Identifying Infection Sources and Regions in Large Networks

Wuqiong Luo, *Student Member, IEEE*, Wee Peng Tay, *Member, IEEE*, and Mei Leng, *Member, IEEE*

***Abstract*—Identifying the infection sources in a network, including the index cases that introduce a contagious disease into a population network, the servers that inject a computer virus into a computer network, or the individuals who started a rumor in a social network, plays a critical role in limiting the damage caused by the infection through timely quarantine of the sources. We consider the problem of estimating the infection sources and the infection regions (subsets of nodes infected by each source) in a network, based only on knowledge of which nodes are infected and their connections, and when the number of sources is unknown *a priori*. We derive estimators for the infection sources and their infection regions based on approximations of the infection sequences count. We prove that if there are at most two infection sources in a geometric tree, our estimator identifies the true source or sources with probability going to one as the number of infected nodes increases. When there are more than two infection sources, and when the maximum possible number of infection sources is known, we propose an algorithm with quadratic complexity to estimate the actual number and identities of the infection sources. Simulations on various kinds of networks, including tree networks, small-world networks and real world power grid networks, and tests on two real data sets are provided to verify the performance of our estimators.**

***Index Terms*—Infection graphs, inference algorithms, security, sensor networks, social networks, source estimation.**

I. Introduction

With rapid urbanization and advancements in transportation technologies, the world has become more interconnected. A contagious disease like Severe Acute Respiratory Syndrome (SARS) can spread quickly through a population and lead to an epidemic [1]. It is crucial to quickly identify the index cases of a contagious disease since it allows us to study the causes, and hence facilitate the search for antiviral drugs and efficacious therapies. Moreover, by inferring the the set of individuals infected by each source, potential containment policies can be formulated to prevent further spreading of the disease due to new index cases [2], [3]. In a similar vein, a computer virus on a few servers of a computer network can quickly spread to other servers or computers in the network. Without prompt identification and isolation of the source servers, significant damage can result [4], [5]. Identifying the servers in the network that are first infected also allows us to detect the latent points of weaknesses in the computer network so that preventive measures can be taken to enhance the protection at these points. The source identification problem also arises in the study of rumor spreading in a social network. A rumor started by a few individuals can spread quickly through the underlying social network [6]–[9]. In many cases, we are interested to find the sources of the rumor. For example, law enforcement agencies may be interested in identifying the perpetrators who fabricate false information to manipulate the market prices of certain stocks.

We can model all the above examples as an infection spreading in a network of nodes. In a population network, the infection is the disease that is transmitted between individuals. In the example of a computer virus spreading in a network, the infection is the computer virus, while for the case of a rumor spreading in a social network, the infection is the rumor. We consider the problem of estimating the infection sources in a network of infected nodes. We are interested in the scenario where the only given information is the set of infected nodes and their connections. This is because typically, complete data about the infection spreading process, like the first times when the infection is detected at each node, is not available. Even when such detection times are available, the naive method of declaring the first detected node in the network as the sole infection source is often incorrect, as the infection may have a random dormant period, the length of which varies from node to node. For example, the spreading of a disease in a population with individuals having varying degrees of resistance, and hence exhibiting symptoms not necessarily in the order in which they are infected, presents such a problem. Our goal is to construct estimators for both the infection sources and their infection regions, i.e., the subset of nodes likely to be infected by each source, when the number and locations of the sources are unknown a priori.

A. Related Works

Existing works related to infection spreading in a network have primarily focused on the parameters of the diffusion process such as the outbreak thresholds and the effect of network structures [10]–[13]. Little work has been done on identifying the infection sources. Our aim is to identify a set of nodes most likely to be the infection sources after the infection has spread for some time. This formulation is of interest in various practical scenarios, including the spreading of a new disease in a population network. By identifying the initial infectious sources, we can focus scarce resources like DNA testing on a small select group of patients instead of on the whole population. Other examples include identifying the initial entry points of a computer virus into a computer network, and the initiators of a rumor in a social network.

Manuscript received March 22, 2012; revised July 12, 2012 and February 01, 2013; accepted March 14, 2013. Date of publication April 04, 2013; date of current version May 09, 2013. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Shuguang (Robert) Cui. This work was supported by the MOE AcRF Tier 1 under Grant M52040000.

The authors are with the Department of Electrical and Electronic Engineering, Nanyang Technological University, 639798, Singapore (e-mail: wluo1@e.ntu.edu.sg, wptay@ntu.edu.sg lengmei@ntu.edu.sg).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2013.2256902

The case where there is a single infection source has been studied in [14]. Based only on the knowledge of which nodes are infected and the underlying network structure, an estimator based on the linear extensions count of a poset or the number of infection sequences (cf. Section II) was derived to identify the most likely infection source. It was shown in [14] that finding a single infection source is a #P-complete problem even in the case where the infection is relatively simple, with infection from an infected node being equally likely to be transmitted to any of its neighbors at each time step. This simple infection model is based on the classical *susceptible-infected* (SI) model [15], which has been widely used in modeling viral epidemics [16]–[21]. An algorithm for evaluating the single source estimator was proposed in [14], and it was shown to have complexity[1] $O(n)$ for tree networks, where n is the total number of infected nodes. Furthermore, it was shown that this estimator performs well in a very general class of tree networks known as the geometric trees (cf. Section III-D), and identifies the infection source with probability going to one as n increases.

In many applications, there may be more than one infection source in the network. For example, an infectious disease may be brought into a country through multiple individuals. Multiple individuals may collude in spreading a rumor or malicious piece of information in a social network. In this paper, we investigate the case where there may be multiple infection sources, and when the number of infection sources is unknown a priori. We also consider the problem of estimating the infection region of each source, and show that a direct application of the algorithm in [14] performs significantly worse than our proposed algorithms if there are more than one infection sources. We also note that [14] provides theoretical performance measures for several classes of tree networks, which we are unable to do here except for the class of geometric trees, because of the greater complexity of our proposed algorithms. Instead, we provide simulation results to verify the performance of our algorithms.

A related problem is the detection and localization of diffusive sources using wireless sensor networks [22]–[27]. The diffusion models used under this framework are based on spatio-temporal diffusion models [22] or state-space models with linear dynamics [23], where information like the physical positions of sensors are known. There is no natural translation of the source detection and localization problem in a sensor network to other networks like a computer network, without performing discretization and introducing a combinatorial aspect to the problem, as is done in [28] and [29]. Similarly, inference of viral epidemic processes in populations has been studied in [10], [12], [15], where various features related to the propagation of a viral epidemic, such as the rates of infection and the length of latency periods are investigated. These works' focus is on specific viral infection processes with assumptions that do not naturally hold for infection processes in other networks. Moreover, there is little work on determining the sources or index cases of a disease.

On the other hand, the infection source estimation algorithms we consider in this paper can be useful in applications like pollution source localization, where we are limited to inexpensive sensors capable only of detecting the presence or absence of a pollutant, and the identities of its neighbors. In this case, spatio-temporal diffusion models are not applicable as we only have knowledge of which nodes are "infected". The algorithms we study in this paper are also applicable to inferring infection sources in viral epidemics, when little information about the epidemic propagation characteristics is available.

B. Our Contributions

In this paper, we consider the estimation of multiple infection sources when the number of infection sources is unknown a priori. We adopt the same SI diffusion model as in [14], as this has been widely used to model various infection spreading processes [16]–[21]. The results of this work are applicable to scenarios where the infection spreads in an approximately homogeneous way, with infections happening independently. Examples include the spreading of a new disease in a human population, where nobody has yet developed any immunity to the disease. A novel computer virus attacking a network can also be modeled using a homogeneous spreading process. On the other hand, our model is highly simplistic and does not model many other spreading processes of practical interest. However, as alluded to earlier, finding the infection sources in this simple model is already very challenging. The focus of our work is not on modeling infection processes. Rather, by restricting our analysis to the simplest homogeneous exponential spreading model, we hope to gain insights into identifying multiple infection sources in real networks. We show that unlike the single source estimation problem, the multiple source estimation problem is much more complex and cannot be solved exactly even for regular trees. Our main contributions are the following.

i) For the case of a tree network, and when it is known that there are two infection sources, we derive an estimator for the infection sources based on the infection sequences count. The estimator can be calculated in $O(n^2)$ time complexity, where n is the number of infected nodes.

ii) When there are at most two infection sources that are at least two hops apart, we derive an estimator for the class of geometric trees based on approximations of the estimator in (i), and we show that our estimator correctly estimates the number of infection sources and correctly identifies the source nodes, with probability going to one as the number of infected nodes increases.

iii) We derive an estimator for the infection regions of every infection source under a simplifying technical condition.

iv) For general graphs, when there are at most $k_{\max}$ infection sources, we provide an estimation procedure for the infection sources and infection regions. Simulations suggest that on average, our estimators are within a few hops of the true infection sources in the infection graph.[2]

v) We test our estimators on real data in Section V-C. The first test is based on real contact tracing data of a

[1] A function $f(n) = O(g(n))$ if $f(n) \leq cg(n)$ for some constant c and for all n sufficiently large.

[2] In general, we do not know the whole underlying network, but rather the subgraph of infected nodes. For example, in the case of a contagious disease spreading in a population, we only perform contact tracing on the patients to construct the connections among them. From our simulation studies, the infection graph typically has an average diameter of more than 27 hops even though the underlying network's diameter is much smaller.

patient cluster during the SARS outbreak in Singapore in 2003. Our estimator correctly identifies the number of index cases for the cluster to be one and successfully finds this index case. The second test considers the Arizona-Southern California cascading power outages in 2011. Our estimator correctly identifies the number of outage sources for the main affected power network to be two, and the distance between our estimators and the real sources are within 1 hop. These tests suggest that our estimator has reasonable performance in some applications even though we have adopted a simplistic infection model.

The rest of the paper is organized as follows. In Section II, we present the system model and problem formulation. In Section III, we derive estimators for infection sources and regions for tree networks, and present algorithms to evaluate them. We also show asymptotic results for geometric tree networks. We discuss estimation algorithms for general graphs in Section IV. In Section V, we present simulations and tests on real data to verify the performance of our proposed estimators. Finally we conclude and summarize in Section VI.

II. Problem Formulation

In this section, we describe our model and assumptions, introduce some notations, and present some preliminary results. Consider an undirected graph $G = (V, E)$, where V is the set of nodes and E is the set of edges. If there is an edge connecting two nodes, we say that they are neighbors. The neighborhood $\mathcal{N}_G(v)$ of a node v is the set of all neighbors of v in G. The length of the shortest path between u and v is denoted as $d(u, v)$. In a computer network, the graph G models the interconnections between computers in the network. In the example of a population or a social network, V is the set of individuals, while an edge in E represents a relationship between two individuals. We define an **infection** to be a property that a node in G possesses, and can be transmitted to a neighboring node. When a node has an infection, we say that it is infected. The neighbors of an infected node is said to be susceptible. We assume the susceptible-infected model [15], where once a node has been infected, it will not lose its infection. We adopt the same infection spreading process as in [14], where the time taken for an infected node to infect a susceptible neighbor is exponentially distributed with rate 1. All infections are independent of each other. Therefore, if a susceptible node has more than one infected neighbors and subsequently becomes infected, its infection is transmitted by one of its infected neighbors, chosen uniformly at random. For mathematical convenience, we also assume that G is large so that boundary effects can be ignored in our analysis.

Suppose that at time 0, there are $k \geq 1$ nodes in the infected node set $S^* = \{s_1, \ldots, s_k\} \subset V$. These are the **infection sources** from which all other nodes get infected. Suppose that after the infection process has run for some time, and n nodes are observed to be infected. Typically, n is much larger than k. These nodes form an **infection graph** $G_n = (V_n, E_n)$, which is a subgraph of G. Let $\mathcal{A}_n^* = \cup_{i=1}^k A_{n,i}$ be a partition of the infected nodes V_n so that $A_{n,i} \cap A_{n,j} = \emptyset$ for $i \neq j$, with each partition $A_{n,i}$ being connected in G_n, and consisting of the nodes whose infection can be traced back to the source node s_i. The set $A_{n,i}$ is called the **infection region** of s_i, and we say that $\mathcal{A}_n^*$ is the **infection partition**. Given G_n, our objective is to infer the sources of infection S^* and to estimate $\mathcal{A}_n^*$. In addition, if we do not have prior knowledge of the number of infection sources k, we also aim to infer the number of infection sources. Without loss of generality, we assume that G_n is connected, otherwise the same estimation procedure can be performed on each of the components of the graph. We also assume that there are at most $k_{\max}$ infection sources, i.e., the number of infection sources $k \leq k_{\max}$. From a practical point of view, if two infection sources are close to each other, we can ignore either one of them and treat the infection as spreading from a single source. Therefore, we are interested in cases where the infection sources are separated by a minimum distance. These assumptions are summarized in the following.

Assumption 1: The number of infection sources is at most $k_{\max}$, and the infection graph G_n is connected.

Assumption 2: For all $s_i, s_j \in S^*$, the length of the shortest path between them $d(s_i, s_j) \geq \tau$, where τ is a constant greater than 1.

Assumption 3: Every node in G has bounded degree, with d_* being the maximum node degree.

Suppose that our priors for S^* and $\mathcal{A}_n^*$ are uniform over all possible realizations, and let $\mathbb{P}$ be the probability measure of the infection process. We seek S and $\mathcal{A}_n$ that maximize the posterior probability of S^* and $\mathcal{A}_n^*$ given G_n,

$$\mathbb{P}(S^* = S, \mathcal{A}_n^* = \mathcal{A}_n \mid G_n) \propto P(G_n \mid S) P(\mathcal{A}_n \mid S, G_n), \tag{1}$$

where $P(G_n \mid S)$ is the probability of observing G_n if S is the set of infection sources, and $P(\mathcal{A}_n \mid S, G_n)$ is the probability that $\mathcal{A}_n^* = \mathcal{A}_n$ conditioned on S being the infection source set and the infection graph being G_n.

For any source set S, let an **infection sequence** $\sigma = (\sigma_1, \ldots, \sigma_{n-k})$ be a sequence of the nodes in G_n, excluding the the k source nodes in S, arranged in ascending order of their infection times (note that with probability one, no two infection times are the same). For any sequence to be an infection sequence, a necessary and sufficient condition is that any infected node σ_i, $i = 1, \ldots, n-k$, has a neighbor in $S \cup \{\sigma_1, \ldots, \sigma_{i-1}\}$. We call this the *infection sequence property*. An example is shown in Fig. 1. Let $\Omega(G_n, S)$ be the set of infection sequences for an infection graph G_n and source set S, and let $C(S \mid G_n) = |\Omega(G_n, S)|$ be the number of infection sequences. We have

$$P(G_n \mid S) = \sum_{\sigma \in \Omega(G_n, S)} P(\sigma \mid S), \tag{2}$$

where $P(\sigma \mid S)$ is the probability of obtaining the infection sequence σ conditioned on S being the infection sources.

Evaluating the expression (2) and maximizing (1) for a general G_n is a computationally hard problem as it involves combinatorial quantities. As shown in [14], if G is a regular tree and $|S| = 1$, $P(G_n \mid S)$ is proportional to $|\Omega(G_n, S)|$, which is equivalent to the number of linear extensions of a poset. It is

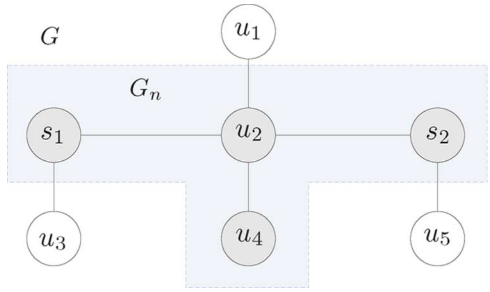


Fig. 1. Example of an infection sequence. The shaded nodes are the infected nodes which form the infection graph G_n. Infection sources are $S = \{s_1, s_2\}$. The sequence (u_2, u_4) is an infection sequence, but (u_4, u_2) is not. The probability of the infection sequence $\sigma = (u_2, u_4)$ is then given by $P(\sigma \mid S) = \frac{2}{4} \times \frac{1}{4} = \frac{1}{8}$. The first fraction $\frac{2}{4}$ is obtained by observing that when only s_1 and s_2 are infected, there are four edges (s_1, u_2), (s_1, u_3), (s_2, u_2), and (s_2, u_5) for the infection to spread. The infection is equally likely to spread along any of these four edges, out of which two results in the infection of node u_2. After u_2 is infected, there are 4 edges over which the infection can spread and this corresponds to the fraction $\frac{1}{4}$.

known that evaluating the linear extensions count is a #P-complete problem [30]. As such, we will make a series of approximations to simplify the problem, and present numerical results in Section V to verify our algorithms. The first approximation we make is to evaluate the estimators

$$\hat{S} = \arg \max_{\substack{S \subset V_n \\ |S| \le k_{\max}}} P(G_n \mid S), \tag{3}$$

$$\hat{\mathcal{A}}_n(\hat{S}) = \arg \max_{\mathcal{A}_n} P(\mathcal{A}_n \mid \hat{S}, G_n), \tag{4}$$

instead of the exact maximum a posteriori (MAP) estimators for (1). Even with this approximation, the optimal estimators are difficult to compute exactly, and may not be unique in general. Therefore, our goal is to design algorithms that are approximately optimal but computationally efficient. In Section III, we make further approximations and design algorithms to evaluate the estimators $\hat{S}$ and $\hat{\mathcal{A}}_n(\hat{S})$ when G is a tree. In Section IV, we consider the case when G is a general graph. For the reader's convenience, we summarize some notations commonly used in this paper in Table I. Several notations have been introduced previously, while we formally define the remaining ones in the sequel where they first appear.

III. Identifying Infection Sources and Regions for Trees

In this section, we consider the problem of estimating the infection sources and regions when the underlying network G is a tree. We first derive an estimator for the infection partition in (4), given any source node set S and G_n. Then, we derive an estimator based on the number of infection sequences. Next, we consider the case where there are two infection sources, propose approximations that allow us to compute the estimator with reasonable complexity, and show that our proposed estimator works well in an asymptotically large geometric tree under some simplifying assumptions. In most practical applications, the number of infection sources is not known a priori. We present a heuristic algorithm for general trees to estimate the infection sources when the number of infection sources is unknown, but bounded by $k_{\max}$.

TABLE I
Summary of Notations Used

Symbol	Definition
G	underlying network
$d(u, v)$	length of the shortest path between u and v
$\mathcal{N}_G(u)$	set of neighbors of u in G
$\deg_G(u)$	number of neighbors of node u in G
G_n	infection graph with n infected nodes
S^*	infection sources
$\mathcal{A}_n^*$	infection partition of an infection graph G_n
$A_{n,i}$	infection region of an infection source s_i
$\Omega(G_n, S)$	set of infection sequences for an infection graph G_n and source set S
$C(S \mid G_n)$	$= \lvert\Omega(G_n, S)\rvert$
Symbol	**Definition** (defined implicitly w.r.t. G_n)
$\rho(u, v)$	path between u and v in the infection graph G_n
$T_v(S)$	tree in G_n, rooted at v w.r.t. source set S
$T_M(S)$	$= \cup_{v \in M} T_v(S)$, where M is a subset of nodes
$I_i(\xi; S)$	$= \sum_{j<i} \lvert T_{\xi_j}(S)\rvert$, where ξ is a sequence of nodes
$I_i^*(s_1, s_2)$	total number of nodes in the i biggest trees in $\{T_u(s_1, s_2) : u \in \rho(s_1, s_2)\}$

A. Infection Partition With Multiple Sources

In this section, we derive an approximate infection partition estimator for (4) given any infection source set S. This estimator is exact under a simplifying technical condition given in Theorem 1 below, the proof of which is provided in Appendix A.

Theorem 1: Suppose that G is a tree with infection sources S, and H_n is the subgraph of G_n consisting of all paths between any pair of nodes in S. If any two paths in H_n do not intersect except possibly at nodes in S, then the optimal estimator $\hat{\mathcal{A}}_n(S)$ for the infection partition is a Voronoi partition of the graph G_n, where the centers of the partitions are the infection sources S.

A Voronoi partition may not produce the optimal estimator for the infection partition in a general infection graph. However, it is intuitively appealing as nodes closer to a particular source are more likely to be infected by that source. For simplicity, we will henceforth use the Voronoi partition of the infection graph G_n as an estimator for $\mathcal{A}_n^*$, and present simulation results in Section V to verify its performance. We will also see in Section III-E that this approximation allows us to design an infection source estimation algorithm with low complexity.

B. Estimation of Infection Sources

We now consider the problem of estimating the set of infection sources S^*. When $|S^*| = 1$, our estimation problem reduces to that in [14], which considers only the single source infection problem. In the following, we introduce some notations, and briefly review some relevant results from [14].

A path between any two nodes u and v in the tree G_n is denoted as $\rho(u, v)$. For any set of nodes S in G_n, consider the connected subgraph $H_n \subset G_n$ consisting of all paths between any pair of nodes in S. Treat this subgraph as a "super" node, with the tree G_n rooted at this "super" node. For any node $v \in G_n \backslash H_n$, we define $T_v(S)$ to be the tree rooted at v with the path from v to H_n removed. For $v \in H_n$, we define $T_v(S)$ to be the tree rooted at v so that all edges between v and its neighbors in

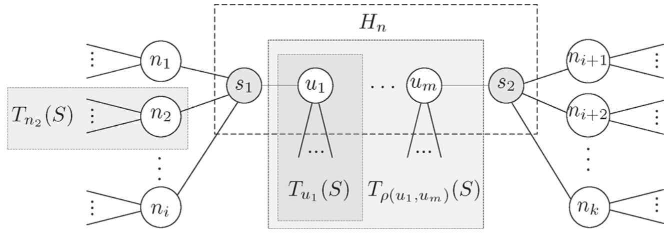


Fig. 2. A sample infection graph with $S = \{s_1, s_2\}$.

H_n are removed.[3] We say that $T_v(S)$ is the tree rooted at v with respect to (w.r.t.) S. For any subset of nodes $M \subset G_n$, we let $T_M(S) = \cup_{v \in M} T_v(S)$. An illustration of these definitions is shown in Fig. 2. If $S = \{s_1, \ldots, s_k\}$, we will sometimes use the notation $T_v(s_1, \ldots, s_k)$ instead.

Recall that $C(S \mid G_n)$ is the number of infection sequences if S is the infection source set. If there is a single infection source node $S = \{s\}$, and G is a regular tree where each node has the same degree, it is shown in [14] that the MAP estimator for the infection source is obtained by evaluating $\hat{S} = \arg\max_{v \in G_n} C(v \mid G_n)$, which seeks to maximize $C(v \mid G_n)$ over all nodes. Therefore, it has been suggested that $C(v \mid G_n)$ can be used as the infection source estimator for general trees. The following result is provided in [14].

Lemma 1: Suppose that G_n is a tree. For any node $s \in G_n$, we have

$$C(s \mid G_n) = n! \prod_{u \in G_n} |T_u(s)|^{-1}. \tag{5}$$

We observe that each term $|T_u(s)|$ in the product on the right hand side (R.H.S.) of (5) is the number of nodes in the sub-tree $T_u(s)$ (and which appears when we account for the number of permutations of these nodes). We can think of the terms in the product being ordered according to the infection spreading sequence, i.e., each time we reach a particular node u, we include terms corresponding to the nodes u can potentially infect. This interpretation is useful in helping us understand the characterization in Lemma 2 for the case where there are two infection sources.

To compute $C(v \mid G_n)$, an $O(n)$ algorithm based on Lemma 1 was provided in [14]. We call this algorithm the Single Source Estimation (SSE) algorithm. We refer the reader to [14] for details about the implementation of the algorithm. Although finding $\hat{S}$ by maximizing $C(s \mid G_n)$ is exact only for regular trees, it was shown in [14] that this estimator has good performance for other classes of trees. In particular, if G is a geometric tree (cf. Section III-D), then the probability, conditioned on $S^* = \{s\}$, of correctly identifying s using $C(s \mid G_n)$ goes to one as $n \to \infty$. Inspired by this result, we propose estimators based on quantities related to $C(S \mid G_n)$ for cases where $|S^*| > 1$. In the following, we first discuss the case where $|S^*| = 2$, and extend the results to the general case where $|S^*|$ is unknown in Section III-E. We then numerically compare our proposed algorithms with a modified SSE algorithm adapted for finding multiple sources in Section V.

C. Two Infection Sources

In this section, we assume that there are two infection sources $S = \{s_1, s_2\}$. Given two nodes u and v in G_n, suppose that $|\rho(u,v)| = m$. For any permutation $\xi = (\xi_1, \ldots, \xi_m)$ of the nodes in $\rho(u,v)$, let

$$I_i(\xi; s_1, s_2) = \sum_{j \le i} \left| T_{\xi_j}(s_1, s_2) \right| \tag{6}$$

be the total number of nodes in the trees rooted at the first i nodes in the permutation ξ. We have the following characterization for $C(s_1, s_2 \mid G_n)$, whose proof is given in Appendix B.

Lemma 2: Suppose that G_n is a tree. Consider any two nodes s_1 and s_2 in G_n, and suppose that $\rho(s_1, s_2) = (s_1, u_1, \ldots, u_m, s_2)$. We have

$$\begin{aligned} C(s_1, s_2 \mid G_n) = (n-2)! \cdot q(u_1, u_m; s_1, s_2) \\ \times \prod_{u \in G_n \setminus \rho(s_1, s_2)} |T_u(s_1, s_2)|^{-1}, \end{aligned} \tag{7}$$

where for $1 \le i \le j \le m$, $q(u_i, u_j; s_1, s_2)$ satisfies the following recursive relationship

$$\begin{aligned} q(u_i, u_j; s_1, s_2) = \left| T_{\rho(u_i, u_j)}(s_1, s_2) \right|^{-1} (q(u_{i+1}, u_j; s_1, s_2) \\ + q(u_i, u_{j-1}; s_1, s_2)) \quad \text{for } i < j, \end{aligned} \tag{8}$$

with $q(v, v; s_1, s_2) = |T_v(s_1, s_2)|^{-1}$ for all $v \in \rho(u_1, u_m)$. Furthermore, we have

$$q(u_1, u_m; s_1, s_2) = \sum_{\xi \in \Gamma(u_1, u_m)} \prod_{i=1}^{m} I_i(\xi; s_1, s_2)^{-1}, \tag{9}$$

and $\Gamma(u_1, u_m)$ is the set of all permutations $\xi = (\xi_1, \ldots, \xi_m)$ of nodes in $\rho(u_1, u_m)$ such that $(\xi_m, \ldots, \xi_1)$ is an infection sequence starting from s_1 and s_2 and resulting in $\rho(s_1, s_2)$.

The characterization for $C(s_1, s_2 \mid G_n)$ is similar to that for the single source case in (5), except for the additional $q(u_1, u_m; s_1, s_2)$ term. We first clarify the meaning of $\Gamma(u_1, u_m)$. Given any infection sequence σ that starts with $\{s_1, s_2\}$ and results in $\rho(s_1, s_2)$, i.e., $\sigma = (\sigma_1, \ldots, \sigma_m) \in \Omega(\rho(s_1, s_2), \{s_1, s_2\})$, we can find a permutation $\xi = (\xi_1, \ldots, \xi_m)$ of nodes in $\rho(u_1, u_m)$ such that $\xi_i = \sigma_{m-i+1}$ for $i = 1, \ldots, m$. In other words, ξ can be interpreted as the *reverse* infection sequence corresponding to σ. Then $\Gamma(u_1, u_m)$ is the set of all such reverse infection sequences corresponding to $\Omega(\rho(s_1, s_2), \{s_1, s_2\})$. We show an illustration of these definitions in Fig. 3. Each term $|T_u(s)|$ in the product in the R.H.S. of (5) can be interpreted as the number of nodes that can be infected via u once u has been infected. Similarly, the sum in (9) is over all possible reverse infection sequences ξ of the nodes in $\rho(u_1, u_m)$, and each

[3] As $T_v(S)$ is defined on G_n, its notation should include G_n. However, in order to avoid cluttered expressions, we drop G_n in our notations. Confusion will be avoided through the context in which these trees are referenced.

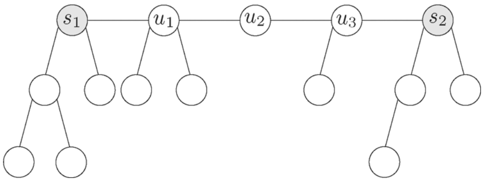


Fig. 3. A sample infection graph with $S = \{s_1, s_2\}$. Given an infection sequence $\sigma = (u_3, u_1, u_2) \in \Omega(\rho(s_1, s_2), \{s_1, s_2\})$, we can find the corresponding reverse infection sequence $\xi = (u_2, u_1, u_3)$. We have $I_1(\xi; s_1, s_2) = |T_{u_2}(s_1, s_2)| = 1$, $I_2(\xi; s_1, s_2) = |T_{u_2}(s_1, s_2)| + |T_{u_1}(s_1, s_2)| = 4$, $I_3(\xi; s_1, s_2) = |T_{u_2}(s_1, s_2)| + |T_{u_1}(s_1, s_2)| + |T_{u_3}(s_1, s_2)| = 6$.

term $I_i(\xi; s_1, s_2)$ in the product within the sum is the number of nodes that can be infected once $\xi_{i+1}, \ldots, \xi_m$ have been infected.

Algorithm 1 Tree Sizes and Products Computation

```
**Inputs**: $G_n$
Choose any node $r \in G_n$ as the root node.
**for** $w \in G_n$ **do**
  Store received messages $f_x(w)$ and $g_x(w)$, for each $x \in \mathrm{ch}(w)$.
  **if** $w$ is a leaf **then**
    $f_w(\mathrm{pa}(w)) = 1$
    $g_w(\mathrm{pa}(w)) = 1$
  **else**
    $f_w(\mathrm{pa}(w)) = \sum_{x \in \mathrm{ch}(w)} f_x(w) + 1$
    $g_w(\mathrm{pa}(w)) = f_w(\mathrm{pa}(w)) \cdot \prod_{x \in \mathrm{ch}(w)} g_x(w)$
  **end if**
  Store $f_{\mathrm{pa}(w)}(w) = n - f_w(\mathrm{pa}(w))$.
  Pass $f_w(\mathrm{pa}(w))$ and $g_w(\mathrm{pa}(w))$ to $\mathrm{pa}(w)$.
**end for**
Set $g_{\mathrm{pa}(r)}(r) = 1$.
**for** $w \in G_n$ **do**
  Store received message $g_{\mathrm{pa}(w)}(w)$ from $\mathrm{pa}(w)$.
  **if** $w$ is not a leaf **then**
    **for** $x \in \mathrm{ch}(w)$ **do**
      $g_w(x) = f_w(x) \cdot g_{\mathrm{pa}(w)}(w) \cdot \prod_{y \in \mathrm{ch}(w) \backslash \{x\}} g_y(w)$
      Pass $g_w(x)$ to $x$.
    **end for**
  **end if**
**end for**
```

By utilizing Lemma 2, we can compute $C(u, v \mid G_n)$ for any two nodes u and v in G_n by evaluating $|T_w(u, v)|$ for all nodes $w \in G_n$, and the quantity $q(u_1, u_m; u, v)$, where $\rho(u, v) = (u, u_1, \ldots, u_m, v)$. With Assumption 3, Algorithm 1 allows us to compute $f_w(u) = |T_w(u)|$ and $g_w(u) = \prod_{v \in T_w(u)} |T_v(u)|$ for all neighbors u of w, and for all $w \in G_n$ in $O(n)$ time complexity. To do this, we first choose any node $r \in G_n$, and consider G_n as a directed tree with r as the root node, and with edges in G_n pointing away from r. Let $\mathrm{pa}(w)$ and $\mathrm{ch}(w)$ be the parent and the set of children of w in the directed tree G_n, respectively. Starting from the leaf nodes, let each non-root node $w \in G_n$ pass two messages containing $f_w(\mathrm{pa}(w))$ and $g_w(\mathrm{pa}(w))$ to its parent. Each node stores the values of these two messages from each of its children, and computes its two messages to be passed to its parent. When r has received all messages from its children, a reverse sweep down the tree is done so that at the end of the algorithm, every node $w \in G_n$ has stored the values $\{f_u(w), g_u(w) : u \in \mathcal{N}_{G_n}(w)\}$. The algorithm is formally described in Algorithm 1. The last product term on the R.H.S. of (7) can then be computed using

$$g(s_1, s_2) = \prod_{w \in \rho(s_1, s_2)} \prod_{x \in \mathcal{N}_{G_n}(w) \backslash \rho(s_1, s_2)} g_x(w), \tag{10}$$

and taking its reciprocal.

Algorithm 2 Two Source Estimation (TSE)

```
**Inputs**: $G_n$
Let $(s_1^*, s_2^*)$ be the maximizer of $C(\cdot, \cdot \mid G_n)$. Set $C^* = 0$.
**for** $d = 1$ to diameter of $G_n$ **do**
  **for** each $s_1 \in G_n$ **do**
    **for** each $s_2$ such that $d(s_1, s_2) = d$ **do**
      Let $\rho(s_1, s_2) = (s_1, u_1, \ldots, u_{d-1}, s_2)$.
      **if** $d = 1$ **then**
        $q(u_1, u_{d-1}; s_1, s_2) = 1$.
      **else if** $d = 2$ **then**
        Store $q(u_1, u_1; s_1, s_2) = |T_{u_1}(s_1, s_2)|^{-1}$ and $|T_{u_1}(s_1, s_2)|$.
      **else**
        Look up $|T_{\rho(u_1, u_{d-2})}(s_1, u_{d-1})|$, $q(u_2, u_{d-1}; u_1, s_2)$, and $q(u_1, u_{d-2}; s_1, u_{d-1})$.
        Store
          $|T_{\rho(u_1, u_{d-1})}(s_1, s_2)| = |T_{\rho(u_1, u_{d-2})}(s_1, u_{d-1})| \cdot |T_{u_{d-1}}(s_1, s_2)|$.
        Store
          $q(u_1, u_{d-1}; s_1, s_2) = \dfrac{q(u_2, u_{d-1}; u_1, s_2) + q(u_1, u_{d-2}; s_1, u_{d-1})}{|T_{\rho(u_1, u_{d-1})}(s_1, s_2)|}$.
      **end if**
      Compute $g(s_1, s_2)$ from (10).
      $C(s_1, s_2 \mid G_n) = (n-2)! q(u_1, u_{d-1}; s_1, s_2) / g(s_1, s_2)$.
      Update $(s_1^*, s_2^*)$ and $C^*$ if $C(s_1, s_2 \mid G_n) > C^*$.
    **end for**
  **end for**
**end for**
```

To compute $C(s_1, s_2 \mid G_n)$ in (7), we still need to compute $q(u_1, u_m; s_1, s_2)$. The recurrence (8) allows us to compute $q(u_1, u_m; s_1, s_2)$ for all $s_1, s_2 \in G_n$ in $O(n^2 d_*^2)$ complexity, where d_* is the maximum node degree. The computation proceeds by first considering each pair of neighbors (u, v). Both nodes have at most d_* neighbors each, so that we need to evaluate $q(u, v; s_1, s_2)$ for all $s_1 \in \mathcal{N}_{G_n}(u) \backslash \rho(u, v)$ and

$s_2 \in \mathcal{N}_{G_n}(v)\backslash\rho(u,v)$. This requires $O(d_*^2)$ computations. The computed values and $T_{\rho(u,v)}(s_1,s_2)$ are stored in a hash table. In the next step, we repeat the same procedure for node pairs that are two hops apart, and so on until we have considered every pair of nodes in G_n. Note that for a path $(u_1,\ldots,u_m)$ and s_1, s_2 neighbors of u_1 and u_m respectively, $q(u_1,u_m;s_1,s_2)$ can be computed in constant time from (8) as $q(u_2,u_m;s_1,s_2) = q(u_2,u_m;u_1,s_2)$ and $q(u_1,u_{m-1};s_1,s_2) = q(u_1,u_{m-1};s_1,u_m)$. A similar remark applies for the computation of $|T_{\rho(u_1,u_m)}(s_1,s_2)|$. In addition, each lookup of the hash table takes $O(1)$ complexity since G_n is known and collision-free hashing can be used. Therefore, the overall complexity is $O(n^2 d_*^2)$. The algorithm to compute the infection sources estimator is formally given in Algorithm 2. We call this the Two Source Estimation (TSE) algorithm, and it forms the basis of our algorithm for multiple sources estimation in the sequel.

D. Geometric Trees With Two Sources

In this section, we study the special case of geometric trees, propose an approximate estimator for geometric trees, and provide theoretical analysis for its performance. First, we give the definition of geometric trees and prove some of its key properties. Then, we derive a lower bound for $C(S \mid G_n)$, and propose an estimator based on this lower bound. We show that our proposed estimator is asymptotically correct, i.e., it identifies the actual infection sources with probability (conditioned on the infection sources) going to one as the infection graph G_n becomes large. For mathematical convenience, instead of letting the number of infected nodes n grow large, we let the time t from the start of the infection process to our observation time become large.

The geometric tree network is defined in [14] w.r.t. a single infection source. In the following, we extend this definition to the case where there are two sources. Let $S^* = \{s_1, s_2\}$ be the infection sources, and let $T'_u(s_1,s_2)$ be defined in the graph G in the same way as $T_u(s_1,s_2)$ is defined for G_n. Let $\mathcal{N}_G(\rho(s_1,s_2))$ be the set of nodes that have a neighboring node in $\rho(s_1,s_2)$. For each node u, let $n(u,r)$ be the number of nodes in $T'_u(s_1,s_2)$ that are at a distance r from u. We say that G is a geometric tree if for all $u \in \mathcal{N}(\rho(s_1,s_2))$, we have

$$br^\alpha \le n(u,r) \le cr^\alpha, \tag{11}$$

where α, b, and c are fixed positive constants with $b \le c$. The condition (11) implies that all trees defined w.r.t. the infection sources are growing polynomially fast at about the same rate. As we have assumed that the infection rates are homogeneous for every node, the resulting infection graph G_n will also be approximately regular with high probability. We have the following properties for a geometric tree, whose proofs are in Appendix C.

Lemma 3: Suppose that G is a geometric tree with two infection sources $S^* = \{s_1, s_2\}$. Let α, b and c be fixed positive constants satisfying (11) for the geometric tree G. Let t be the time from the start of the infection process to our observation time. For any $\epsilon \in (0,1)$, let $\mathcal{E}_t$ be the event that all nodes within distance $t(1-t^{-1/2+\epsilon})$ of either source nodes are infected, and no nodes greater than distance $t(1+t^{-1/2+\epsilon})$ of either source nodes are infected. Then, there exists t_0 such that for all $t \ge t_0$, $\mathbb{P}(\mathcal{E}_t) \ge 1-\epsilon$. Furthermore, conditioned on $\mathcal{E}_t$, we have for all $u \in \mathcal{N}_G(s_1) \cup \mathcal{N}_G(s_2)$ or $u = \rho(s_1,s_2)\backslash S^*$,

$$N_{\min}(t) \le |T_u(s_1,s_2)| \le N_{\max}(t), \tag{12}$$

where

$$N_{\min}(t) = \frac{b}{1+\alpha}\left(t - t^{\frac{1}{2}+\epsilon} - d(s_1,s_2) - 2\right)^{\alpha+1}, \tag{13}$$

and

$$N_{\max}(t) = \frac{c}{1+\alpha}\left(t + t^{\frac{1}{2}+\epsilon}\right)^{\alpha+1}. \tag{14}$$

In addition, for $t \ge t_0$, we have

$$\frac{N_{\min}(t)}{N_{\max}(t)} \ge \frac{b}{c}(1-\epsilon).$$

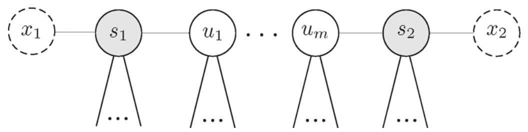


Fig. 4. Addition of virtual nodes x_1 and x_2.

The infection sequences count in (7) is not amendable to analysis. In the following, we seek an approximation to simplify our analysis. For $s_1, s_2 \in G_n$, suppose that $\rho(s_1,s_2) = (s_1,u_1,\ldots,u_m,s_2)$, with $p = |\rho(s_1,s_2)| = m+2$. Instead of computing $C(s_1,s_2 \mid G_n)$, we consider a new infection graph G'_n with two "virtual" nodes x_i, $i=1,2$ added, where x_i is attached to s_i (see Fig. 4). We now consider the infection sequence count $C(x_1,x_2 \mid G'_n) \ge C(s_1,s_2 \mid G_n)$. Since the trees rooted at x_i are single node trees, we have

$$\begin{aligned} C(x_1,x_2 \mid G'_n) &= C(s_1,x_2 \mid G'_n) + C(x_1,s_2 \mid G'_n) \\ &\le 2(n-1)C(s_1,s_2 \mid G_n), \end{aligned}$$

where the last inequality follows because if s_1 and x_2 are sources, then s_2 can be inserted in any of at most $n-1$ positions in an infection sequence from $\Omega(G_n,\{s_1,s_2\})$, so that $C(s_1,x_2 \mid G'_n) \le (n-1)C(s_1,s_2 \mid G_n)$. A similar argument holds for $C(x_1,s_2 \mid G'_n) \le (n-1)C(s_1,s_2 \mid G_n)$.

Let $\xi^* = (\xi_1^*,\ldots,\xi_p^*)$ be a permutation of the nodes in $\rho(s_1,s_2)$ such that $|T_{\xi_i^*}(s_1,s_2)| \ge |T_{\xi_j^*}(s_1,s_2)|$ for all $1 \le i \le j \le p$, i.e., the nodes in ξ^* are arranged in descending order of the size of the sub-trees rooted at them. Let $I_i^*(s_1,s_2) = I_i(\xi^*;s_1,s_2)$ (cf. the definition in (6)) be the total number of nodes in the i biggest sub-trees in $\{T_u(s_1,s_2) : u \in \rho(s_1,s_2)\}$. From Lemma 2, we have

$$\begin{aligned} C(x_1,x_2 \mid G'_n) \ge n! \cdot 2^{p-1} \prod_{i=1}^{p} I_i^*(s_1,s_2)^{-1} \\ \cdot \prod_{u \in G_n \backslash \rho(s_1,s_2)} |T_u(s_1,s_2)|^{-1}, \end{aligned} \tag{15}$$

where the inequality holds because $|\Gamma(s_1, s_2)| = 2^{p-1}$, and each term in the sum on the R.H.S. of (9) is lower bounded by $\prod_{i=1}^{p} I_i^*(s_1, s_2)^{-1}$. We use the lower bound in (15) as a proxy for $C(s_1, s_2 \mid G_n)$. However, we have used a very loose lower bound in (15), so we propose the estimator

$$\tilde{S} = \arg \max_{s_1, s_2 \in G_n} \tilde{C}(s_1, s_2 \,|\, G_n), \tag{16}$$

where

$$\tilde{C}(s_1, s_2 \,|\, G_n) = n! \cdot Q(s_1, s_2) \prod_{u \in G_n \setminus \rho(s_1, s_2)} |T_u(s_1, s_2)|^{-1}, \tag{17}$$

$$Q(s_1, s_2) = [2(1+\delta)]^{p-1} \prod_{i=1}^{p} I_i^*(s_1, s_2)^{-1},$$

and δ is a fixed positive constant, to be chosen based on prior knowledge about the graph G. Algorithm 2 can be modified to find the maximizer for $\tilde{C}(\cdot, \cdot \,|\, G_n)$. We call this the geometric tree TSE algorithm. The following result provides a way to choose δ, and shows that our proposed estimator $\tilde{S}$ is asymptotically correct in a geometric tree. A proof is provided in Appendix D.

Theorem 2: Suppose that G is a geometric tree with two infection sources $S^* = \{s_1^*, s_2^*\}$. Let $d_{\min}$ and $d_{\max}$ be constants such that $\deg_G(s_i) \in [d_{\min}, d_{\max}]$ for $i = 1, 2$. Let b and c be fixed positive constants satisfying (11) for the geometric tree G. Suppose that

$$d_{\min} \geq \frac{3}{2} + \frac{c}{b}\sqrt{2d_{\max}}. \tag{18}$$

Then, for any δ in the non-empty interval

$$\left(\frac{cd_{\max}}{b(d_{\min}-1)} - 1, \frac{b(d_{\min}-2)}{2c} - 1\right), \tag{19}$$

we have

$$\lim_{t\to\infty} \mathbb{P}(\tilde{S} = S^* \mid S^*) = 1.$$

Theorem 2 implies that if we know the constants governing the regularity condition (11) for G, we can choose a δ so that our estimator $\tilde{S}$ gives the true infection sources with high probability if the infection graph G_n is large. The class of geometric trees as defined by (11) can be used to model various scenarios in practice, e.g., a tree spanning a wireless sensor network with nodes randomly scattered. However, the assumption (11) may also be overly strong for other applications. In Section V, we perform numerical studies to gain insights into the performance of our proposed estimator for different classes of tree networks.

E. Unknown Number of Infection Sources

In most practical applications, the number of infection sources is not known a priori. However, typically we may be able to guess the maximum number of infection sources $k_{\max}$, or we can choose a reasonable value of $k_{\max}$ depending on the size of the infection graph G_n. In this section, we present a *heuristic* algorithm that allows us to estimate the infection sources with a given $k_{\max}$.

We first consider the instructive case where $k_{\max} = 2$ and G is a geometric tree. In this case, the number of infection sources can be either one or two. Suppose we run the geometric tree TSE algorithm on G_n. We have the following result, whose proof is in Appendix E.

Theorem 3: Suppose that there is a single infection source s and G is a geometric tree with (11) holding for all nodes u that are neighbors of s. Suppose that s has degree $\deg_G(s) \in [d_{\min}, d_{\max}]$, where $d_{\min}$ and $d_{\max}$ are positive constants satisfying (18). Then, for any δ in the interval (19), the geometric tree TSE algorithm estimates as sources s and one of its neighbors with probability (conditioned on s being the infection source) going to 1 as $t \to \infty$.

Algorithm 3 Infection Partitioning (IP)

```
Inputs: An infection source set $S^{(0)} = \{s_i^{(0)} : i = 1, \ldots, m\}$ in $G_n$.
Iterations:
for $l = 1$ to MaxIter do
  Run the Voronoi partitioning algorithm with centers in $S^{(l-1)}$ to obtain the infection partition $\mathcal{A}^{(l)} = \cup_{i=1}^{m} A_i^{(l)}$.
  for $i = 1$ to $m$ do
    Run SSE algorithm in $A_i^{(l)}$ to obtain
        $s_i^{(l)} = \arg\max_{s \in A_i^{(l)}} C\left(s \,\middle|\, A_i^{(l)}\right)$.
  end for
  $S^{(l)} := \{s_i^{(l)} : i = 1, \ldots, m\}$
  if $\max_{1 \leq i \leq m} d(s_i^{(l)}, s_i^{(l-1)}) \leq \eta$ for some fixed small positive $\eta$ then
    break
  end if
end for
return $(S^{(l)}, \mathcal{A}^{(l)})$
```

Theorem 3 implies that when there exists only one source, the geometric tree TSE algorithm finds two neighboring nodes, one of which is the true source. From Theorem 2 and Assumption 2, if there are two sources, our algorithm identifies the two source nodes, which are at least two hops from each other, with high probability. Therefore, by checking the distance between the two nodes identified by the geometric tree TSE algorithm, we can estimate the number of source nodes in the infection graph. This observation together with Theorem 1 suggest the following heuristic.

i) Randomly choose $k_{\max}$ nodes satisfying Assumption 2 as the infection sources and find a Voronoi partition for G_n. Use the SSE algorithm to find a source node for each infection region. Repeat these steps until for every region, the distance between estimated source nodes between iterations is below a fixed threshold or a maximum number of iterations is reached. We call this the Infection Partition (IP) Algorithm (see Algorithm 3).

ii) For any two regions in the partition obtained from step (i) that are connected by an edge in G_n, run the TSE algorithm in the combined region to find two source estimates. If the two estimates have distance less than τ, we decrement the number of source nodes, and repeat step (i).

iii) The above two steps are repeated until no two pairs of regions in the Voronoi partition can be combined. The formal algorithm is given as the Multiple Sources Estimation and Partitioning (MSEP) algorithm in Algorithm 4.

Algorithm 4 Multiple Sources Estimation and Partitioning (MSEP)

```
1: Inputs: $G_n$ and $k_{\max}$.
2: Initialization:
3: $k := k_{\max}$ and choose $S := \{s_1, \ldots, s_k\}$ randomly
   in $G_n$.
4: Iterations:
5: while $k > 1$ do
6:   $(S, \mathcal{A})$ = Algorithm IP$(S)$
7:   $S' := S$
8:   for all regions $A_i$ and $A_j$ in the partition $\mathcal{A}$ such that
     there exists an edge $(u, v)$ in $G_n$ with $u \in A_i$ and
     $v \in A_j$ do
9:     Set $(u, v)$ = Algorithm TSE$(A_i \cup A_j)$.
10:    if $d(u, v) < \tau$ then
11:      Merge $A_i$ and $A_j$, set $s_i = u$ and discard $s_j$
12:      $k := k - 1$
13:      break
14:    end if
15:  end for
16:  if $S = S'$ then
17:    break
18:  end if
19: end while
20: return $(S, \mathcal{A})$
```

To compute the complexity of the MSEP algorithm, we note that since the IP algorithm is based on the SSE algorithm, it has complexity $O(n)$. For each value of $k = 1, \ldots, k_{\max}$ in the MSEP algorithm, there are $O(k^2)$ pairs of neighboring regions in the infection partition. For each pair of region, the TSE algorithm makes $O(n^2)$ computations. Summing over all $k = 1, \ldots, k_{\max}$, the time complexity of the MSEP algorithm can be shown to be $O({k_{\max}}^3 n^2)$. On the other hand, to compute $C(S \mid G_n)$ for $|S^*| = k_{\max}$ would require $O(n^{k_{\max}})$ computations.

IV. Identifying Infection Sources and Regions for General Graphs

In this section, we generalize the MSEP algorithm to identify multiple infection sources in general graphs G. In [14], the SSE algorithm is extended to general graphs when it is known that there is only a single infection source in the network using a heuristic. The algorithm first chooses a node s of G_n as the root node, and generates a spanning tree $T_{\text{bfs}}(s, G_n)$ of G_n rooted at s using the breadth-first-search (BFS) procedure. The SSE algorithm is then applied on this spanning tree to compute $C(s \mid T_{\text{bfs}}(s, G_n))$. In addition, the infection sequences count is weighted by the likelihood of the BFS tree. This is repeated using every node in G_n as the root node, and the node $\hat{s}$ with the maximum weighted infection sequences count is chosen as the source estimator, i.e.,

$$\hat{s} = \arg\max_{v \in G_n} P(\sigma_v \mid v) C(s \mid T_{\text{bfs}}(v, G_n)),$$

where σ_v is the sequence of nodes that corresponds to an infection spreading from v along the BFS tree. It can be shown that this algorithm has complexity $O(n^2)$. For further details, the reader is referred to [14]. We call this algorithm the SSE-BFS algorithm in this paper.

We adapt the MSEP algorithm for general graphs using the same BFS heuristic. Specifically, we replace the SSE algorithm in line 6 of the IP algorihm with the SSE-BFS algorithm. In addition, in line 9, we run the TSE algorithm on $T_{\text{bfs}}(s_i, A_i) \cup T_{\text{bfs}}(s_j, A_j)$, where the two BFS trees are connected by randomly selecting an edge (u, v) in G_n with $u \in T_{\text{bfs}}(s_i, A_i)$ and $v \in T_{\text{bfs}}(s_j, A_j)$. We call this modified algorithm the MSEP-BFS algorithm. Since the worst case complexity for the SSE-BFS algorithm is $O(n^2)$, the complexity of the MSEP-BFS algorithm can be shown to be $O({k_{\max}}^3 n^2)$, which is the same complexity as the MSEP algorithm. To verify the effectiveness of the MSEP-BFS algorithm, we conduct simulations on both synthetic and real world networks in Section V.

V. Simulation Results and Tests

In this section, we present results from simulations and tests on real data to verify our proposed algorithms. We first consider geometric tree networks and regular tree networks with various numbers of infection sources, and then we present results on small-world networks and a real world power grid network. We also apply our algorithms to the contact tracing data obtained during the SARS outbreak in Singapore in 2003 [1] and the Arizona-Southern California cascading power outages in 2011 [31].

A. Synthetic Networks

We perform simulations on geometric trees, regular trees, and small-world networks. The number of infection sources $|S^*|$ are chosen to be 1, 2, or 3, and we set $k_{\max} = 3$. For each type of network and each number of infection sources, we perform 1000 simulation runs with 500 infected nodes. We randomly choose infection sources satisfying Assumption 2 and obtain the infection graph by simulating the infection spreading process using the SIR model. Finally, the MSEP or MSEP-BFS algorithm for tree networks and small-world networks respectively, is applied to the infection graph to estimate the number and locations of the infection sources. The estimation results for the number of infection sources $|\hat{S}|$ in different scenarios are shown in Fig. 5. It can be seen that our algorithm correctly finds the number of infection sources more than 93% of the time for geometric trees, and more than 71% of the time for regular trees. The accuracy of about 69.2% for small-world networks is worse than that for

TABLE II
PERFORMANCE COMPARIONS

Simulation settings		Average diameter of G_n	Average error distance Δ					Average minimum infection region covering percentage (%)
network topology	$\|S^*\|$		MSEP/MSEP-BFS		nSSE			
			$\eta = 0$	$\eta =$ diameter	$\eta = 0$	$\eta =$ diameter	known $\|S^*\|$	
geometric trees	2	63.7	0.61	1.72	9.65	30.16	12.85	97.06
	3	66.2	0.91	2.42	7.69	29.95	14.84	89.77
regular trees	2	40.5	0.84	6.07	4.50	17.70	6.13	73.82
	3	43.7	0.94	6.24	3.39	17.47	6.59	65.95
small-world networks	2	35.5	2.95	8.19	5.40	17.13	8.28	76.62
	3	40.9	2.58	8.18	4.99	18.56	10.37	60.69
power grid network	2	27.3	3.65	7.39	5.50	14.66	7.89	70.29
	3	30.8	2.85	8.47	4.71	14.75	8.89	59.95

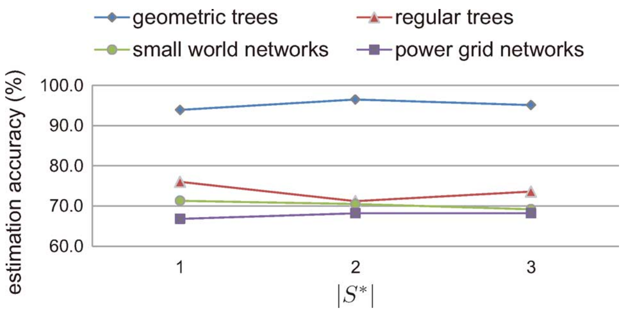


Fig. 5. Estimating the number of infection source nodes.

the tree networks, as the infection tree for a small-world network has to be estimated using the BFS heuristics, thus additional errors are introduced into the procedure.

When there are more than one infection sources, we compare the performance of the MSEP algorithm with a naive estimator based on the SSE algorithm. We call this the nSSE algorithm. Specifically, in the estimator for tree networks, we first compute $C(u \mid G_n)$ for all nodes $u \in G_n$, and choose the $|S^*|$ nodes with the largest counts as the source nodes. In non-tree networks, we use the SSE-BFS algorithm. Since the nSSE algorithm can not estimate $|S^*|$, we consider two variants. In the first variant, we assume the nSSE algorithm has prior knowledge of $|S^*|$. In the second variant, we guess $|S^*|$ by choosing uniformly from $\{1, \ldots, k_{\max}\}$.

To quantify the performance of each algorithm, we first match the estimated source nodes $\hat{S} = \{\hat{s}_i : i = 1, \ldots, |\hat{S}|\}$ with the actual sources S^* so that the sum of the error distances between each estimated source and its match is minimized. Let this matching be denoted by the function π, which matches each actual source s_i to $\hat{s}_{\pi(i)}$. If we have incorrectly estimated the number of infection sources, i.e., $|\hat{S}| \neq |S^*|$, we add a penalty term to this sum. The average error distance is then given by

$$\Delta = \frac{1}{|S^*|}\left(\sum_{i=1}^{\min(|S^*|,|\hat{S}|)} d\left(\hat{s}_{\pi(i)}, s_i\right) + \eta\left||\hat{S}| - |S^*|\right|\right),$$

where η is a penalty weight for incorrectly estimating the number of infection sources. For different applications, we may assign different values to η depending on how important it is to estimate correctly the number of infection sources. In our simulations, we consider the cases where $\eta = 0$, and where η is the diameter of the infection graph. The average error distances for the different types of networks are provided in Table II. Clearly, the MSEP/MSEP-BFS algorithm outperforms the nSSE algorithm, even when the nSSE algorithm has prior knowledge of the number of sources. When $|S^*|$ is known a priori, the performance of the nSSE algorithm deteriorates with increasing $|S^*|$. This is to be expected as the SSE algorithm assumes that the node with the largest infection sequence count is the only source, and this node tends to be close to the distance center [32] of the infection graph. The histogram of the average error distances when $\eta = 0$ are shown in Fig. 6.

The MSEP/MSEP-BFS algorithm also estimates the infection region of each source. To evaluate its accuracy, we first perform the matching process described previously. Let the true infection region of s_i be $A_{n,i}$ and the estimated infection region of $\hat{s}_{\pi(i)}$ be $\hat{A}_{n,i}$, where we set $\hat{A}_{n,i} = \emptyset$, if we have underestimated the number of sources and s_i is unmatched. We define the correct infection region covering percentage for s_i as the ratio between $|\hat{A}_{n,i} \cap A_{n,i}|$ and $|A_{n,i}|$, and we compute the minimum (or worst case) infection region covering percentage as

$$\min_{i\in\{1,\ldots,|S^*|\}} \frac{|\hat{A}_{n,i} \cap A_{n,i}|}{|A_{n,i}|}.$$

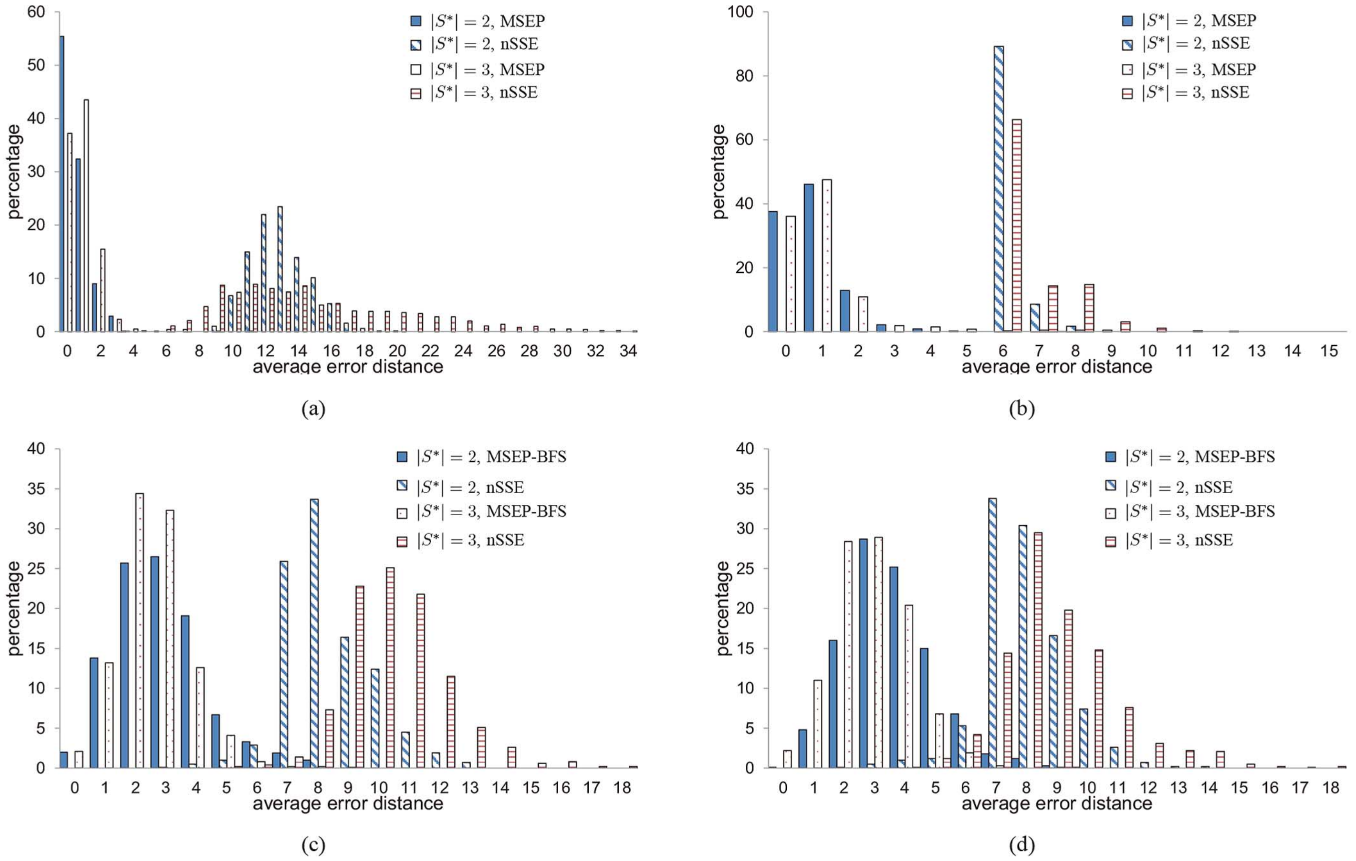
This is then averaged over all simulation runs. We find that the average minimum infection region covering percentage is more than 59% for all networks, as shown in Table II.

B. Real World Networks

We verify the performance of the MSEP-BFS algorithm on the western states power grid network of the United States [33]. We simulate the infection spreading process on the power grid network, which contains 4941 nodes. For each simulation run, 1, 2 or 3 infection sources are randomly chosen from the power grid network under Assumption 2, and the spreading process is simulated so that a total of 500 nodes are infected. For each value of $|S^*|$, 1000 simulation runs are performed. The simulation results are shown in Figs. 5 and 6(d), and Table II. We see that the MSEP-BFS algorithm outperforms the nSSE algorithm in every scenario. The average infection region covering percentage is above 59%.

C. Tests on Real Data

In order to get some insights in the performance of the MSEP-BFS algorithm in real infection spreads, we conduct two tests on real infection spreads data. We first apply the MSEP-BFS algorithm to to a network of nodes that represent the individuals who were infected with the SARS virus



(a)

(b)

(c)

(d)

Fig. 6. Histogram of the average error distances for various networks. We assume $\eta = 0$ and that the nSSE algorithm has prior knowledge of the number of infection sources. (a) Geometric trees. (b) Regular trees. (c) Small-world networks. (d) US power grid network.

during an epidemic in Singapore in the year 2003. The data is collected using contact tracing of patients [1], where an edge between two nodes indicate that there is some form of interaction or relationship between the individuals (e.g., they are family members, classmates, colleagues, or commuters who shared the same public transport system). A part of the SARS infection network corresponding to a cluster of 193 patients is shown in Fig. 7. We test the MSEP-BFS algorithm on the network in Fig. 7, assuming that there are at most $k_{\max} = 3$ infection sources. It turns out that the MSEP-BFS algorithm correctly estimates the number of infection sources to be one, and correctly identifies the real infection source.

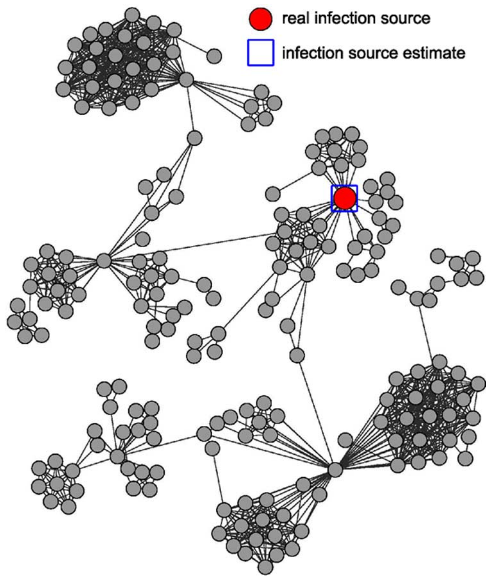


Fig. 7. Illustration of a cluster of the SARS infection network with a single source.

We next consider the Arizona-Southern California cascading power outages in 2011 [31]. The affected power network is represented by a graph where a node represents a key facility (substation or generating plant) affected by an outage, and an edge between two nodes indicate that there is a transmission line between these two facilities. The cascading outage starts with the loss of a single transmission line. However, as indicated in [31], this transmission line alone would not cause a cascading outage. After the loss of this transmission line, instantaneous power flow redistributions led to large voltage deviations, resulting in the nuclear units at San Onofre Nuclear Generating Station being taken off the power grid. The failures of these two key facilities together serve as the main causes of the subsequent cascading outages, so these two facilities are considered as the two infection sources. The main affected power network containing 48 facilities is shown in Fig. 8. We test the MSEP-BFS algorithm on the network in Fig. 8, and assume that there are at most $k_{\max} = 3$ infection sources. We can see that the MSEP-BFS algorithm correctly estimates the number of infection sources to

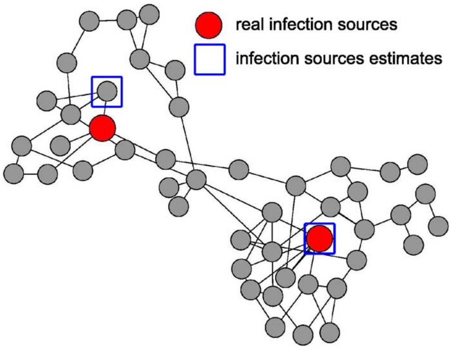


Fig. 8. Illustration of the main affected power network with two infection sources.

be two. We also found one of the sources correctly, and one estimate 1 hop away from the real source.

VI. Conclusion

We have derived estimators for the infection sources and regions when the number of infection sources is bounded but unknown a priori. The estimators are based only on knowledge of the infected nodes and their underlying network connections. We provide an approximation for the infection source estimator for the class of geometric trees, and when there are at most two sources in the network. We show that this estimator asymptotically correctly identifies the infection sources when the number of infected nodes grows large. We also propose an algorithm that estimates the number of source nodes, and identify them and their respective infection regions for general infection graphs. Simulation results on geometric trees, regular trees, small-world networks, the US power grid network, and experimental results on the SARS infection network and cascading power outages show that our proposed estimation procedure performs well in general, with an average error distance of less than 4. The estimation accuracy of the number of source nodes is over 65% in all the networks we consider, with the geometric tree networks having an accuracy of over 90%. Furthermore, the minimum infection region covering percentage is more than 59% for all networks. Our estimation procedure assumes only knowledge of the underlying network connections. In practical applications where more information about the infection process is available, a more accurate and intelligent guess of the number of infection sources can be made.

In this paper, we have adopted a simple SI infection model with homogeneous spreading rates, allowing us to derive analytical results that provide useful insights into infection source estimation for practical networks. However, this simplistic diffusion model does not adequately capture the real world dynamics of many networks. Future research includes the use of richer diffusion models that allow the inclusion of drifts and other dynamics in the infection spreading process, and tools from statistics to approximate optimal estimators for the infection sources. Our proposed algorithms find a set of nodes most likely to infect or influence a network, and are thus potentially useful for various practical applications. For example, our algorithm may be integrated with non-model-based consensus methods [34], [35] to design multi-agent control systems that uses only a small subset of agents as controllers. In cloud-centric media platforms [36], variants of our proposed algorithm may be used for intelligent content cache management. These are all areas of future research.

Appendix A
Proof of Theorem 1

Let nodes that are infected by source s_i be colored with color i, with $i = 1, \ldots, k$. Then a partition $\mathcal{A}_n$ corresponds to a coloring of the graph H_n, and to quantify the probability of a partition, it is sufficient to consider only infection sequences in the graph H_n. We have

$$P(\mathcal{A}_n \mid S, G_n) = \sum_{\sigma \in \Omega(H_n, S, \mathcal{A}_n)} P(\sigma \mid S), \tag{20}$$

where

$$\Omega(H_n, S, \mathcal{A}_n) = \{\sigma \in \Omega(H_n, S) : \sigma \cap A_{n,i} \\ \text{is an infection sequence, for all } i = 1, \ldots, k.\},$$

and $\sigma \cap A_{n,i}$ is the subsequence of σ containing only nodes that are in $A_{n,i}$.

Let $h = |H_n| - k$, and consider an infection sequence $\sigma = (\sigma_1, \ldots, \sigma_h) \in \Omega(H_n, S, \mathcal{A}_n)$. Let the set of edges connecting susceptible nodes to infected nodes be called the susceptible edge set. We have assumed that the infection times of susceptible nodes are independent and identically exponentially distributed. Therefore, given the infection sequence $\sigma_1, \ldots, \sigma_{l-1}$, the next edge along which the infection is spread is chosen uniformly at random from the susceptible edge set at time index $l-1$. Since H_n is a tree where all nodes except those in S have degree 2, after infection of a new node, the susceptible edge set size remains the same except in the case where the infected node is the last node to be infected amongst those on a path connecting two infection sources. In that case, the susceptible edge set size reduces by 2. Let J_σ be the set of indices of the last infected nodes on every path connecting infection sources. Letting $n_l = 1$ if $l \notin J_\sigma$ and 2 otherwise, we then have

$$\begin{aligned} P(\sigma \mid S) &= \prod_{l=1}^{h} n_l p_l(\sigma \mid H_n, S) \\ &= 2^p \prod_{l=1}^{h} p_l(\sigma \mid H_n, S) \end{aligned} \tag{21}$$

where p is the number of paths connecting infection sources, and

$$p_l(\sigma \mid H_n, S) = \left(\sum_{s \in S} \deg_{H_n}(s) - 2 \sum_{j \in J_\sigma} \mathbf{1}_{\{j<l\}} \right)^{-1}. \tag{22}$$

Choose two sources s_i and s_j and let m be the number of nodes in the path $\rho(s_i, s_j)$ connecting s_i and s_j, excluding the source nodes. Suppose that $r > \lceil m/2 \rceil$ nodes in this path have color i. Construct a new coloring $\mathcal{A}_n'$ so that $\lceil m/2 \rceil$ nodes in $\rho(s_i, s_j)$ closest to s_i have color i and the rest have color j. The rest of the nodes in $\mathcal{A}_n'$ have the same colors as that in $\mathcal{A}_n$. Each infection sequence $\sigma \in \Omega(H_n, S, \mathcal{A}_n)$ corresponds to an infection sequence $\sigma' \in \Omega(H_n, S, \mathcal{A}'_n)$, where the last $x = r - \lceil m/2 \rceil$ color-i nodes in σ become the last x color-j nodes in σ'. From (22), we have $p_l(\sigma \mid H_n, S) = p_l(\sigma' \mid H_n, S)$ for all l. Since $\binom{m}{\lceil m/2 \rceil} \geq \binom{m}{r}$, we have $|\Omega(H_n, S, \mathcal{A}'_n)| \geq |\Omega(H_n, S, \mathcal{A}_n)|$, therefore (20) yields $P(\mathcal{A}'_n \mid S, G_n) \geq P(\mathcal{A}_n \mid S, G_n)$.

The same argument can be repeated a finite number of times for all paths in H_n connecting infection sources. This shows that the estimator $\hat{\mathcal{A}}_n(S)$ is a Voronoi partition of G_n, and the proof is complete.

Appendix B
Proof of Lemma 2

To simplify notations, we write $T_u(s_1, s_2)$ as T_u, with the implicit understanding that all trees are defined w.r.t. $\{s_1, s_2\}$. The number of infection sequences can be found by counting the number of ways to form such a sequence. The $n-2$ slots in a sequence are occupied by nodes from $T_{s_i} \backslash \{s_i\}$, $i = 1, 2$, and $T_{\rho(u_1, u_m)}$. Therefore, we have

$$\begin{aligned} C(s_1, s_2 \mid G_n) &= (n-2)! \prod_{i=1}^{2} \frac{C(s_i \mid T_{s_i})}{(|T_{s_i}| - 1)!} \cdot \frac{R(u_1, u_m)}{\left|T_{\rho(u_1, u_m)}\right|!} \\ &= \frac{(n-2)!}{\left|T_{\rho(u_1, u_m)}\right|!} \cdot R(u_1, u_m) \cdot \prod_{\substack{v \in T_{s_i}, i=1,2 \\ v \neq s_1, s_2}} |T_v|^{-1}, \end{aligned}$$

where $R(u_i, u_j)$ for $i \leq j$ is the number of ways of permuting the nodes in $T_{\rho(u_i, u_j)}$ such that the infection sequence property is maintained, and the last equality follows from Lemma 1. To simplify the notations, for $1 \leq i \leq j \leq m$, let

$$J(u_i, u_j) = \prod_{v \in T_{\rho(u_i, u_j)} \backslash \rho(u_i, u_j)} |T_v|^{-1}.$$

For example, from Lemma 1, we have $C(u_i \mid T_{u_i}) = (|T_{u_i}| - 1)! J(u_i, u_i)$. In the following, we show that for $1 \leq i \leq j \leq m$,

$$R(u_i, u_j) = \left|T_{\rho(u_i, u_j)}\right|! \cdot q(u_i, u_j; s_1, s_2) \cdot J(u_i, u_j). \quad (23)$$

The proof proceeds by induction on $j - i$. If $j = i$, we have $R(u_i, u_i) = C(u_i \mid T_{u_i})$ and the claim follows from Lemma 1. Suppose that the claim (23) holds for all nodes u_k and u_p such that $p - k < j - i$. The number of permutations $R(u_i, u_i)$ can be computed by considering a sequence with $m = |T_{\rho(u_i, u_j)}|$ slots. The first slot can be filled with either u_i or u_j. Therefore, we have

$$\begin{aligned} R(u_i, u_j) &= (m-1)! \left(\frac{C(u_i \mid T_{u_i})}{(|T_{u_i}| - 1)!} \frac{R(u_{i+1}, u_j)}{\left|T_{\rho(u_{i+1}, u_j)}\right|!} \right. \\ &\qquad \left. + \frac{C(u_j \mid T_{u_j})}{(|T_{u_j}| - 1)!} \frac{R(u_i, u_{j-1})}{\left|T_{\rho(u_i, u_{j-1})}\right|!} \right) \\ &= (m-1)! \cdot (q(u_{i+1}, u_j; s_1, s_2) \\ &\qquad + q(u_i, u_{j-1}; s_1, s_2)) \prod_{v \in T_{\rho(u_i, u_j)} \backslash \rho(u_i, u_j)} \frac{1}{|T_v|}, \end{aligned}$$

where the penultimate equality follows from the inductive hypothesis and Lemma 1, and the last equality follows by noting that $J(u_i, u_i) J(u_{i+1}, u_j) = J(u_j, u_j) J(u_{i+1}, u_j) = J(u_i, u_j)$. The claim (23) now follows from (8). Finally, (9) follows by an inductive argument using (8), which we omit. The proof is now complete.

Appendix C
Proof of Lemma 3

The proof follows easily from Theorems 5 and 6 of [14]. Consider the infection spreading along a path in G_n. Let $\Pi(t)$ be the counting process of the number of infected nodes in this path. The process $\Pi(t)$ consists of exponentially distributed arrivals with rate 1, and at most one arrival with rate 2 if the path is between the two infection sources. Let $\Pi_1(t)$ be a unit rate Poisson process corresponding to the rate 1 arrivals. Then $\Pi_1(t) \leq \Pi(t) \leq \Pi_1(t) + 1$. From Theorem 6 of [14], we have for any positive $\gamma < 0.2$,

$$\begin{aligned} \mathbb{P}(\Pi(t) \leq t(1-\gamma)) &\leq \mathbb{P}(\Pi_1(t) \leq t(1-\gamma) - 1) \\ &\leq \exp\left(-\frac{1}{4} t \left(\gamma + \frac{1}{t} \right)^2 \right), \\ \mathbb{P}(\Pi(t) \geq t(1+\gamma)) &\leq \mathbb{P}(\Pi_1(t) \geq t(1+\gamma)) \leq \exp\left(-\frac{1}{4} t \gamma^2 \right). \end{aligned}$$

The rest of the proof is the same as that of Theorem 5 of [14], and the proof is complete.

Appendix D
Proof of Theorem 2

We first show that under (18), the interval (19) is non-empty. The condition (18) implies that

$$d_{\min} > \frac{3}{2} + \sqrt{2 d_{\max} \frac{c^2}{b^2} - \frac{1}{4}},$$

which after some algebraic manipulations yields

$$\begin{aligned} b^2 (d_{\min} - 1)(d_{\min} - 2) &> 2 c^2 d_{\max}, \\ 1 \leq \tfrac{c d_{\max}}{b(d_{\min} - 1)} &< \tfrac{b(d_{\min} - 2)}{2c}. \end{aligned}$$

Therefore (19) is a non-empty interval. Fix a δ in the interval. Then for all $\epsilon > 0$ sufficiently small, we have

$$\begin{aligned} \frac{b(d_{\min} - 1)(1+\delta)}{c d_{\max}} &> \frac{1}{1-\epsilon}, \\ \frac{b(d_{\min} - 2)}{2(1+\delta)c} &> \frac{1}{1-\epsilon}. \end{aligned}$$

Recall that t is the time from the start of the infection spreading to our observation of G_n. From Lemma 3, for each ϵ, there exists t_0 such that if $t \geq t_0$, we have

$$\frac{(d_{\min} - 1)(1+\delta) N_{\min}(t)}{d_{\max} N_{\max}(t)} > 1, \quad (24)$$

$$\frac{(d_{\min} - 2) N_{\min}(t)}{2(1+\delta) N_{\max}(t)} > 1. \quad (25)$$

We will make use of the two inequalities (24) and (25) extensively in the following proof steps. Let $\mathcal{E}_t$ be the event defined in Lemma 3. Then from Lemma 3, we have for $t \geq t_0$,

$$\begin{aligned}\mathbb{P}(\tilde{S} = S^* \mid S^*) &\geq \mathbb{P}(\tilde{S} = S^* \mid S^*, \mathcal{E}_t)\mathbb{P}(\mathcal{E}_t \mid S^*) \\ &\geq (1-\epsilon)\mathbb{P}(\tilde{S} = S^* \mid S^*, \mathcal{E}_t). \qquad (26)\end{aligned}$$

In the following, we show that $\mathbb{P}(\tilde{S} = S \mid S, \mathcal{E}_t) = 1$ for $t \geq t_0$. The proof then follows from (26) as ϵ can be chosen arbitrarily small.

To show that $\mathbb{P}(\tilde{S} = S \mid S, \mathcal{E}_t) = 1$ is equivalent to showing that with probability one, $\tilde{C}(S \,|\, G_n) > \tilde{C}(u_m, v_l \,|\, G_n)$, for all node pairs $u_m, v_l \in G_n$ such that at least one of them is not in S. Let u_0 and v_0 be the first nodes in $\rho(s_1, s_2)$ that are connected to u_m and v_l respectively. We divide the proof into two cases, depending on whether u_0 and v_0 are distinct or not, as shown in Figs. 9 and 10.

Suppose that $u_0 \neq v_0$. A typical network for this case is shown in Fig. 9, where m, l, n, p, and k are non-negative integers, and at least one of u_m and v_l is not in S, i.e., either $m + l > 0$ or $n + p > 0$. We let $u_0 = s_1$ if $n = 0$, and $v_0 = s_2$ if $p = 0$.

We will show that if either $m + l > 0$ or $n + p > 0$, we have for $t \geq t_0$,

$$\frac{\tilde{C}(s_1, s_2 \,|\, G_n)}{\tilde{C}(u_m, v_l \,|\, G_n)} = \frac{\tilde{C}(s_1, s_2 \,|\, G_n)}{\tilde{C}(u_0, v_0 \,|\, G_n)} \cdot \frac{\tilde{C}(u_0, v_0 \,|\, G_n)}{\tilde{C}(u_m, v_l \,|\, G_n)} > 1. \quad (27)$$

The proof follows by showing that $\tilde{C}(u_0, v_0 \,|\, G_n) \geq \tilde{C}(u_m, v_l \,|\, G_n)$, where strict inequality holds if $m + l > 0$, and $\tilde{C}(s_1, s_2 \,|\, G_n) \geq \tilde{C}(u_0, v_0 \,|\, G_n)$ with strict inequality holding if $n + p > 0$. From (17), we have [4]

$$\begin{aligned}&\frac{\tilde{C}(u_0, v_0 \,|\, G_n)}{\tilde{C}(u_m, v_l \,|\, G_n)} \\ &\quad = \frac{Q(u_0, v_0)}{Q(u_m, v_l)} \cdot \prod_{w \in \rho(u_m, u_1) \cup \rho(v_l, v_1)} |T_w(u_0, v_0)|^{-1} \\ &\quad = [2(1+\delta)]^{-(m+l)} \cdot \frac{\prod_{i=1}^{m+l+k+2} I_i^*(u_m, v_l)}{\prod_{i=1}^{k+2} I_i^*(u_0, v_0)} \\ &\qquad \cdot \prod_{w \in \rho(u_m, u_1) \cup \rho(v_l, v_1)} |T_w(u_0, v_0)|^{-1} \\ &\quad \geq [2(1+\delta)]^{-(m+l)} \cdot \prod_{i=1}^{m+l} I_i^*(u_m, v_l) \\ &\qquad \cdot \prod_{w \in \rho(u_m, u_1) \cup \rho(v_l, v_1)} |T_w(u_0, v_0)|^{-1} \\ &\quad \geq \left[\frac{\max\{|T_{u_0}(u_m, v_l)|, |T_{v_0}(u_m, v_l)|\}}{2(1+\delta) \cdot \max\{|T_{u_1}(u_0, v_0)|, |T_{v_1}(u_0, v_0)|\}}\right]^{m+l} \\ &\quad \geq \left[\frac{(d_{\max} - 2)N_{\min}(t) + 1}{2(1+\delta) \cdot N_{\max}(t)}\right]^{m+l} \\ &\quad > 1,\end{aligned}$$

if $m + l > 0$. The first inequality follows because $I_{m+l+i}^*(u_m, v_l) \geq I_i^*(u_0, v_0)$ for $i = 1, \ldots, k+2$, and the last inequality follows from (25) when $t \geq t_0$.

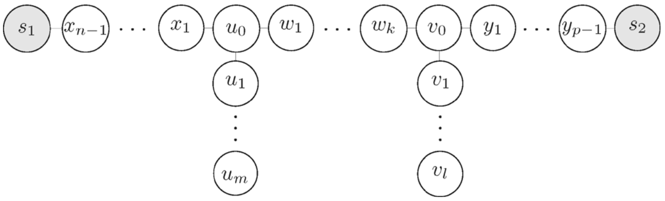


Fig. 9. Illustration of the network structure when $u_0 \neq v_0$. Not all nodes are shown.

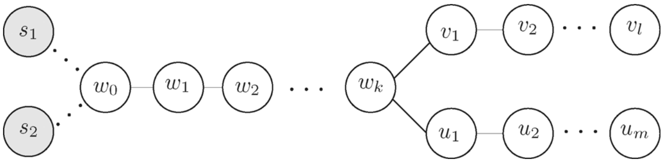


Fig. 10. Illustration of the case where $u_0 = v_0 = w_0$.

Let $\psi = \deg_G(s_1) + \deg_G(s_1)$. We have for $t \geq t_0$,

$$\begin{aligned}&\frac{\tilde{C}(s_1, s_2 \,|\, G_n)}{\tilde{C}(u_0, v_0 \,|\, G_n)} \\ &\quad = \frac{Q(s_1, s_2)}{Q(u_0, v_0)} \cdot \prod_{w \in \rho(s_1, x_1) \cup \rho(y_1, s_2)} |T_w(u_0, v_0)| \\ &\quad = [2(1+\delta)]^{n+p} \cdot \frac{\prod\limits_{i=1}^{k+2} I_i^*(u_0, v_0)}{\prod\limits_{i=1}^{n+p+k+2} I_i^*(s_1, s_2)} \\ &\qquad \cdot \prod_{w \in \rho(s_1, x_1) \cup \rho(y_1, s_2)} |T_w(u_0, v_0)| \\ &\quad \geq [2(1+\delta)]^{n+p} \cdot \prod_{i=k+3}^{n+p+k+2} I_i^*(s_1, s_2)^{-1} \\ &\qquad \cdot \prod_{w \in \rho(s_1, x_1) \cup \rho(y_1, s_2)} |T_w(u_0, v_0)| \\ &\quad \geq \left[\frac{2(1+\delta) \cdot \min\{|T_{s_1}(u_0, v_0)|, |T_{s_2}(u_0, v_0)|\}}{\psi N_{\max}(t) + 2}\right]^{n+p} \\ &\quad \geq \left[\frac{(1+\delta)(d_{\min} - 1) \cdot N_{\min}(t) + 1 + \delta}{d_{\max} N_{\max}(t) + 1}\right]^{n+p} \\ &\quad > 1,\end{aligned}$$

where the first inequality follows because $I_i^*(u_0, v_0) \geq I_i^*(s_1, s_2)$ for $i = 1, \ldots, k+2$, and the last inequality follows from (24) if $n + p > 0$. The bound (27) is now proved.

We next consider the case where $u_0 = v_0 = w_0$ in Fig. 10, where k, m and l are non-negative integers. When $t \geq t_0$, we have the following bounds, which are straight forward to verify and whose proofs are omitted here.

i) $I_i^*(u_m, v_l) \geq (\psi - 2)N_{\min}(t) + 2 \geq (d_{\min} - 2)N_{\min}(t)$ for $i = 1, \ldots, d(u_m, v_l) + 1$,

ii) $I_i^*(s_1, s_2) \leq \psi N_{\max}(t) + 2 \leq 2d_{\max}N_{\max}(t) + 2$ for all $i = 1, \ldots, d(s_1, s_2) + 1$,

iii) $|T_{w_i}(u_m, v_l)| \geq (\psi - 2)N_{\min}(t) + 2 \geq (d_{\min} - 2)N_{\min}(t)$ for all $i = 1, \ldots, k-1$,

[4] We define products over empty sets to be 1.

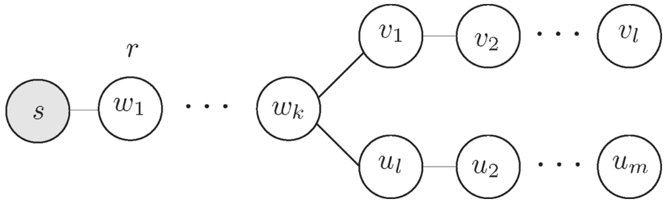


Fig. 11. A typical network for a single source tree.

iv) $|T_w(u_m, v_l)| \geq (d_{\min} - 1)N_{\min}(t) + 1$ for all $w \in \rho(s_1, s_2)$,

v) $|T_{w_i}(s_1, s_2)| \leq N_{\max}(t)$ for all $i = 1, \ldots, k-1$, and

vi) $|T_w(s_1, s_2)| \leq N_{\max}(t)$ for all $w \in \rho(u_m, v_l)$.

The above bounds yield

$$\begin{aligned}
&\frac{\tilde{C}(s_1, s_2 \,|\, G_n)}{\tilde{C}(u_m, v_l \,|\, G_n)} \\
&\quad= \frac{Q(s_1, s_2)}{Q(u_m, v_l)} \cdot \frac{\prod_{w \in G_n \backslash \rho(u_m, v_l)} |T_w(u_m, v_l)|}{\prod_{w \in G_n \backslash \rho(s_1, s_2)} |T_w(s_1, s_2)|} \\
&\quad= (2(1+\delta))^{d(s_1,s_2) - d(u_m,v_l)} \cdot \frac{\prod_{i=1}^{d(u_m,v_l)+1} I_i^*(u_m, v_l)}{\prod_{i=1}^{d(s_1,s_2)+1} I_i^*(s_1, s_2)} \\
&\qquad \cdot \frac{\prod_{i=1}^{k-1} |T_{w_i}(u_m, v_l)| \prod_{w \in \rho(s_1,s_2)} |T_w(u_m, v_l)|}{\prod_{i=1}^{k-1} |T_{w_i}(s_1, s_2)| \prod_{w \in \rho(u_m,v_l)} |T_w(s_1, s_2)|} \\
&\quad= \prod_{i=1}^{k-1} \frac{|T_{w_i}(u_m, v_l)|}{|T_{w_i}(s_1, s_2)|} \\
&\qquad \cdot (2(1+\delta))^{-d(u_m,v_l)-1} \frac{\prod_{i=1}^{d(u_m,v_l)+1} I_i^*(u_m, v_l)}{\prod_{w \in \rho(u_m,v_l)} |T_w(s_1, s_2)|} \\
&\qquad \cdot (2(1+\delta))^{d(s_1,s_2)+1} \frac{\prod_{w \in \rho(s_1,s_2)} |T_w(u_m, v_l)|}{\prod_{i=1}^{d(s_1,s_2)+1} I_i^*(s_1, s_2)} \\
&\quad\geq \left[\frac{(d_{\min} - 2)N_{\min}(t)}{N_{\max}(t)}\right]^{k-1} \\
&\qquad \cdot \left[\frac{(d_{\min} - 2)N_{\min}(t)}{2(1+\delta)N_{\max}(t)}\right]^{d(u_m,v_l)+1} \\
&\qquad \cdot \left[\frac{(1+\delta)((d_{\min} - 1)N_{\min}(t) + 1)}{d_{\max}N_{\max}(t) + 1}\right]^{d(s_1,s_2)+1} \\
&\quad> 1,
\end{aligned}$$

where the last inequality follows from (24) and (25). The theorem is now proved.

Appendix E
Proof of Theorem 3

Let t be the elapsed time from the start of an infection spreading from a single s to the time we observe G_n. We wish to show that Algorithm TSE estimates as sources s and one of its neighbors with probability (conditioned on s being the infection source) converging to 1 as $t \to \infty$. This is equivalent to showing that for t sufficiently large, and for each pair of nodes $u_m, v_l \in G_n$ where either $d(u_m, s) > 1$ or $d(v_l, s) > 1$, there exists a neighbor r of s such that $\tilde{C}(s, r \,|\, G_n) > \tilde{C}(u_m, v_l \,|\, G_n)$.

A typical network is shown in Fig. 11, where k, m and l are non-negative integers. If m, l and k are positive, we let r be the neighbor of s that lies on the path connecting s to u_m (i.e., the node w_1 in Fig. 11). If m and l are positive and $k = 0$, then r is chosen to be either u_1 or v_1. If $m = 0$, we must have $k > 0$ so that $w_k = u_m$ and $r = w_1$. A similar remark applies for the case $l = 0$. Note that $m + l > 0$. For t sufficiently large, we have

$$\begin{aligned}
&\frac{\tilde{C}(s, r \,|\, G_n)}{\tilde{C}(u_m, v_l \,|\, G_n)} \\
&\quad= \frac{Q(s, r)}{Q(u_m, v_l)} \cdot \frac{\displaystyle\prod_{w \in G_n \backslash \rho(u_m, v_l)} |T_w(u_m, v_l)|}{\displaystyle\prod_{w \in G_n \backslash \{s, r\}} |T_w(s, r)|} \\
&\quad= [2(1+\delta)]^{1-(m+l)} \cdot \frac{\prod_{i=1}^{m+l+1} I_i^*(u_m, v_l)}{\prod_{i=1}^{2} I_i^*(s, r)} \\
&\qquad \cdot \frac{\prod_{w \in \rho(s, w_{k-1})} |T_w(u_m, v_l)|}{\prod_{i=2}^{k-1} |T_{w_i}(s, r)| \cdot \displaystyle\prod_{w \in \rho(u_m, v_l)} |T_w(s, r)|} \\
&\quad= [2(1+\delta)]^{1-(m+l)} \cdot \prod_{i=1}^{m+l} I_i^*(u_m, v_l) \\
&\qquad \cdot \frac{\prod_{i=1}^{k-1} |T_{w_i}(u_m, v_l)|}{\prod_{i=2}^{k-1} |T_{w_i}(s, r)| \cdot \displaystyle\prod_{w \in \rho(u_m, v_l)} |T_w(s, r)|} \\
&\quad\geq [2(1+\delta)]^{1-(m+l)} \cdot |T_{w_k}(u_m, v_l)|^{m+l} \\
&\qquad \cdot \frac{|T_s(u_m, v_l)|^{k-1}}{N_{\max}(t)^{k-2} \cdot N_{\max}(t)^{m+l+1}} \\
&\quad\geq [2(1+\delta)]^{k} \cdot \left[\frac{(d_{\min} - 1)N_{\min}(t)}{2(1+\delta) \cdot N_{\max}(t)}\right]^{m+l+k-1} \\
&\quad> 1,
\end{aligned}$$

where the last inequality follows from (25) and Lemma 3. The proof of the theorem is now complete.

References

[1] K.-T. Goh, J. Cutter, B.-H. Heng, S. Ma, B. K. W. Koh, C. Kwok, C.-M. Toh, and S.-K. Chew, "Epidemiology and control of SARS in Singapore," *Anna. Acad. Med. Singapore*, vol. 35, no. 5, pp. 301–316, 2006.

[2] C. Scoglio, W. Schumm, P. Schumm, T. Easton, S. R. Chowdhury, A. Sydney, and M. Youssef, "Efficient mitigation strategies for epidemics in rural regions," *PLoS ONE*, vol. 5, no. 7, 2010.

[3] M. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Phys. Rev. E*, vol. 69, no. 2, 2004.

[4] J. O. Kephart and S. R. White, "Directed-graph epidemiological models of computer viruses," in *Proc. IEEE Comput. Soc. Symp. Res. Security Privacy*, 1991, pp. 343–359.

[5] L. Han, S. Han, Q. Deng, J. Yu, and Y. He, "Source tracing and pursuing of network virus," in *Proc. 8th IEEE Int. Conf. Comput. Inf. Technol. Workshops*, 2008, pp. 230–235.

[6] J. Weng, E.-P. Lim, J. Jiang, and Q. He, "Twitterrank: Finding topic-sensitive influential twitterers," in *Proc. 3rd ACM Int. Conf. Web Search Data Mining*, 2010, pp. 261–270.

[7] E. Bakshy, J. M. Hofman, W. A. Mason, and D. J. Watts, "Everyone's an influencer: Quantifying influence on Twitter," in *Proc. 4th ACM Int. Conf. Web Search Data Mining*, 2011, pp. 65–74.

[8] S.-H. Lim, S.-W. Kim, S. Park, and J. H. Lee, "Determining content power users in a blog network: An approach and its applications," *IEEE Trans. Syst., Man, Cybern. A*, vol. 41, no. 5, pp. 853–862, May 2011.

[9] L. Akritidis, D. Katsaros, and P. Bozanis, "Identifying the productive and influential bloggers in a community," *IEEE Trans. Syst., Man, Cybern. C*, vol. 41, no. 5, pp. 759–764, May 2011.

[10] C. Moore and M. E. J. Newman, "Epidemics and percolation in small-world networks," *Phys. Rev. E*, vol. 61, pp. 5678–5682, 2000.

[11] M. E. J. Newman, "Spread of epidemic disease on networks," *Phys. Rev. E*, vol. 66, no. 1, p. 016 128+, Jul. 2002.

[12] P. D. ONeill, "A tutorial introduction to Bayesian inference for stochastic epidemic models using Markov chain Monte Carlo methods," *Math. Biosci.*, vol. 180, no. 1–2, pp. 103–114, 2002.

[13] A. Ganesh, L. Massouli, and D. Towsley, "The effect of network topology on the spread of epidemics," presented at the IEEE INFOCOM, 2005.

[14] D. Shah and T. Zaman, "Rumors in a network: Who's the culprit?," *IEEE Trans. Inf. Theory*, vol. 57, no. 8, pp. 5163–5181, 2011.

[15] N. Bailey*, The Mathematical Theory of Infectious Diseases and Its Applications*. Duxbury, MA, USA: Griffin, 1975.

[16] L. J. Allen, "Some discrete-time SI, SIR, and SIS epidemic models," *Math. Biosci.*, vol. 124, no. 1, pp. 83–105, 1994.

[17] M. Barthelemy, A. Barrat, R. Pastor-Satorras, and A. Vespignani, "Velocity and hierarchical spread of epidemic outbreaks in scale-free networks," *Phys. Rev. Lett.*, vol. 92, p. 178701, 2004.

[18] G. Yan, T. Zhou, J. Wang, Z. Q. Fu, and B. H. Wang, "Epidemic spread in weighted scale-free networks," *Chinese Phys. Lett.*, vol. 22, no. 2, p. 510, 2005.

[19] T. Zhou, G. Yan, and B.-H. Wang, "Maximal planar networks with large clustering coefficient and power-law degree distribution," *Phys. Rev. E*, vol. 71, p. 046141, 2005.

[20] T. Zhou, J.-G. Liu, W.-J. Bai, G. Chen, and B.-H. Wang, "Behaviors of susceptible-infected epidemics on scale-free networks with identical infectivity," *Phys. Rev. E*, vol. 74, p. 056109, 2006.

[21] S. Tang and W. Li, "An epidemic model with adaptive virus spread control for wireless sensor networks," *Int. J. Security Netw.*, vol. 6, no. 4, pp. 201–210, 2011.

[22] T. Zhao and A. Nehorai, "Distributed sequential Bayesian estimation of a diffusive source in wireless sensor networks," *IEEE Trans. Signal Process.*, vol. 55, no. 4, pp. 1511–1524, Apr. 2007.

[23] E. B. Fox, J. W. Fisher, and A. S. Willsky, "Detection and localization of material releases with sparse sensor configurations," *IEEE Trans. Signal Process.*, vol. 55, no. 5, pp. 1886–1898, May 2007.

[24] W. P. Tay, J. N. Tsitsiklis, and M. Z. Win, "Bayesian detection in bounded height tree networks," *IEEE Trans. Signal Process.*, vol. 57, no. 10, pp. 4042–4051, Oct. 2009.

[25] W. P. Tay, J. N. Tsitsiklis, and M. Z. Win, "On the impact of node failures and unreliable communications in dense sensor networks," *IEEE Trans. Signal Process.*, vol. 56, no. 6, pp. 2535–2546, Jun. 2008.

[26] P. Bianchi, M. Debbah, M. Maida, and J. Najim, "Performance of statistical tests for single-source detection using random matrix theory," *IEEE Trans. Inf. Theory*, vol. 57, no. 4, pp. 2400–2419, Apr. 2011.

[27] S. Aldalahmeh and M. Ghogho, "Robust distributed detection, localization and estimation of a diffusive target in clustered wireless sensor networks," presented at the IEEE Int. Conf. Acoust., Speech, Signal Process., 2011.

[28] D. Kempe, J. Kleinberg, and E. Tardos, "Maximizing the spread of influence through a social network," in *Proc. 9th ACM SIGKDD Int. Conf. Knowledge Discovery Data Mining*, 2003, pp. 137–146.

[29] M. Kimura, K. Saito, R. Nakano, and H. Motoda, "Extracting influential nodes on a social network for information diffusion," *Data Min. Knowl. Discov.*, vol. 20, pp. 70–97, 2010.

[30] G. Brightwell and P. Winkler, "Counting linear extensions is #P-complete," in *Proc. 23rd Annu. ACM Symp. Theory Comput.*, 1991, pp. 175–181.

[31] U.S. Federal Energy Regulatory Commission and North American Electric Reliability Corporation, Arizona-Southern California Outages on September 8, 2011: Causes and Recommendations, 2012.

[32] G. Sabidussi, "The centrality index of a graph," *Psychometrika*, vol. 31, no. 4, pp. 581–603, 1966.

[33] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, no. 6684, pp. 440–442, 1998.

[34] G. Hu, "Robust consensus tracking for an integrator-type multi-agent system with disturbances and unmodelled dynamics," *Int. J. Contr.*, vol. 84, no. 1, pp. 1–8, 2011.

[35] G. Hu, "Robust consensus tracking of a class of second-order multi-agent dynamic systems," *Syst. Contr. Lett.*, vol. 61, no. 1, pp. 134–142, 2012.

[36] Y. Wen, G. Shi, and G. Wang, "Designing an inter-cloud messaging protocol for content distribution as a service (CoDaas) over future internet," in *Proc. Int. Conf. Future Internet Technol.*, 2011.

Wuqiong Luo (S'12) received the B.Eng. degree in electrical and electronic engineering (with first class hons.) from Nanyang Technological University, Singapore, in 2010. He is currently working toward the Ph.D. degree in electrical and electronic engineering at Nanyang Technological University.

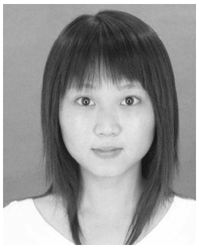
His research interests are in source estimation and identification in communication networks.

Mr. Luo was coawarded the Best Student Paper Award at the 46th Asilomar Conference on Signals, Systems, and Computers.

Wee Peng Tay (S'06–M'08) received the B.S. degree in electrical engineering and mathematics, and the M.S. degree in electrical engineering from Stanford University, Stanford, CA, USA, in 2002. He received the Ph.D. degree in electrical engineering and computer science from the Massachusetts Institute of Technology, Cambridge, MA, USA, in 2008.

He is currently an Assistant Professor in the School of Electrical and Electronic Engineering at Nanyang Technological University, Singapore. His research interests include distributed decision making, data fusion, distributed algorithms, communications in ad hoc networks, machine learning, and applied probability.

Dr. Tay received the Singapore Technologies Scholarship in 1998, the Stanford University President's Award in 1999, and the Frederick Emmons Terman Engineering Scholastic Award in 2002. He is the coauthor of the best student paper award at the 46th Asilomar Conference on Signals, Systems, and Computers.



Mei Leng (S'07–M'10) received the B.Eng. degree from University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2005, and the Ph.D. degree from The University of Hong Kong, Hong Kong, in 2011.

She is currently a Research Fellow at the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. Her current research interests include statistical signal processing, optimization, machine learning, as well as Bayesian analysis, with applications to wireless sensor networks and wireless communication systems.