs used in probing whose transmission activated at least one identifiable sensor. Similarly $L_I^c \subseteq L^c$ is the set of (s,d) pairs not used in probing whose transmission activated at least one identifiable sensor. Note that for 100% sensor coverage, $|L_I| = 18$ always but $|L_I^c|$ may be less than 18. For 75% sensor coverage both $|L_I|$ and $|L_I^c|$ may be less than 18.

For each suspect transmission, we computed the approximate posterior distribution $\hat{P}(s,d|p_x)$ with uniform distributions for both of the priors $P_o(A)$ and $P(s,d)$. Also, a weight function

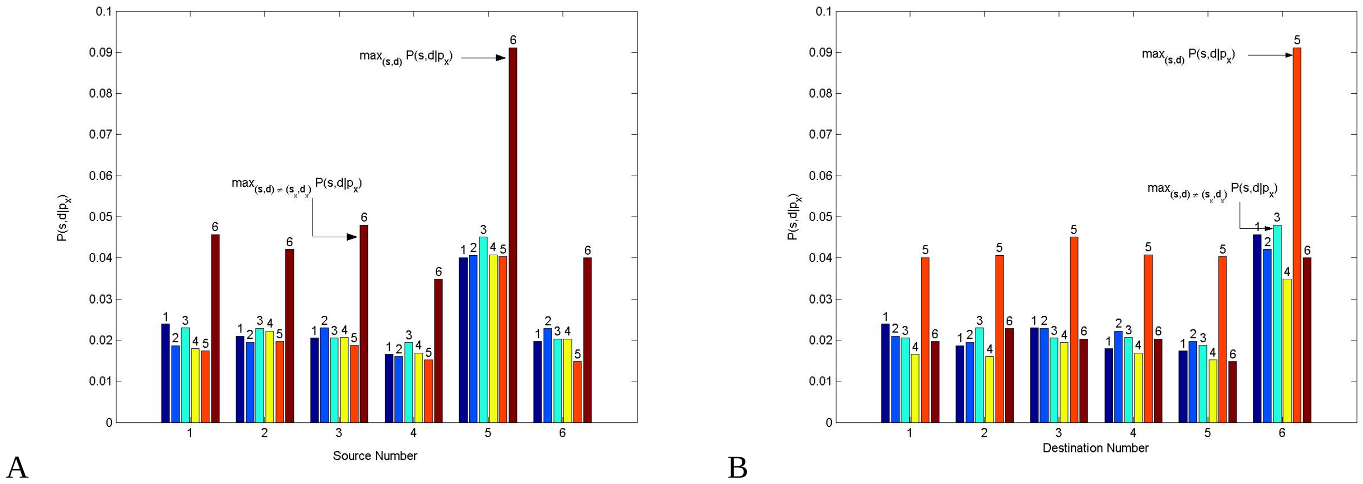

Fig. 4. Example endpoint posterior distribution $P(s,d|p_x)$ for a suspect transmission p_x with endpoints $(s_x,d_x) \in L_I$. In plot A, the probabilities are grouped by source, with each of 6 bars in a group corresponding to a different destination (noted above the individual bar). Plot B displays the same information except probabilities are grouped by destination with source number noted above each individual bar. The largest and second largest values of the posterior are indicated–it is these values that are used in computing the Λ-ratio of Eq. (36), calculated as $\Lambda_{sd}(p_x) = 0.655$. It is clear in this example that the endpoints of this transmission (source number 5 and destination number 6) will be correctly estimated by the collective MAP estimate.

of $w(x) = 1$ was used in computation of the conditional path probabilities $P(p|s,d,A)$ so that the normalization constants $\gamma^K_{sd|A}$ cancelled out. From these, we computed the MAP estimates collectively using the joint distribution $\hat{P}(s,d|p_x)$ and individually using the appropriate marginal $\hat{P}(s|p_x)$ or $\hat{P}(d|p_x)$. We noted average values of each of the Λ-ratios in Eqs. (36), (37), and (38) averaged separately over sets L_I and L^c_I for each graph. An example endpoint posterior distribution for a suspect path with endpoints in L_I is shown in Fig. 4. The marginals of this distribution are shown in Fig. 5.

We repeated the simulation procedure for 30 undirected power law random networks and recorded the values of interest. Table I shows the number of identifiable edges, fraction of the total number of edges that are identifiable, cardinality of the set L^c_I, and topology sample error data for undirected graphs with 100% sensor coverage. Table II gives the same data along with the cardinality of the set L_I for undirected graphs with 75% sensor coverage (note that $|L_I| = 18$ always for graphs with 100% sensor coverage).

It is apparent that the theoretical expected value of the squared error, and accordingly the average squared sample error, tend to be lower for graphs with 75% sensor coverage. This is

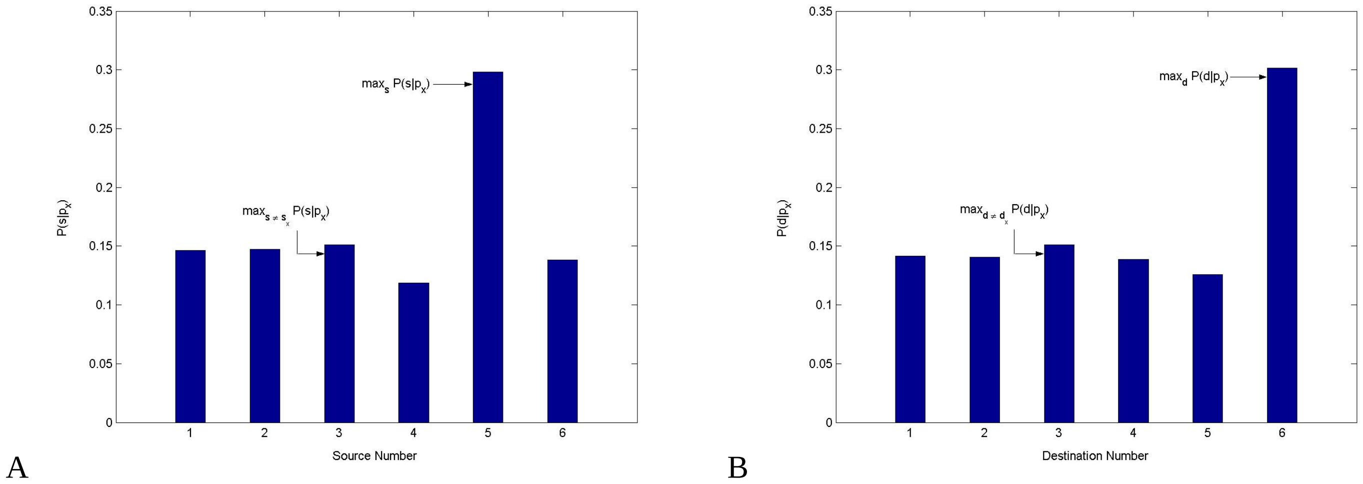

Fig. 5. Marginal distributions ($P(s|p_x)$ in A and $P(d|p_x)$ in B) associated with the example endpoint posterior distribution shown in Fig. 4. The largest and second largest values of the marginal posteriors are indicated–it is these values that are used in computing the Λ-ratios of Eqs. (37) and (38), calculated as $\Lambda_s(p_x) = 0.663$ and $\Lambda_d(p_x) = 0.666$. It is clear in this example that the endpoints of this transmission (source number 5 and destination number 6) will be correctly estimated by the individual MAP estimates as well.

consistent with our earlier comments (see section II-C) since these graphs would tend to have shorter path lengths, simply because there are fewer identifiable edges. However, fewer suspect transmissions actually intercept any of our sensors in the 75% coverage case (indicated by smaller values of $|L_I|$ and $|L_I^c|$).

Plots of proportion of endpoint estimates correct for a given set (L_I or L_I^c) versus the ratios from Eqs. (36)-(38) averaged over the corresponding set for the undirected graphs with 100% sensor coverage are shown in Fig. 6. Corresponding plots for the undirected graphs with 75% sensor coverage are shown in Fig. 7. Plots are shown for collective estimates of (s_x, d_x) via the joint distribution as well as for individual estimates of s_x and d_x from the marginals.

In Figs. 6 and 7, we observe an approximately linear relation between the proportion of correct estimates and the appropriate Λ ratio when the Λ ratio exceeds 0.58 and 0.60 respectively. In this regime, the Λ ratio might be used as a measure of confidence in the endpoint estimates. Also note that transmissions in set L_I tend to have higher Λ ratios (and are correct more often) than those in set L_I^c because it is the transmissions in set L_I (those with endpoints used in probing) that actually determine the constraints from which the topology samples are generated. Note that the data in set L_I^c often has incorrect joint estimation of source and destination, however,

Graph	$\lvert E_I\rvert$	$\frac{\lvert E_I\rvert}{\lvert E\rvert}$	$\lvert L_I^c\rvert$	$\frac{1}{M}\sum_k \left\Vert Q\hat{x}^k - b\right\Vert^2$	$E\left[\Vert Q\hat{x} - b\Vert^2\right]$
1	18	0.35	16	7.28	6.97
2	18	0.33	14	0.1	0.07
3	19	0.40	16	1.8	1.79
4	19	0.33	16	4.2	4.35
5	18	0.42	16	2.2	2.42
6	19	0.32	15	3.24	3.16
7	20	0.42	15	2.5	2.36
8	19	0.40	16	0.04	0.13
9	17	0.37	15	0.24	0.24
10	18	0.31	13	2.14	2.00
11	22	0.45	18	11.28	10.84
12	18	0.45	18	3.34	3.43
13	17	0.31	14	2.16	1.65
14	18	0.24	10	0.08	0.05
15	19	0.22	10	0.06	0.05
16	17	0.35	14	0.06	0.08
17	20	0.31	18	0.22	0.29
18	19	0.31	10	0	0.01
19	17	0.45	16	0.34	0.32
20	20	0.36	17	5.5	5.18
21	19	0.49	18	16.36	16.93
22	23	0.59	14	12.22	12.86
23	21	0.43	15	8.14	8.60
24	21	0.41	17	0.02	0.14
25	23	0.40	17	8.84	8.86
26	21	0.49	15	11.74	11.47
27	17	0.50	18	0.34	0.39
28	14	0.24	12	0	0.01
29	20	0.38	14	8.3	8.39
30	22	0.48	15	3.64	3.77

TABLE I

Simulation results for the undirected graphs with 100% sensor coverage. $|E_I|$ is the number of identifiable edges (those activated during probing), $|E|$ is the total number of edges in the network, and $|L_I^c|$ is the number of (s,d) pairs not used in probing whose transmission activated at least one identifiable sensor–here $|L_I| = 18$ as there is 100% sensor coverage. $|L_I^c|$ tends to be smaller when a smaller fraction of the edges are identifiable, as expected. Note also that the sample and ensemble means for the topology samples $\hat{x}$ agree quite well. The sample mean is computed over 100 samples.

Graph	$\lvert E_I\rvert$	$\frac{\lvert E_I\rvert}{\lvert E\rvert}$	$\lvert L_I\rvert$	$\lvert L_I^c\rvert$	$\frac{1}{M}\sum_k \left\lVert Q\hat{x}^k - b\right\rVert^2$	$E\left[\lVert Q\hat{x} - b\rVert^2\right]$
1	15	0.29	17	14	6.82	7.21
2	12	0.32	13	13	0.46	0.12
3	16	0.25	17	9	0.12	0.08
4	13	0.20	17	11	0	0.00
5	18	0.37	18	16	0	0.00
6	13	0.19	15	10	0.12	0.11
7	16	0.44	16	16	3.5	3.55
8	13	0.29	18	18	0.04	0.07
9	16	0.39	17	11	3.74	3.40
10	16	0.25	14	10	5.1	5.12
11	14	0.30	17	14	0.1	0.12
12	17	0.39	18	16	0.16	0.11
13	16	0.32	15	10	0.04	0.12
14	19	0.51	18	15	16.88	15.89
15	15	0.42	14	14	1.6	1.79
16	14	0.37	16	13	0.16	0.15
17	13	0.31	17	17	0.3	0.28
18	18	0.38	18	16	3.86	3.50
19	16	0.43	15	18	4.26	4.25
20	19	0.25	18	14	0.12	0.17
21	13	0.21	14	9	0.16	0.12
22	14	0.29	18	13	0.5	0.27
23	18	0.27	17	15	1.86	1.93
24	14	0.21	16	13	0	0.01
25	14	0.23	18	10	0.02	0.03
26	19	0.44	18	14	5.36	5.42
27	16	0.31	16	12	0.64	0.47
28	19	0.35	18	14	2.3	2.10
29	12	0.24	16	11	0.04	0.06
30	15	0.39	15	14	2.06	1.97

TABLE II

SIMULATION RESULTS FOR THE UNDIRECTED GRAPHS WITH 75% SENSOR COVERAGE. $|E_I|$ IS THE NUMBER OF IDENTIFIABLE EDGES (THOSE ACTIVATED DURING PROBING), $|E|$ IS THE TOTAL NUMBER OF EDGES IN THE NETWORK, $|L_I|$ IS THE NUMBER OF (s,d) PAIRS USED IN PROBING WHOSE TRANSMISSION ACTIVATED AT LEAST ONE IDENTIFIABLE SENSOR, AND $|L_I^c|$ IS THE NUMBER OF PAIRS NOT USED IN PROBING WHOSE TRANSMISSION ACTIVATED AT LEAST ONE IDENTIFIABLE SENSOR. $|L_I^c|$ TENDS TO BE SMALLER WHEN A SMALLER FRACTION OF THE EDGES ARE IDENTIFIABLE, AS EXPECTED. NOTE ALSO THAT THE SAMPLE AND ENSEMBLE MEANS FOR THE TOPOLOGY SAMPLES $\hat{x}$ AGREE QUITE WELL. THE SAMPLE MEAN IS COMPUTED OVER 100 SAMPLES.

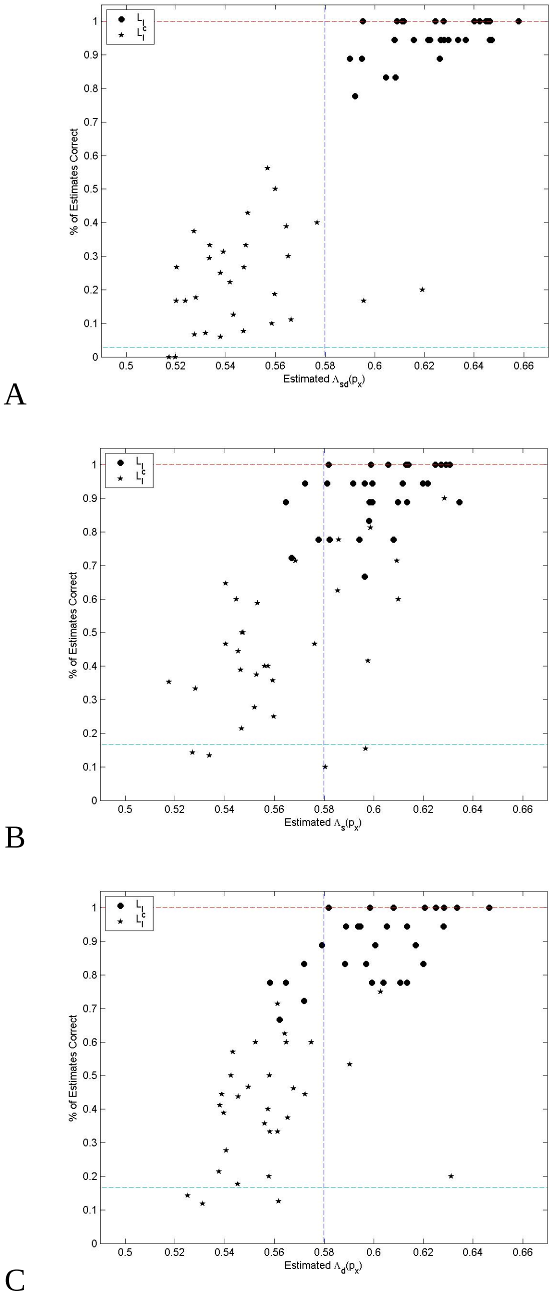

Fig. 6. Plots of proportion of endpoint estimates correct for a given set (L_I or L_I^c) versus the ratios from Eqs. (36)-(38) averaged over the corresponding set for the undirected graphs with 100% sensor coverage. Circles indicate averages over paths from set L_I and pentagrams indicate averages over paths from set L_I^c. Plot A is for collective estimation of (s_x, d_x) from joint distribution $P(s, d|p_x)$; the ratio from Eq. (36) is used. Plot B is for individual estimation of s_x from marginal distribution $P(s|p_x)$; the ratio from Eq. (37) is used. Plot C is for individual estimation of d_x from marginal distribution $P(d|p_x)$; the ratio from Eq. (38) is used. Some reference lines are also plotted. The chance line for randomly selecting endpoints is drawn in each plot (1/36 for collective estimation and 1/6 for individual estimation). Note that above $\Lambda(p_x) = 0.58$, an approximately linear behavior is observed. This behavior is somewhat washed out for the marginalized estimates, however marginalizing tends to increase the percent of correct estimates.

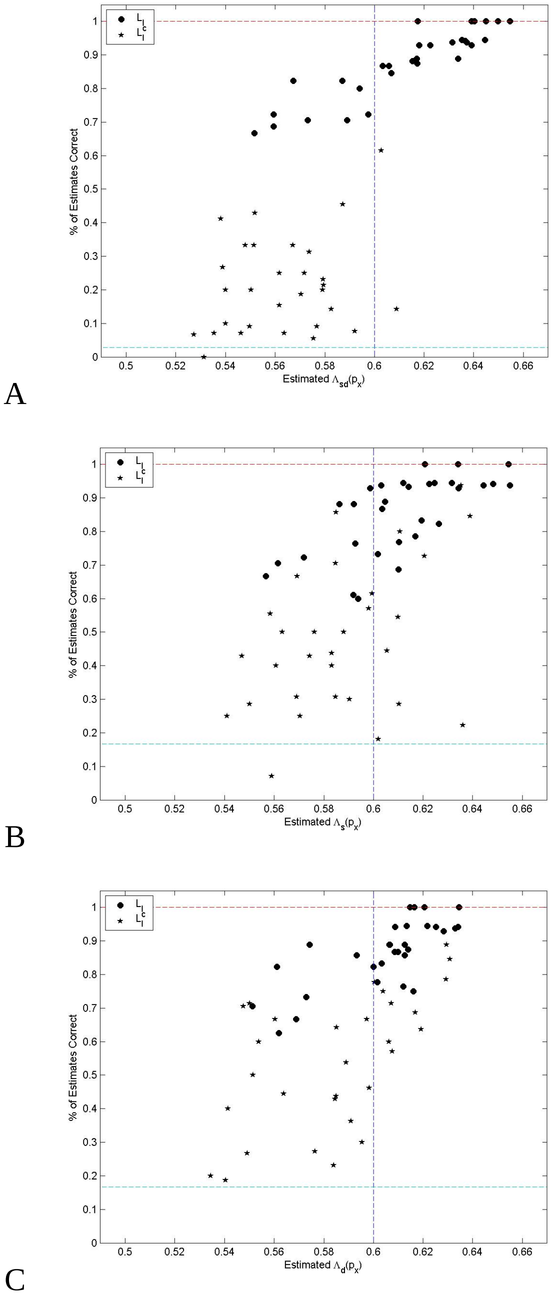

Fig. 7. Plots of proportion of endpoint estimates correct for a given set (O or N) versus the ratios from Eqs. (36)-(38) averaged over the corresponding set for the undirected graphs with 75% sensor coverage. Circles indicate averages over paths from set O and pentagrams indicate averages over paths from set N. Plot A is for collective estimation of (s_x, d_x) from joint distribution $P(s, d|p_x)$; the ratio from Eq. (36) is used. Plot B is for individual estimation of s_x from marginal distribution $P(s|p_x)$; the ratio from Eq. (37) is used. Plot C is for individual estimation of d_x from marginal distribution $P(d|p_x)$; the ratio from Eq. (38) is used. Some reference lines are also plotted. The chance line for randomly selecting endpoints is drawn in each plot (1/36 for collective estimation and 1/6 for individual estimation). Note that above $\Lambda(p_x) = 0.60$, an approximately linear behavior is observed. As with 100% coverage, this behavior is not as clear for the marginalized estimates, however marginalizing often increases the percent of correct estimates. It is not surprising that there appears to be some degradation in the quality of the estimates when only 75% of the links are equipped with sensors.

marginalized estimates of each of these individually tends to be better. Marginalization certainly blurs the linear relation in the higher confidence regime. We also observe some degradation in the quality of the estimates when only 75% of the links are equipped with sensors; this is to be expected though. Recall that these results are obtained with completely random placement of sensors and random choices for the (s, d) pairs to use in the probing phase. These two factors will clearly affect the estimates of suspect transmission endpoints, and therefore provide an interesting direction for future work.

IV. Summary and Extensions

In this paper, we have developed a methodology for estimating the endpoints of a suspect transmission in a network using link-level transmission interceptions. It is possible to envision applications of the method in all sorts of networks, or systems with key features modeled by networks. We have displayed simulations of its utility on some power law random graphs. We now discuss some possibilities for future work on this problem.

A key ingredient of the method is the network topology samples provided by rounding the solution to the semidefinite program. There are some computational scaling issues associated with this approach. As mentioned previously, the quality of the topology samples (as measured by the weighted error in Eq. (11)) tends to degrade for larger problems. Also, the computational effort necessary to solve the SDP relaxation is $O(n^7)$ for n (s, d) probing pairs and therefore may be prohibitive for large problems.

If it is possible to identify loosely connected clusters in the network based on the probing measurements p_{sd}, then one might remedy the scaling problem by solving several smaller SDP's to generate topology samples for each cluster. After mapping each cluster, a final SDP may be solved to determine how they are connected (hopefully this too would be small compared to the SDP generated by constraints on the network as a whole). At this point, it is not clear how one might identify such clusters in general. Suppose $E_c \subset E_l$ and $T_c \subset T$ represent the links and external nodes (respectively) belonging to one particular cluster. Then for all $u, v \in T_c$ it is likely the case that $p_{uv} \subseteq E_c$, i.e. the path between nodes in a cluster does not leave the cluster.

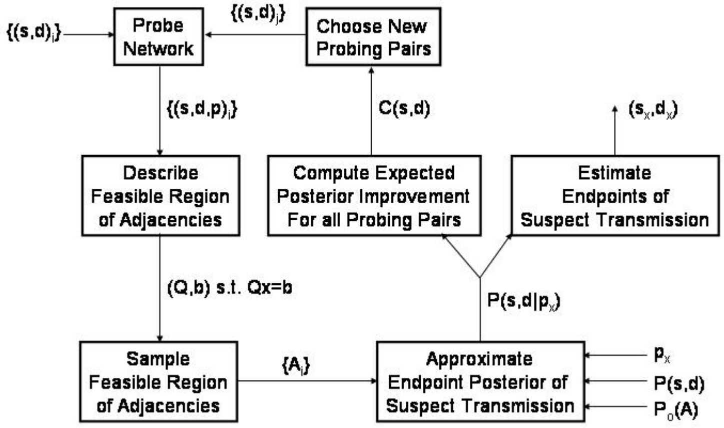
{(s,d)$_i$}
Probe Network
{(s,d)$_j$}
Choose New Probing Pairs
{(s,d,p)$_i$}
C(s,d)
(s$_x$,d$_x$)
Describe Feasible Region of Adjacencies
Compute Expected Posterior Improvement For all Probing Pairs
Estimate Endpoints of Suspect Transmission
(Q,b) s.t. Qx=b
P(s,d|p$_x$)
Sample Feasible Region of Adjacencies
{A$_i$}
Approximate Endpoint Posterior of Suspect Transmission
p$_x$
P(s,d)
P$_0$(A)

Fig. 8. Diagram of the adaptive transmission endpoint estimation system with feedback of new (s, d) pairs to probe.

This observation might be a useful certificate for the existence of such a decomposition of the network topology.

Another interesting direction for future work would be to develop an adaptive probing scheme. It is obvious that the quality of endpoint estimates for suspect transmissions will depend on which endpoints were used in the probing phase. This is quite visible in the simulation results of the previous section. The idea here is to use the approximate endpoint posterior distributions $\hat{P}(s, d|p)$ to suggest additional external node pairs (s, d) that should be probed in order to improve the estimates. The diagram for such a system is shown in Fig. 8.

One can hypothesize various criteria for determining the new probing pairs. For example, nodes that tend to have similar posterior probabilities over several suspect paths might be selected for probing so as to distinguish them more explicitly in the constraints. Also new probing pairs might be selected so as to generate constraints that reduce variablility in the pairwise graph edit distance between topology samples, thus addressing uncertainties in the network topology [25]. An information gain optimization could be used to incorporate both of these facets [31]. The question of efficient online implementation naturally arises in this context. Since solving the SDP generated by the constraints is an expensive operation, one would want to consider ’warm start' methods whereby the old SDP solution is used as a starting point to find the optimal solution with the additional constraints added. These represent some interesting extensions of the solution presented here.

ACKNOWLEDGEMENTS

This work was partially supported by a Dept. of EECS Graduate Fellowship to the first author and by the National Science Foundation under ITR contract CCR-0325571.

REFERENCES

[1] C. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*. Englewood Cliffs, NJ: Prentice Hall, Inc., 1982.

[2] L. Lovasz and A. Schrijver, "Cones of matrices and set-functions and 0-1 optimization," *SIAM Journal on Optimization*, vol. 1, no. 2, pp. 166–190, 1991.

[3] R. Saigal, *Linear Programming: A Modern Integrated Analysis*. Boston: Kluwer Academic Publishers, 1995.

[4] C. Helmberg, F. Rendl, R. Vanderbei, and H. Wolkowicz, "An interior-point method for semidefinite programming," *SIAM J. Optimization*, vol. 6, no. 2, pp. 342–361, May 1996.

[5] F. Alizadeh, "Interior point methods in semidefinite programming with applications to combinatorial optimization," *SIAM Journal on Optimization*, vol. 5, no. 1, pp. 13–51, 1995.

[6] M. Goemans and D. Williamson, "Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming," *Journal of the ACM*, vol. 42, no. 6, pp. 1115–1145, Nov. 1995.

[7] M. Goemans and F. Rendl, "Semidefinite programming in combinatorial optimzation," in *Handbook of Semidefinite Programming*, H. Wolkowicz, R. Saigal, and L. Vandenberghe, Eds. Kluwer Academic Publishers, 2000, pp. 343–360.

[8] C. Helmberg, "Fixing variables in semidefinite relaxations," *SIAM J. Matrix Anal. and Apps.*, vol. 21, no. 3, pp. 952–969, 2000.

[9] J. Yen, "Finding the k shortest loopless paths in a network," *Management Science*, vol. 17, no. 11, pp. 712–716, July 1971.

[10] Y. Vardi, "Network tomography: estimating the source-destination traffic intensities from link data," *J. Amer. Stat. Assoc.*, vol. 91, pp. 365–377, 1996.

[11] G. Liang and B. Yu, "Maximum pseudo likelihood estimation in network tomography," *IEEE Transactions on Signal Processing*, vol. 51, no. 8, pp. 2043–2053, Aug. 2003.

[12] M. Shih and A. Hero, "Unicast-based inference of network link delay distributions with finite mixture models," *IEEE Transactions on Signal Processing*, vol. 51, no. 8, pp. 2219–2228, Aug. 2003.

[13] Y. Tsang, M. Coates, and R. Nowak, "Network delay tomography," *IEEE Transactions on Signal Processing*, vol. 51, no. 8, pp. 2125–2136, Aug. 2003.

[14] R. Caceres, N. Duffield, J. Horowitz, and D. Towsley, "Multicast-based inference of network-internal loss characteristics," *IEEE Transactions on Information Theory*, vol. 45, no. 7, pp. 2462–2480, 1999.

[15] M. Coates, R. Castro, and R. Nowak, “Maximum likelihood network topology identification from edge-based unicast measurements,” *ACM Sigmetric 2002*, June 2002.

[16] N. Duffield, J. Horowitz, F. L. Presti, and D. Towsley, “Multicast topology inference from measured end-to-end loss,” *IEEE Transactions on Information Theory*, vol. 48, pp. 26–45, Jan 2002.

[17] M. Coates, A. Hero, R. Nowak, and B. Yu, “Internet tomography,” *IEEE Signal Processing Magazine*, vol. 19, no. 3, pp. 47–65, May 2002.

[18] J. Treichler, M. Larimore, S. Wood, and M. Rabbat, “Determining the topology of a telephone system using internally sensed network tomography,” *Proc. of 11th Digital Signal Processing Workshop*, Aug. 2004.

[19] M. Rabbat and R. Nowak, “Telephone network topology inference,” Univ. of Wisconsin, Madison, WI, Tech. Rep., Dec 2004.

[20] M. Newman, “The structure and function of complex networks,” *SIAM Review*, vol. 45, pp. 167–256, 2003.

[21] E. Moore, “Gedanken-experiments on sequential machines,” in *Automata Studies, Annals of Mathematics Studies*. Princeton, N.J.: Princeton University Press, 1956, no. 34, pp. 129–153.

[22] D. Lee and M. Yannakakis, “Principles and methods of testing finite state matchines - a survey,” *Proc. of the IEEE*, vol. 84, no. 8, pp. 1090–1123, Aug. 1996.

[23] M. Faloutsos, P. Faloutsos, and C. Faloutsos, “On power-law relationships of the internet topology,” *Proc. of ACM SIGCOMM’99*, pp. 251–262, Aug. 1999.

[24] M. Molloy and B. Reed, “A critical point for random graphs with a given degree sequence,” *Random Structures and Algorithms*, vol. 6, pp. 161–179, 1995.

[25] D. Justice and A. Hero, “A linear formulation of the graph edit distance for graph recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, submitted January, 2005.

[26] M. Garey and D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. San Francisco, CA: W.H. Freeman, 1979.

[27] S. Milgram, “The small world problem,” *Psychology Today*, vol. 2, pp. 60–67, 1967.

[28] C.-J. Lin and R. Saigal, “A predictor corrector method for semidefinite linear programming,” Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor, MI, Tech. Rep. TR95-20, Oct. 1995.

[29] M. Nayakkankuppam and Y. Tymofyeyev, “A parallel implementation of the spectral bundle method for large-scale semidefinite programs,” *Proc. of the 8th SIAM Conf. on App. Lin. Alg.*, 2003.

[30] B. Borchers, “Csdp: A c library for semidefinite programming,” *Optimization Methods and Software*, vol. 11, no. 1, pp. 613–623, 1999.

[31] C. Kreucher, K. Kastella, and A. Hero, “Sensor management using relevance feedback learning,” *IEEE Transactions on Signal Processing*, submitted 2003.