Information Source Detection in Networks

by

Kai Zhu

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved October 2015 by the
Graduate Supervisory Committee:

Lei Ying, Chair
Ying-Cheng Lai
Huan Liu
Paulo Shakarian

ARIZONA STATE UNIVERSITY

December 2015

ABSTRACT

The purpose of information source detection problem (or called rumor source detection) is to identify the source of information diffusion in networks based on available observations like the states of the nodes and the timestamps at which nodes adopted the information (or called infected). The solution of the problem can be used to answer a wide range of important questions in epidemiology, computer network security, etc. This dissertation studies the fundamental theory and the design of efficient and robust algorithms for the information source detection problem.

For tree networks, the maximum a posterior (MAP) estimator of the information source is derived under the independent cascades (IC) model with a complete snapshot and a Short-Fat Tree (SFT) algorithm is proposed for general networks based on the MAP estimator. Furthermore, the following possibility and impossibility results are established on the Erdos-Renyi (ER) random graph: (i) when the infection duration $< \frac{2}{3}t_u$, SFT identifies the source with probability one asymptotically, where $t_u = \left\lceil \frac{\log n}{\log \mu} \right\rceil + 2$ and μ is the average node degree, (ii) when the infection duration $> t_u$, the probability of identifying the source approaches zero asymptotically under any algorithm; and (iii) when infection duration $< t_u$, the breadth-first search (BFS) tree starting from the source is a fat tree. Numerical experiments on tree networks, the ER random graphs and real world networks show that the SFT algorithm outperforms existing algorithms.

In practice, other than the nodes' states, side information like partial timestamps may also be available. Such information provides important insights of the diffusion process. To utilize the partial timestamps, the information source detection problem is formulated as a ranking problem on graphs and two ranking algorithms, cost-based ranking (CR) and tree-based ranking (TR), are proposed. Extensive experimental evaluations of synthetic data of different diffusion models and real world data demonstrate the effectiveness and robustness of CR and TR compared with existing algorithms.

To my parents and grandparents.

To my wife and best friend, Tu.

To my teachers and mentors.

ACKNOWLEDGMENTS

First and foremost, I would like to express my sincere gratitude to my advisor, Prof. Lei Ying for his guidance, encouragements and patience. His profound intuition, sharp thinking and deep insight have changed my way of tackling problems. It was him who taught me how to write an academic paper by revising my draft line by line. He has been a mentor, colleague and friend to me. I still vividly recall the day when Prof. Ying first introduced the information source detection problem to me. The effort to tackle this problem flourished into several academic papers and become my dissertation. His guidance has made this a thoughtful and rewarding journey.

In addition, I would like to thank Prof. Ying-Cheng Lai, Prof. Huan Liu, and Prof. Paulo Shakarian for serving on my committee and providing valuable insights and suggestions.

I would like to thank all my collaborators, colleagues and friends: Francois Baccelli, Jun Chen, Zhen Chen, Sabarna Choudhuri, Yong Guan, Bruce Hajek, Xiaohan Kang, Chong Li, Shihuan Liu, Xin Liu, Tien V Nguyen, Ming Ouyang, R Srikant, Sundar Subramanian, Jian Tan, Weina Wang, Rui Wu, Xinzhou Wu, Yu Wu, Jiaming Xu, Li Zhang, Zhengyu Zhang, Shan Zhou. It was a pleasure to have the privilege to work with all these talented researchers.

Last but not the least, I would like to thank my wife and best friend, Tu Lu for her understanding and love during the past five years. Her encouragement and support was in the end what makes this dissertation possible. My parents and grandparents receive my deepest gratitude for providing me the foundation upon everything.

TABLE OF CONTENTS

Page

LIST OF TABLES vii

LIST OF FIGURES ix

CHAPTER

1 INTRODUCTION 1

1.1 Motivation and Background 1

1.2 Our Contribution 4

1.3 Thesis Outline 6

2 INFORMATION SOURCE DETECTIONS: TREES AND BEYOND 7

2.1 Model and Algorithm 9

2.1.1 Model 9

2.1.2 The Short-Fat Tree Algorithm 11

2.2 Main Results 15

2.2.1 Main Result 1 (The MAP Estimator on Tree Networks) 15

2.2.2 Main Result 2 (Detection with Probability One on the ER Graph) 16

2.2.3 Main Result 3 (The Fat Tree Result on the ER Graph) 17

2.2.4 Main Result 4 (The Impossibility Result on the ER Graph) . 18

2.3 Performance Evaluation 20

2.3.1 Algorithms 20

2.3.2 Evaluation Metrics 21

2.3.3 Binomial Trees 23

2.3.4 The ER Random Graph 24

2.3.5 The Internet Autonomous System Network 25

2.3.6 Running Time vs Performance 26

CHAPTER Page

3 INFORMATION SOURCE DETECTION WITH PARTIAL TIMESTAMPS 27

3.1 A Ranking Approach for Source Localization 30

3.1.1 Quadratic Cost and Sample Path Approach 32

3.2 EIF: A Greedy Algorithm 34

3.3 Cost-Based and Tree-Based Ranking 39

3.4 Experimental Evaluation 41

3.4.1 Performance of EIF on a Small Network 41

3.4.2 Comparison with Other Algorithms 42

3.4.3 The Impact of Timestamp Distribution 46

3.4.4 The Impact of the Diffusion Model 48

3.4.5 The Impact of Network Topology 51

3.4.6 Weibo Data Evaluation 52

3.5 Extensions 55

3.6 Additional Experimental Evaluation 56

3.6.1 Comparison to Lappas' Algorithm (Lappas *et al.*, 2010) 56

3.6.2 Scalability 57

3.6.3 Normalized Rank 58

4 CONCLUSIONS 66

REFERENCES 67

APPENDIX

A PROOFS OF CHAPTER 2 70

B PROOFS OF CHAPTER 3 121

LIST OF TABLES

Table Page

3.1 The 10%-accuracy under Different Source Localization Algorithms with 50% Timestamps 27

3.2 The Timestamps on the Spreading Tree in the 3rd Iteration 35

3.3 The Costs of the Modified Shortest Paths 36

3.4 Statistics of Extracted Tweet Cascades 53

3.5 10%-accuracy for Different Tweet Cascade Sizes 54

3.6 Normalized Rank for Different Distributions and Sizes of Timestamps on the IAS Network when $\eta = 1$ 60

3.7 Normalized Rank for Different Distributions and Sizes of Timestamps on the IAS Network when $\eta = 10$ 60

3.8 Normalized Rank for Different Distributions and Sizes of Timestamps on the IAS Network when $\eta = 100$ 61

3.9 Normalized Rank for Different Distributions and Sizes of Timestamps on the PG Network when $\eta = 1$ 61

3.10 Normalized Rank for Different Distributions and Sizes of Timestamps on the PG Network when $\eta = 10$ 62

3.11 Normalized Rank for Different Distributions and Sizes of Timestamps on the PG Network when $\eta = 100$ 62

3.12 Normalized Rank for Different Distributions and Sizes of Timestamps on the IAS Network under the IC Model 63

3.13 Normalized Rank for Different Distributions and Sizes of Timestamps on the IAS Network under the SpikeM Model 63

3.14 Normalized Rank for Different Distributions and Sizes of Timestamps on the PG Network under the IC Model 64

3.15 Normalized Rank for Different Distributions and Sizes of Timestamps on the PG Network under the SpikeM Model 64

3.16 Normalized Rank as the Number of Removed Edges Increases in the IAS Network 65

3.17 Normalized Rank for Different Tweet Cascade Sizes on the Weibo Dataset 65

LIST OF FIGURES

Figure Page

2.1 Summary of the Main Results 8

2.2 An Example of the Short-Fat Tree Algorithm 13

2.3 A Pictorial Example of $\mathcal{Z}_t^t(u)$ in BFS Tree $T^\dagger$ 17

2.4 Performance in the Binomial Trees 21

2.5 Performance in the ER Random Graph 22

2.6 Performance in the IAS Graph 25

2.7 Detection Rate versus Running Time in the ER Random Graph 26

3.1 An Example Illustrating Available Information and a Spreading Tree . . 31

3.2 An Example for Illustrating Step 4 and Step 5 of EIF. 38

3.3 The Approximation Ratio of TR 42

3.4 Comparison with Existing Algorithms with 50% Timestamps. 45

3.5 The Impacts of the Distribution and Size of Timestamps. 49

3.6 The Performance of CR, TR and GAU under Different Diffusion Models 50

3.7 The γ%-accuracy as the Number of Removed Edges Increases 51

3.8 Performance on Weibo Data 52

3.9 An Example of Extensions with Direction Information 55

3.10 The Performance Comparison to the Lappas' Algorithm 56

3.11 Execution Time in the IAS Network under the IC Model. 57

A.1 A Pictorial Description of the Positions of Nodes a, u, w and ξ 75

A.2 A Pictorial Example of $\mathcal{Z}_t^t(u)$ in BFS Tree $T^\dagger$ 80

A.3 A Pictorial Example of Upper Bounds of H 93

Chapter 1

INTRODUCTION

1.1 Motivation and Background

Diffusion processes in networks refer to the spread of information throughout the networks, and have been widely used to model many real-world phenomena such as the outbreak of epidemics, the spreading of gossips over online social networks, the spreading of computer virus over the Internet, and the adoption of innovations. A large body of existing works on information diffusion focused on the influence maximization problem (see e.g., (Kempe *et al.*, 2003; Chen *et al.*, 2010, 2009; Goyal *et al.*, 2011)) and inferring topological properties of information cascades (see e.g., (Gruhl *et al.*, 2004; Sadikov *et al.*, 2011; Myers *et al.*, 2012)).

In this dissertation, we are interested in the reverse of the diffusion problem: given a snapshot of the diffusion process, can we tell which node is the source of the diffusion? This source detection problem has a wide range of applications. In epidemiology, identifying patient zero can provide important information about the disease. For example, in the Cholera outbreak in London in 1854 (Snow, 1854), the spreading pattern of the Cholera suggested that the water pump located at the center of the spreading was likely to be the source. Later, it was confirmed that the Cholera indeed spreads via contaminated water. In online social networks, identifying the source can reveal the user who started a rumor or the user who first announced certain breaking news. For rumors, rumor source detection helps hold people accountable for their online behaviors; and for news, the news source can be used to evaluate the credibility of the news. While locating the information source has these important applications in practice, the problem is difficult to solve, in particular, in complex networks.

We call this problem *information source detection* problem. Informally speaking, information source detection refers to the problem of identifying a node in the network that provides the best explanation of the observed diffusion. (Shah and Zaman, 2011) is one of the first papers that study the information source detection problem, in which a new graph centrality called rumor centrality was proposed and proved to be the maximum likelihood estimator (MLE) on regular trees under the susceptible-infected (SI) model. In addition, the detection probability (the probability that the estimator is the source) for regular trees was proved to be greater than zero and the detection probability for geometric trees approaches one asymptotically as the increase of the spreading time. Later, (Shah and Zaman, 2012) quantified the detection probability of the rumor centrality on general random trees.

The rumor centrality has been further studied under different scenarios: 1) (Luo *et al.*, 2013) extended the rumor centrality to multiple sources and showed that the detection probability goes to one as the number of infected nodes increases for geometric trees when there are at most two sources; 2) (Karamchandani and Franceschetti, 2013) proved a similar performance guarantee for the single source case when only a subset of infected nodes are observed; 3) (Dong *et al.*, 2013) studied the detection probability when the prior knowledge of suspect nodes is available in the single source detection problem for trees; 4) (Wang *et al.*, 2014) analyzed the detection probability of the rumor centrality for tree networks when there are multiple observations of independent diffusion processes from the same source.

(Zhu and Ying, 2014a) proposed the sample path based approach for the single source detection problem. Define the infection eccentricity of a node to be the maximum distance between the node and the infected nodes. (Zhu and Ying, 2014a) proved that on tree networks, under the homogeneous susceptible-infected-recovered (SIR) model, the root of the most likely sample path is a node with the minimum infection eccentricity (a Jordan infection center), which is within a constant distance to the actual source with a high probability. The approach has been extended to several directions: 1) (Zhu and Ying, 2014b) extended the approach to the case with partial observations and under the heterogeneous SIR model; 2) (Chen *et al.*, 2014) extended the analysis to multiple sources under the SIR model and proved that the distance between the estimator and its closest actual source is bounded by a constant with a high probability in tree networks; 3) (Luo and Tay, 2013a,b) proved that the Jordan infection centers are the optimal sample path estimators under the SI model (Luo and Tay, 2013a) and the susceptible-infected-susceptible (SIS) model (Luo and Tay, 2013b) for tree networks, respectively.

Besides the rumor centrality and the Jordan infection center, several other heuristic algorithms based on a single snapshot of the network have been proposed in the literature: 1) (Lappas *et al.*, 2010) studied a similar problem under the IC model (Goldenberg *et al.*, 2001) to minimize the l1 distance between the expected states and observed states of the nodes. A dynamic programming algorithm was proposed to solve the problem for tree networks and a Steiner tree heuristic was used for general networks; 2) (Prakash *et al.*, 2012) proposed an algorithm called NETSLEUTH which ranks the nodes according to an eigen vector based metric under the SI model. The algorithm was designed based on the Minimum Description Length principle; 3) (Lokhov *et al.*, 2014) proposed a dynamic message passing algorithm based on the mean field approximation of the maximum likelihood estimation (MLE) of the source.

In addition, there exist several other algorithms which tackled the problem under the assumption that a subset of the infection timestamps are known: 1) (Pinto *et al.*, 2012) solved the MLE problem with partial timestamps for tree networks and extended the algorithm to general networks using a BFS tree heuristic. The algorithm is similar to CR in Chapter 3 in spirit, but uses the BFS tree as the spreading tree from a given infected node. In the experiment evaluation on the Internet autonomous systems network, not only the performance of the algorithm is worse than ours, the gap also increases significantly as the amount of timestamps increases. We conjecture this is because the spreading trees constructed by our Earliest-Infection-First algorithm is far more accurate than the BFS trees; 2) (Agaskar and Lu, 2013) proposed a simulation based Monte Carlo algorithm which utilizes the states of the sparsely placed observers within a fixed time window. The approach, however, requires the infection time distributions of all edges, which are difficult to obtain in practice; 3) (Zejnilovic *et al.*, 2013) obtained sufficient conditions on the number of timestamps needed to locate the source correctly under the deterministic slotted SI models. The model considered in this dissertation is probabilistic which is far more challenging than deterministic ones.

Several related works also investigated similar problems: (1) detecting the first adopter of an innovation based on game theory (Subramanian and Berry, 2012), in which the maximum likelihood estimator is derived but the computational complexity of finding the estimator is exponential in the number of nodes; (2) distinguishing epidemic infection from random infection under the SI model (Milling *et al.*, 2012); (3) geospatial abduction which deals with reasoning certain locations in a two-dimensional geographical area that can explain observed phenomena (Shakarian *et al.*, 2011; Shakarian and Subrahmanian, 2011).

1.2 Our Contribution

Despite significant efforts and successes over last few years, theoretical guarantees of source localization algorithms were established only for tree networks due to the complexity of the problem. In Chapter 2, we derive the MAP estimator of the source for tree networks and propose a SFT algorithm for general networks based on the MAP estimator. The algorithm selects the Jordan infection center (Zhu and Ying, 2014a) and breaks ties according the degree of boundary infected nodes. Loosely speaking, the algorithm selects the node such that the BFS tree from it has the minimum depth but the maximum number of leaf nodes. On the ER random graph, we establish the following possibility and impossibility results[1]: (i) when the infection duration $< \frac{\log n}{(1+\alpha)\log\mu}$ for some $\alpha > 0.5$, SFT identifies the source with probability 1 (w.p.1) asymptotically (as network size increases to infinity), where n is the network size and μ is the average node degree; (ii) when the infection duration $> \lceil\frac{\log n}{\log\mu}\rceil + 2$, the probability of identifying the source approaches zero asymptotically under any algorithm; and (iii) when infection duration $< \frac{\log n}{(1+\alpha)\log\mu}$ for some $\alpha > 0$, asymptotically, at least $1-\delta$ fraction of the nodes on the BFS tree starting from the source are leaf-nodes, where $\delta = 3\sqrt{\frac{\log n}{\mu}}$, i.e., the BFS tree starting from the actual source is a fat tree. Numerical experiments on tree networks, the ER random graphs and real world networks with different evaluation metrics show that the SFT algorithm outperforms existing algorithms.

While Chapter 2 answered some fundamental questions about information source detection in tree and non-tree networks. In practice, the problem is difficult to solve, in particular, in complex networks. In addition, real world diffusion patterns are of great diversity and some of them would be very different from the IC model. Therefore, the next focus of this dissertation is to design an efficient algorithm which is robust to diffusion models. In Chapter 3, we considered the case when timestamp information are available and found that even partial timestamps, which are available in many practical scenarios, provide important insights about the location of

[1] The results hold under some other minor conditions, which will be presented in Section 2.2.

the source while most related literatures ignore the timestamp information. We use a *ranking-on-graphs* approach to exploit the timestamp information where infected nodes are ranked according to their likelihood of being the source. Two ranking algorithms, cost-based ranking (CR) and tree-based ranking (TR), are proposed in Chapter 3. Experimental evaluations with synthetic and real-world data show that our algorithms significantly improve the ranking accuracy compared with existing algorithms. Furthermore, we obtain the following observations from the experiments: (1) locating the source in networks with small diameters and hub nodes is more difficult than in networks that are locally tree-like; and (2) both ranking algorithms perform well under different diffusion models.

1.3 Thesis Outline

The rest of the dissertation is organized as follows. In Chapter 2, we propose the SFT algorithms and present the theoretical performance guarantees of the algorithm for tree and non-tree networks. Two ranking algorithms in the graphs which utilize the partial timestamps are proposed in Chapter 3 along with extensive experiments on both synthetic and real world datasets. We concludes the dissertation and discuss future directions in Chapter 4. All the proofs are presented in the appendices.

Chapter 2

INFORMATION SOURCE DETECTIONS: TREES AND BEYOND

In this chapter, we first develop the SFT algorithm, and then present a comprehensive performance analysis of the algorithm under the IC model for both tree networks and the ER random graph. To the best of our knowledge, SFT is the first algorithm that has provable performance guarantees on both tree networks and the ER random graph (Erdos and Renyi, 1959) (non-tree networks).

The fundamental possibility and impossibility results are summarized as follows.

1. For tree networks, we prove that the Jordan infection center with the maximum weighted boundary node degree (WBND) is the MAP estimator of the source under the heterogeneous IC model. Based on the derivation, we propose the SFT algorithm which is applicable to both tree and general networks.

2. We analyze the performance of the SFT algorithm on the ER random graph. Under some mild conditions on the average node degree, we establish the following three results:

   (i) Assume the infection duration $< \frac{\log n}{(1+\alpha)\log \mu}$ for some $\alpha > 0.5$, SFT identifies the source with probability 1 (w.p.1) asymptotically (as network size increases to infinity).

   (ii) Assume the infection duration $\geq \left\lceil \frac{\log n}{\log \mu} \right\rceil + 2$, the probability of identifying the source approaches zero asymptotically under any information source detection algorithm, i.e., it is impossible to detect the source with non-zero probability.

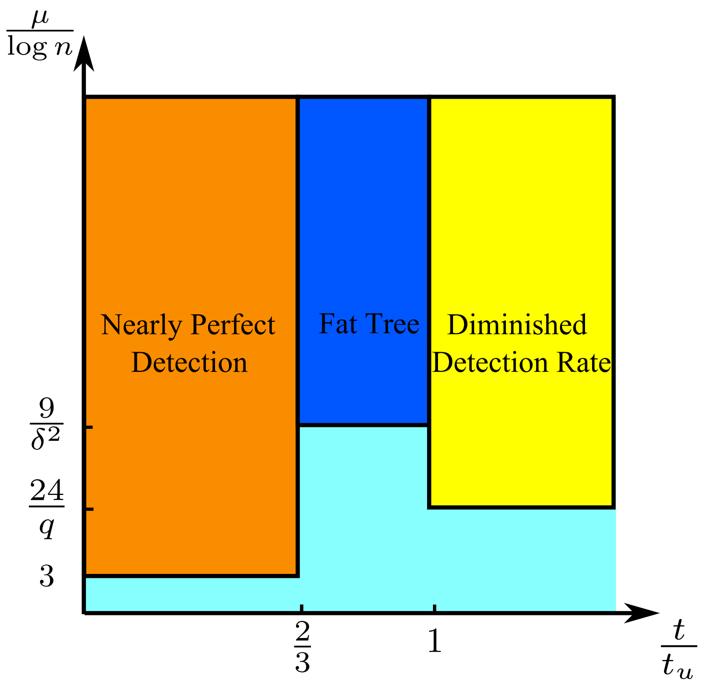


Figure 2.1: Summary of the main results. This figure summarizes the key results in terms of t, the infection time, and μ, the average node degree. In the figure, q is the lower bound on the infection probability; "fat tree" means that there are $1-\delta$ fraction of nodes are boundary nodes on the BFS tree rooted at the source; and $t_u = \left\lceil \frac{\log n}{\log \mu} \right\rceil + 2$, which is the lower bound of the observation time (we proved that all algorithms fail when $t > t_u$.)

(iii) Assume the infection duration $< \frac{\log n}{(1+\alpha)\log \mu}$ for some $\alpha > 0$, asymptotically, at least $1-\delta$ fraction of the nodes on the BFS tree starting from the source are leaf-nodes, where $\delta > 3\sqrt{\frac{\log n}{\mu}}$. This result does not provide any guarantee on the probability of correctly localizing the source, but states that the BFS tree starting from the true source is a "fat" tree, which further justifies the SFT algorithm.

The results are summarized in Figure 2.1. We remark that results (i) and (iii) are highly nontrivial because a subgraph of the ER random graph is a tree with high probability only when the diameter is $\frac{\log n}{2 \log \mu}$, and (i) and (iii) deal with subgraphs that are not trees. To the best of our knowledge, these are the first theoretical results on information source detection on non-tree networks under probabilistic diffusion models.

3. One drawback of the WBND tie-breaking is that it requires the infection probabilities of all edges in the IC model. We simplify WBND to BND by using the boundary node degree in SFT. As shown in Section 2.3, the performance of BND tie-breaking is very close to WBND tie-breaking. We conducted extensive simulations on trees, ER random graphs and real world networks. SFT outperforms existing algorithms by having a higher detection rate and being closer to the actual source. We further evaluated the scalability of the algorithm by measuring the running time. Our results demonstrate that SFT achieves a better performance with a reasonably short execution time.

The rest of the chapter is organized as follows. In Section 2.1.1, we first introduce the IC model and formulate a MAP problem for information source detection and SFT will be presented in Section 2.1.2. Section 2.2 summarizes the main theoretical results of the paper including the analysis on both tree networks and the ER random graph. The simulation based performance evaluation will be presented in Section 2.3. All the proofs are provided in the appendices.

2.1 Model and Algorithm

2.1.1 Model

Given an undirected graph g, denote by $\mathcal{E}(g)$ the set of edges in g and denote by $\mathcal{V}(g)$ the set of nodes in g. We consider the IC model (Goldenberg *et al.*, 2001) for information diffusion and assume a time-slotted system. Each node has two possible states: active (or called infected) and inactive (or called susceptible). At time slot 0, all nodes are inactive except the source. At the beginning of each time slot, an active node attempts to activate its inactive neighbors. If an attempt is successful, the corresponding node becomes active at next time slot; otherwise, the node remains inactive. The weight of each edge represents the success probability of the attempt, called the *infection probability* of the edge and each attempt is independent of others. Each active node only attempts to activate each of its inactive neighbors once. Denote by q_{uv} the infection probability of edge (u, v) and we assume $q_{uv} = q_{vu}$ throughout the chapter since the graph is undirected. We assume that a complete snapshot $\mathcal{O} = \{\mathcal{I}, \mathcal{H}\}$ of the network at time t (called the *observation time*) is given, where $\mathcal{I}$ is the set of active nodes and $\mathcal{H}$ is the set of inactive nodes. Based on $\mathcal{O}$, we want to detect the source. We further assume the observation time t is unknown. The problem can be formulated as a MAP problem as follows,

$$\arg\max_{v \in \mathcal{V}(g)} \Pr(v|\mathcal{O}).$$

where $\Pr(v|\mathcal{O})$ is the probability that v is the source given the snapshot $\mathcal{O}$. The infected nodes form a connected component under the IC model, called the *infection subgraph* and denoted by g_i. Since the source must be an infected node, the MAP problem can be simplified to

$$\arg\max_{v \in \mathcal{I}} \Pr(v|\mathcal{O}),$$

and the search of the information source can be restricted to the infection subgraph. We assume the observation time t, which itself is a random variable, is independent of the source node.

2.1.2 The Short-Fat Tree Algorithm

In this section, we first present the SFT algorithm. We will show in Theorem 2 that the algorithm outputs the MAP estimator for tree networks, which motivates the algorithm. The performance on the ER random graph is studied in Theorems 3 and 4.

We first introduce several necessary definitions. Denote by d^g_{uv} the distance from node u to node v in graph g, where the distance is the minimum number of hops between two nodes. Define the *infection eccentricity* of an infected node to be the maximum distance from the node to all infected nodes on the infection subgraph g_i, denote by $e(v, \mathcal{I})$,

$$e(v, \mathcal{I}) = \max_{u \in \mathcal{I}} d^{g_i}_{uv}.$$

Recall that the *Jordan infection centers* of a graph are the nodes with the minimum infection eccentricity (Zhu and Ying, 2014a).

Consider a BFS tree T_v rooted at node v on the infection subgraph g_i. Denote by $\text{par}_v(u)$ the parent of node u in T_v. Define the set of *boundary nodes* of T_v to be

$$\mathcal{B}(v, \mathcal{I}) = \{w \in \mathcal{I} | d^{T_v}_{vw} = e(v, \mathcal{I})\},$$

which are the set of active nodes furthest away from node v in the infection subgraph.

The weighted boundary node degree (WBND) with respect to node v is defined to be

$$\sum_{(u,w) \in \mathcal{F}'_v} |\log(1 - q_{uw})|, \tag{2.1}$$

where

$$\mathcal{F}'_v = \{(u, w) | (u, w) \in \mathcal{E}(g), w \neq \text{par}_v(u), u \in \mathcal{B}(v, \mathcal{I})\}. \tag{2.2}$$

The SFT algorithm, presented in Algorithm 1, identifies the source based on the BFS trees on the infection subgraph. The algorithm is called the *Short-Fat Tree* algorithm because (1) it first identifies the *shortest* BFS tree; and (2) the shortest BFS tree that maximizes the WBND is then selected in tie-breaking, which is usually the tree with a large number of leaf-nodes, i.e., a *fat* tree. The pseudo codes of the algorithms are presented in Algorithm 1 and 2, which can be executed in a parallel fashion.

A simple example is presented in Figure 2.2 to illustrate algorithm. Each node has a unique node ID. The red nodes are infected and the white nodes are healthy. For simplicity, we assume the weights of all edges equal to $|\log(0.5)|$. The vector next to each infected node records the distance from it to all infected nodes. Initially at Iteration 0, each infected node only knows the distance to itself. For example, $[0 * * *]$ next to node 1 means that the distance from node 1 to itself is 0 and the distance from node 1 to node 2 is unknown. At Iteration 1, each infected node broadcasts its ID to its neighbors in next iteration. Upon receiving the node ID from node 1, node 2 updates its vector to $[1\ 0 * *]$, and broadcasts node 1's ID to its neighbors. The figure in the middle shows the updated vectors after all node ID exchanges occur at Iteration 1. At Iteration 2, node 1 and 2 do not receive any new node IDs. Therefore, node 1 and node 2 report themselves as the Jordan infection centers which are circled with blue in Figure 2.2. The boundary nodes of the BFS tree rooted at node 1 are 2,3,4. The WBND of node 1 is $13|\log(0.5)|$. Similarly, the boundary nodes of the BFS tree rooted at node 2 are 1,3,4 and the WBND is $9|\log(0.5)|$. Therefore, node 1 has a larger WBND and is chosen to be our estimator of the information source.

Remark: Note Equation (2.1) requires the infection probabilities of all edges in the network which could be hard to obtain in practice. When the infection probabilities are not available, we can assume each edge has the same infection probability q

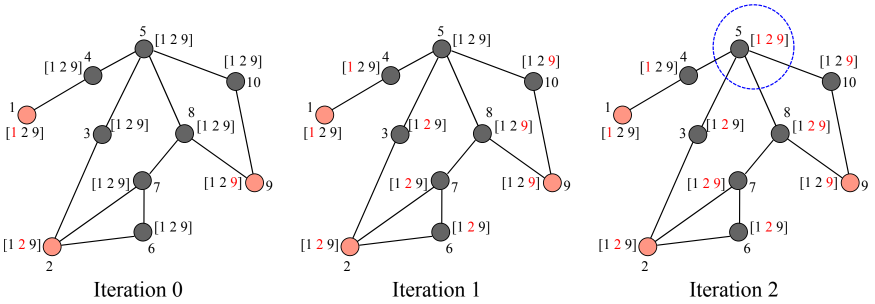


Figure 2.2: An example of the Short-Fat Tree algorithm

```
Algorithm 1: The Short-Fat Tree Algorithm
  Input: $\mathcal{I}, g$;
  Output: $v^\dagger$ (the estimator of information source)
  Set subgraph $g_i$ to be a subgraph of $g$ induced by node set $\mathcal{I}$.
  for $v \in \mathcal{I}$ do
      Initialize an empty dictionary $D_v$ associating with node $v$.
      Set $D_v[v] = 0$.
  end
  Each node receives its own node ID at time slot 0.
  Set time slot $t = 1$.
  do
      for $v \in \mathcal{I}$ do
          if $v$ received new node IDs in $t-1$ time slot, where "new" IDs means
          node $v$ did not receive them before time slot $t-1$ then
              $v$ broadcasts the new node IDs to its neighbors in $g_i$.
          end
      end
      for $v \in \mathcal{I}$ do
          if $v$ receives a new node ID $u$ which is not in $D_v$. then
              Set $D_v[u] = t$.
          end
      end
      $t = t + 1$.
  while No node receives $|\mathcal{I}|$ distinct node IDs;
  Set $\mathcal{S}$ to be the set of nodes who receive $|\mathcal{I}|$ distinct node IDs.
  for $v \in \mathcal{S}$ do
      Compute WBND of $T_v$ using Algorithm 2.
  end
  return $v^\dagger \in \mathcal{S}$ with the maximum WBND.
```

Algorithm 2: The WBND Algorithm

```
Input: $v$,$D_v$ (Dictionary of distance from $v$ to other nodes), $g$, $\mathcal{I}$, $t$;
Output: WBND($v$)
Set $\mathcal{B}$ to be empty.
for $u$ in the keys of $D_v$ do
    if $D_v[u] = t$ then
        Add $u$ to $\mathcal{B}$.
    end
end
Set $x = 0$;
for $w \in \mathcal{B}$ do
    Find the neighbor $u$ of $w$ such that $D_v[u] = t - 1$.
    Set $x = x + \sum_{y \in \text{neighbors}(w)} |\log(1 - q_{wy})| - |\log(1 - q_{wu})|$.
end
return $x$.
```

and WBND becomes,

$$\left(\sum_{u \in \mathcal{B}(v,\mathcal{I})} \deg(u) - |\mathcal{B}(v,\mathcal{I})| \right) |\log(1-q)|,$$

where $\deg(u)$ is the degree of node u.

Define the boundary node degree (BND) of node v to be

$$\sum_{u \in \mathcal{B}(v,\mathcal{I})} \deg(u) - |\mathcal{B}(v,\mathcal{I})| \tag{2.3}$$

which is only related to the degree of the boundary nodes and can be used to replace WBND as the tie-breaking among the Jordan infection center in SFT when the infection probabilities are unknown. As shown in Section 2.3, the performance using BND and WBND are similar. To differentiate the two algorithms, we call the algorithm which uses WBND as wSFT and the one which uses BND as SFT. Next, we analyze the complexity of the algorithm.

Theorem 1. *The worst case computational complexity of the SFT algorithm is* $O(|\mathcal{I}|deg(\mathcal{I}))$ *where* $deg(\mathcal{I})$ *is the total degree of nodes in* $\mathcal{I}$ *in graph* g.

The detailed proof can be found in Appendix A.

2.2 Main Results

In this section, we summarize the main results of the chapter and present the intuitions of the proofs.

2.2.1 Main Result 1 (The MAP Estimator on Tree Networks)

On tree networks, the Jordan infection center of the infection subgraph with the maximum WBND is a MAP estimator.

Theorem 2. *Consider a tree network. Assume the following conditions hold.*

- *The probability distribution of the observation time satisfies* $\Pr(t) \geq \Pr(t+1)$ *for all* t.
- *The source is uniformly and randomly selected, i.e.,* $\Pr(u) = \Pr(v)$.

Denote by $\mathcal{J}$ *the set of Jordan infection centers of the infection subgraph* g_i. *We have*

$$\arg\max_{u \in \mathcal{J}} \sum_{(v,w) \in \mathcal{F}'_u} |\log(1 - q_{vw})| \subset \arg\max_{u} \Pr(u|\mathcal{O}). \tag{2.4}$$

where $\mathcal{F}'_u$ *is defined in Equation (2.2).*

The detailed proof can be found in Appendix A. The theorem has been proved in two steps: 1) We show that one of the Jordan infection centers maximizes the posterior probability on tree networks following similar arguments in (Zhu and Ying, 2014a). In particular, for two neighboring nodes, we show the one with smaller infection eccentricity has a larger posterior probability of being the source. Since there exists a path from any node to a Jordan infection center on the infection subgraph, along which the infection eccentricity strictly decreases, we conclude that a MAP estimator of the source must be a Jordan infection center; 2) Consider the case where the tree network has more than one (at most two according to (Harary, 1991)) Jordan infection centers. When the observation time is larger than the infection eccentricity of the Jordan infection center, the probability of having the observed infected subgraph from any Jordan infection center is the same. When the observation time equals the infection eccentricity, we prove that the probability for a Jordan infection center to be the source is an increasing function of WBND of the BFS tree starting from it.

2.2.2 Main Result 2 (Detection with Probability One on the ER Graph)

Denote by n the number of nodes in the ER random graph and p the wiring probability of the ER random graph. Let $\mu = np$. Recall that t is the observation time. We show that the Jordan infection center is the actual source in the ER random graph with probability one asymptotically when $t < \frac{\log n}{(1+\alpha)\log \mu}$, which implies that SFT can locate the source w.p.1 asymptotically.

Theorem 3. *If the following conditions hold, source s is the only Jordan infection center on the infection subgraph with probability one asymptotically.*

- $\mu > 3 \log n$.
- $t \leq \frac{\log n}{(1+\alpha)\log \mu}$, *for some* $\alpha \in (\frac{1}{2}, 1)$.

We present a brief overview of the proof and the details can be found in Appendix A. Note the infection eccentricity of the actual source is no larger than the observation time t. We show in the proof that the infection eccentricity of an infected node other than the source is larger than t. Consider the BFS tree $T^{\dagger}$ rooted at the actual source s. A node is said to be on level i if its distance to the source is i. Consider another infected node s'. Denote by $a(s')$ the ancestor of s' on level 1 of $T^{\dagger}$. As shown in Figure 2.3, the yellow area shows the level t infected nodes on subtree T_u^{-s}, which is the subtree of $T^{\dagger}$ rooted at node u, and the distance from s' to a node in the yellow area is larger than t if any path between the two nodes can only traverse the edges on tree $T^\dagger$. If s' has an infection eccentricity no larger than t, there must exist a path from s' to each node in the yellow area with length no larger than t. Such a path must contain edges that are not in $T^\dagger$ (we call these edges *collision edges*). We show in the proof that the number of nodes that are within t hops from s' via collision edges are strictly less than the number of nodes in the yellow area. Therefore, the infection eccentricity of s' must be larger than t, which implies that s is the only Jordan infection center.

Existing theoretical results in the literature on information source detection problems are only for tree networks. As shown in the proof of Theorem 3, the infection subgraph of the ER random graph is not a tree when $t > \frac{\log n}{2\log\mu}$. From the best of our knowledge, this result is the first one on non-tree networks.

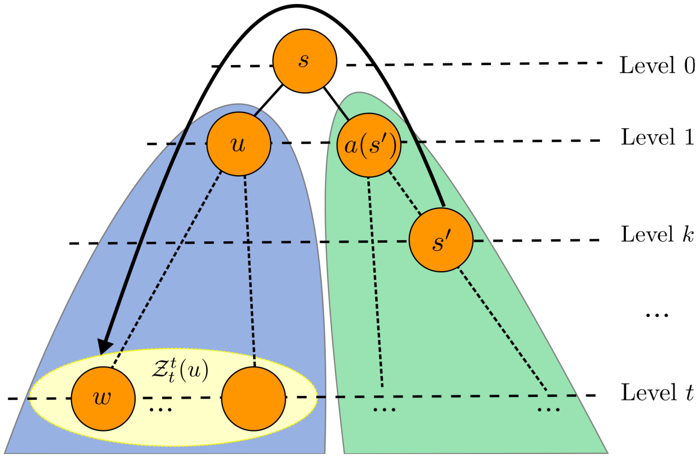


Figure 2.3: A pictorial example of $\mathcal{Z}_t^t(u)$ in BFS tree $T^\dagger$

2.2.3 Main Result 3 (The Fat Tree Result on the ER Graph)

Theorem 4. *If the following conditions hold,*

- $\mu > \frac{9}{\delta^2}\log n$.
- $t \leq \frac{\log n}{(1+\alpha)\log\mu}$, *for some* $\alpha \in (0,1)$.

the leaf-nodes of the BFS tree starting from the actual source consists of at least $1-\delta$ *fraction of the BFS tree asymptotically.*

The detailed proof can be found in Appendix A. Consider the BFS tree from the source s in graph g. The boundary nodes are the nodes at level t and all boundary nodes must be infected at time t. If we ignore the presence of collision edges, the number of infected nodes roughly increases by a factor of $q\mu$ at each level where $q = \min_{(u,v)\in\mathcal{E}(g)} q_{uv}$. Due to this exponential growth nature, the total number of infected nodes is dominated by those infected at the last time slot. We show this property holds with the presence of collision edges. Theorem 4 suggests that the BFS tree rooted at the actual source is a "fat" tree and the BND of the actual source is large. Hence, in the tie breaking, the SFT algorithm has a good chance to select the actual source, which suggests that BND is a good tie breaking rule for the ER random graph.

2.2.4 Main Result 4 (The Impossibility Result on the ER Graph)

We next present the threshold of t after which it is impossible for any algorithm to find the actual source with a non-zero probability asymptotically. The result is based on the analysis of the diameter of an ER random graph in Theorem 4.2 in (Draief and Massouli, 2010). For clarity purpose, we rephrase that theorem with our notations in the following lemma.

Lemma 5. *If* $24\log n < np << \sqrt{n}$, *we have*

$$\lim_{n\to\infty} \Pr(\mathit{Diameter}(g) \le D+2) = 1,$$

where $D = \lceil \frac{\log n}{\log np} \rceil$.

We remark that in (Draief and Massouli, 2010), the condition is $\log n << (n-1)p << \sqrt{n}$. We explicitly calculated the lower bound according to the proof in (Draief and Massouli, 2010). For the sake of completeness, we present the proof in Appendix A.

Based on Lemma 5, we obtain the following impossibility result.

Theorem 6. *If* $24\log n < q\mu << \sqrt{n}$ *and* $q > 0$ *is a constant,*

$$\lim_{n\to\infty} \Pr(\mathcal{I} = \mathcal{V}(g)) = 1$$

when the observation time

$$t \geq \left\lceil \frac{\log n}{\log\mu + \log q} \right\rceil + 2 \triangleq t_u. \tag{2.5}$$

In other words the entire network is infected. In such a case, asymptotically, the probability of any node being the source is $1/n$*.*

The process to generate the ER random graph and the process of the information diffusion under the IC model can be viewed as a combined process. In this combined process, an edge exits only when the edge exists in the ER random graph and is live in the IC model. The detailed definition of the live edge could be found in Appendix A. Loosely speaking, an edge (u, v) is said to be live if node v is infected by node u under the IC model. When the observation time is larger than or equal to the diameter of the coupled ER random graph, all nodes in the network are infected. In such a case, the probability of a node being the source is $1/n$ as the source was uniformly chosen. Based on Lemma 5, the diameter of the combine network is smaller than $\lceil\frac{\log n}{\log q + \log\mu}\rceil + 2$ w.p.1 asymptotically.

Remark 1: We compare t_u in Equation (2.5) and the upper bound in Theorem 3. Since q is a constant, the ratio between t_u and the upper bound becomes $\frac{1}{1+\alpha}$ asymptotically. Since α can be arbitrarily close to $\frac{1}{2}$, the ratio becomes $\frac{2}{3}$. Therefore, the Jordan infection center is the actual source when the observation time is in the range of $(0, \frac{2}{3}t_u)$ and it is impossible to locate the source when the observation time is (t_u, ∞).

Remark 2: We compare t_u and the upper bound in Theorem 4 and asymptotically the ratio between t_u and the upper bound becomes $\frac{1}{1+\alpha}$ where $\alpha \in (0, 1)$. Since α can be arbitrarily close to 0 and the ratio are close to 1 which means the BFS tree from the source has large BND before it becomes impossible to locate the source. While the theorem does not provide any guarantee on the detection rate, it justifies the tie-breaking using BND and WBND.

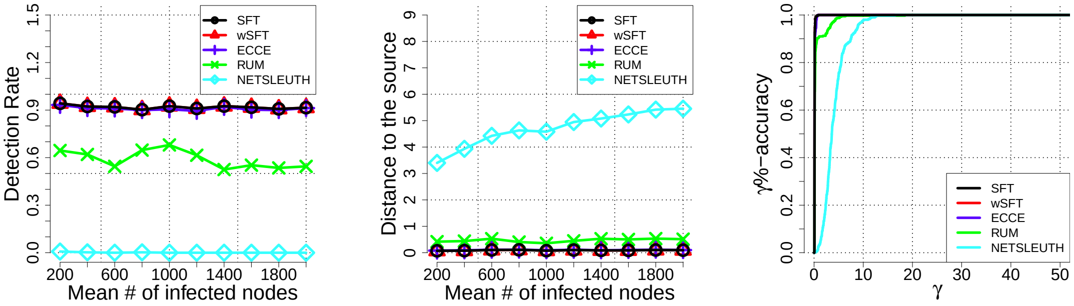
2.3 Performance Evaluation

In this section, we compare the proposed algorithms with existing algorithms on different networks such as tree networks, the ER random graphs and real world networks.

2.3.1 Algorithms

Among all the existing algorithms discussed in Chapter 1, we choose the algorithms which require only a single snapshot of the network but not the infection probabilities which could be difficult to obtain in practice. We compared SFT and wSFT with the algorithms summarized as follows.

- **ECCE:** Select the node with minimum infection eccentricity. Ties are breaking randomly. Recall the definition of the infection eccentricity is the maximum distance from the node to all infected nodes. (Zhu and Ying, 2014a) showed that the optimal sample path estimator on tree networks is the Jordan infection center of the graph under the SIR model.



(a) Detection rate

(b) Distance to the source

(c) γ%-accuracy

Figure 2.4: Performance in the binomial trees

- **RUM:** Select the node with maximum rumor centrality proposed in (Shah and Zaman, 2011). The rumor centrality was proved to be the maximum likelihood estimator on regular trees under the continuous time SI model in which the infection time follows exponential distribution.

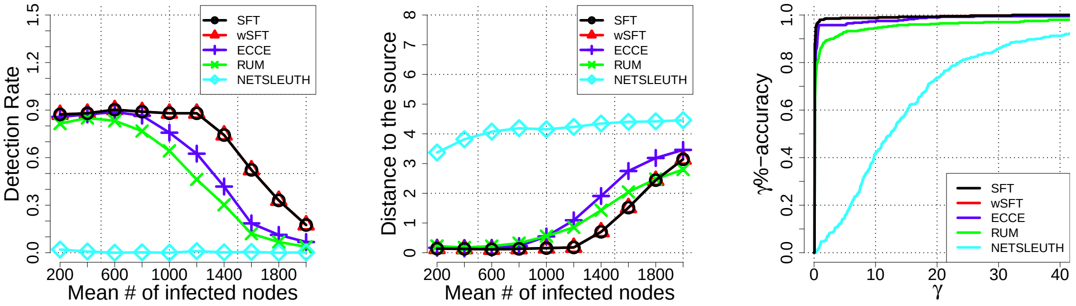
- **NETSLEUTH:** Select the node with maximum value in the eigenvector corresponding to the largest eigenvalue of a submatrix which is constructed from the infected nodes based on the graph Laplacian matrix. The algorithm was proposed in (Prakash *et al.*, 2012).

Among the selected algorithms, only wSFT requires the infection probabilities. We included wSFT to evaluate the importance of the knowledge of edge weights to our algorithm. We will see that the performance of SFT is almost identical to wSFT, so the infection probabilities are not important for our detection algorithm.

2.3.2 Evaluation Metrics

We evaluated the performance of the algorithms with three different metrics.

- Detection rate is the probability that the node identified by the algorithm is



(a) Detection rate

(b) Distance to the source

(c) γ%-accuracy

Figure 2.5: Performance in the ER random graph

the actual source. A desired goal of the information source detection is to have a high detection rate.

- Distance is the number of hops from the source estimator to the actual source. The distance is an often used metric for information source detection.

- γ%-accuracy is the probability with which the source is ranked among top γ percent. Note that besides providing a source estimator, an information source algorithm can also be used to rank the infected nodes according to their likelihood to be the source. For example, SFT can rank the nodes in an ascendant order according to their infection eccentricity and then breaks ties using BND. Other algorithms can be used to rank nodes as well. γ%-accuracy is a less ambitious alternation to the detection rate. When the detection rates of all algorithms are low, it is useful to compare γ%-accuracy as a high γ%-accuracy guarantee that the actual source is among the top ranked nodes with a high probability.

2.3.3 Binomial Trees

In this section, we evaluate the algorithms on binomial trees. Denote by $\mathrm{Bi}(m, \beta)$ the binomial distribution with m number of trials and each trial succeeds with probability β. A binomial tree is a tree where the number of children of each node follows a binomial distribution $\mathrm{Bi}(m, \beta)$. In the experiments, we set $m = 20$ and $\beta = 0.5$. We adopted the IC model where the infection probability of each edge is assigned with a uniform distribution in $(0.2, 0.5)$. The lower bound on the infection probability is set to be 0.2 to prevent the diffusion process dies out quickly. We evaluated the performance for different infection size x. Under a discrete infection model, it is hard to obtain the diffusion snapshots with exact x infected nodes. Therefore, for each infection size x, we generate the diffusion samples where the number of infected nodes are in range $[0.75x, 1.25x]$. The source was chosen uniformly at random among all nodes in the network. We varied x from 200 to 2000 with a step size 200. For each infection size, we generate 400 diffusion samples.

Figure 2.4a shows the detection rates for different infection sizes. The detection rates of ECCE, SFT and wSFT do not change for different infection sizes since the structure of the binomial tree is simple. SFT, wSFT and ECCE have the highest detection rate (more than 0.9) while the detection rate of RUM and NETSLEUTH are much lower. The distance results are shown in Figure 2.4b. As expected, SFT, wSFT and ECCE outperform RUM, which are all much better than NETSLEUTH. Figure 2.4c shows the $\gamma\%$-accuracy versus the rank percentage γ. We picked infection size 1,000. As shown in Figure 2.4c, all three algorithms based on infection eccentricity (ECCE, SFT, wSFT) have better performance than RUM and NETSLEUTH. Recall that the node identified by wSFT is a MAP estimator of the actual source.

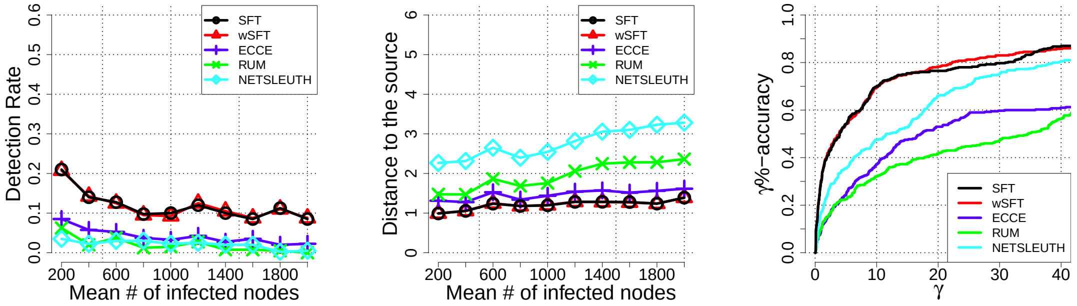
2.3.4 The ER Random Graph

In this section, we compared the performance of the algorithms on the ER random graph. In the experiments, we generated the ER random graph with $n = 5,000$ and wiring probability $p = 0.002$. We again varied the infection network size from 200 to $2,000$. The infection probability of each edge is assigned with a uniform distribution in $(0.2, 0.5)$. We generated 400 diffusion samples.

Figure 2.5a shows the detection rate versus the infection size. The detection rate decreases as the infection size increases. SFT and wSFT have higher detection rates compared to other algorithms. Figure 2.5b shows the results on distance. As we expected, SFT and wSFT outperform other algorithms when the infection size is less than 1,600 nodes. As the size of the infected nodes increase, SFT and wSFT become close to RUM in term of distance to the source. However, the detection rate of both algorithms are still much higher than that of RUM. Another observation is that SFT and wSFT have identical performance which indicates that the performance of SFT is robust to edge weights.

Figure 2.5c shows the γ%-accuracy versus the rank percentage γ with 1000 infected nodes. SFT and wSFT have similar or better performance compared to all other algorithms.

Although the performance of ECCE and SFT algorithms are similar in tree networks, SFT outperforms ECCE significantly on the ER random graphs. The observation indicates that BND is an effective tie breaking rule and increases the detection accuracy.



(a) Detection rate (b) Distance to the source (c) γ%-accuracy

Figure 2.6: Performance in the IAS graph

2.3.5 The Internet Autonomous System Network

The Internet autonomous systems (IAS) network[1] is the Internet autonomous system from Oregon route-views on March, 31st, 2001 with 10,670 nodes and 22,002 edges. The IAS network is a small world network. We adopted similar settings as in Section 2.3.4.

The detection rates are shown in Figure 2.6a. The detection rate of ECCE is low since the IAS graph is a small world network and there are multiple Jordan infection centers due to the small diameter of the network. With the tie breaking rule BND, the detection rate doubles in most cases which demonstrates the effectiveness of BND. While the detection rate of SFT is only 10% when the infection size is 1,000, the distance to the actual source is slightly more than one-hop away as shown in Figure 2.6b. In addition, the γ%-accuracy versus γ for 1,000 infection size is shown in Figure 2.5c. The 10%-accuracies of SFT and wSFT are close to 70% which are significantly higher than that of other algorithms.

[1] Available at `http://snap.stanford.edu/data/index.html`

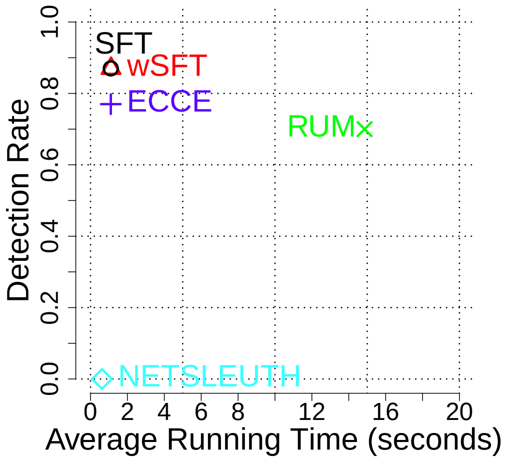


Figure 2.7: Detection rate versus running time in the ER random graph

2.3.6 Running Time vs Performance

In this section, we evaluated the scalability of the algorithms by comparing the running time. The experiments were conducted on an Intel Core i5-3210M CPU with four cores and 8G RAM with a Windows 7 Professional 64 bit system. All algorithms were implemented with python 2.7. The ER random graphs with 5,000 nodes and $p = 0.002$ edge generation probability were used in the experiments. The infection probability of each edge is uniformly distributed over $(0.2, 0.5)$. We generated 100 diffusion samples for the experiments. Figure 2.7 show the average running time versus the detection rate. The infection size is chosen to be 1,000. SFT and wSFT took 1.11 seconds and achieves 0.87 detection rate while NETSLEUTH took 0.62 seconds with 0 detection rate and RUM took 14.86 seconds with 0.7 detection rate. The detection rate of SFT is much higher than NETSLEUTH and SFT is 14 times faster than RUM.

Chapter 3

INFORMATION SOURCE DETECTION WITH PARTIAL TIMESTAMPS

The focus of this chapter is to develop source localization algorithms that utilize partial timestamp information.

We remark that while the information source detection problem (or called rumor source detection problem) has been studied recently under a number of different models, most of them ignore timestamp information. As we will see from the experimental evaluations, even limited timestamp information can significantly improve the accuracy of information source detection. In this chapter, we assume that there is only one information source in the network. We use a *ranking-on-graphs* approach to exploit the timestamp information, and develop source localization algorithms that perform well on different networks and under different diffusion models. The main contributions of this chapter are summarized below.

(1) We formulate the source detection problem as a ranking problem on graphs, where infected nodes are ranked according to their likelihood of being the source. Define a *spreading tree* to include (i) a directed tree with all infected nodes and (ii) the complete timestamps of contagion propagation (the detailed definition will be

	CR	TR	GAU	NETSLEUTH	ECCE	RUM
IAS	0.76	0.68	0.57	0.43	0.15	0.15
PG	0.98	0.99	0.98	0.43	0.43	0.39

Table 3.1: The 10%-accuracy under different source localization algorithms with 50% timestamps

presented in Section 3.1). Given a spreading tree rooted at Node v, denoted by $\mathcal{P}_v$, we define a quadratic cost $C(\mathcal{P}_v)$ depending on the structure of the tree and the timestamps. The cost of Node v is then defined to be

$$C(v) = \min_{\mathcal{P}_v} C(\mathcal{P}_v), \tag{3.1}$$

i.e., the minimum cost among all spreading trees rooted at Node v. Based on the costs and spreading trees, we propose two ranking methods:

(i) rank the infected nodes in an ascendent order according to $C(v)$, called *cost-based ranking (CR)*, and

(ii) find the minimum cost spreading tree, i.e.,

$$\mathcal{P}^* = \arg\min_{\mathcal{P}} C(\mathcal{P}),$$

and rank the infected nodes according to their timestamps on the minimum cost spreading tree, called *tree-based ranking (TR).*

(2) The computational complexity of $C(v)$ is very high due to the large number of possible spreading trees. We prove that problem (3.1) is NP-hard by connecting it to the longest-path problem (Garey and Johnson, 1979).

(3) We propose a greedy algorithm, named Earliest Infection First (EIF), to construct a spreading tree to approximate the minimum cost spreading tree for a given root Node v, denoted by $\tilde{\mathcal{P}}_v$. The greedy algorithm is designed based on the minimum cost solution for line networks. EIF first sorts the infected nodes with observed timestamps in an ascendent order of the timestamps, and then iteratively attaches these nodes using a modified breadth-first search algorithm. In CR, the infected nodes are then ranked based on $C(\tilde{\mathcal{P}}_v)$; and in TR, the nodes are ranked based on the complete timestamps of the spreading tree $\tilde{\mathcal{P}}^*$ such that

$$\tilde{\mathcal{P}}^* = \arg\min C(\tilde{\mathcal{P}}_v).$$

We remark that for infected nodes with unknown infection time, EIF assigns the infection timestamps during the construction of the spreading tree $\tilde{\mathcal{P}}_v$. The details can be found in Section 3.2.

(4) We conducted extensive experimental evaluations using both synthetic data and real-world social network data (Sina Weibo[1]). The performance metric is the probability with which the source is ranked among top γ percent, named γ%-accuracy. We have the following observations from the experimental results:

   (i) Both CR and TR significantly outperform existing source detection algorithms in both synthetic data and real-world data. Table 3.1 summarizes the 10%-accuracy in the Internet autonomous systems (IAS) network and the power grid (PG) network. The readers could refer to Section 3.4.2 for the abbreviations of other baseline algorithms.

   (ii) Our results show that both TR and CR perform well under different diffusion models and different distributions of timestamps.

   (iii) Early timestamps are more valuable for locating the source than recent ones.

   (iv) Network topology has a significant impact on the performance of source localization algorithms, including both ours and existing ones. For example, the γ%-accuracy in the IAS network is lower than that in the PG network (see Table 3.1 for the comparison). This suggests that the problem is more difficult in networks with small diameters and hubs than in networks that are locally tree-like.

   (v) The performance in terms of normalized rank is also evaluated in Section 3.6.3.

[1] `http://www.weibo.com/`

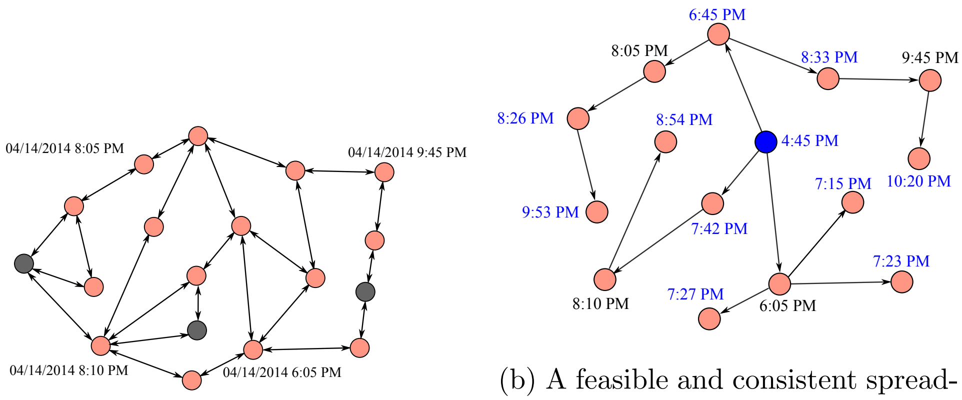
3.1 A Ranking Approach for Source Localization

Ideally, the output of a source localization algorithm should be a single node, which matches the source with a high probability. However, with limited timestamp information, this goal is too ambitious, if not impossible, to achieve. From the best of our knowledge, almost all evaluations using real-world networks show that the detection rates of existing source localization algorithms are very low (Shah and Zaman, 2011; Zhu and Ying, 2014a; Chen *et al.*, 2014; Luo *et al.*, 2013), where the detection rate is the probability that the detected node is the source.

When the detection rate is low, instead of providing a single source estimator, a better and more useful output of a source localization algorithm would be a node ranking, where nodes are ordered according to their likelihood of being the source. With such a ranking, further investigation can be conducted to locate the source. The more accurate the ranking, the less amount of resources is needed in the further investigation. Furthermore, the authority may only have the resources to search a small portion of the entire network. Therefore, we also want the ranking is more accurate at the top, called the accuracy at the top in (Boyd *et al.*, 2012). In this chapter, we will evaluate the $\gamma\%$ accuracy, which is the probability that the source is ranked among the top γ percent and the normalized rank.

In this chapter, we assume the input of a source localization algorithm includes the following information:

- *A network* $g = (\mathcal{V}(g), \mathcal{E}(g))$*:* The network is an unweighted and directed graph. A Node v in the network represents a physical entity (such as a user of an online social network, a human being, or a mobile device). A directed edge $\omega(v, u)$ from Node v to Node u indicates that the contagion can be transmitted from Node v to Node u.



(a) Available partial timestamps

(b) A feasible and consistent spreading tree

Figure 3.1: An example illustrating available information and a spreading tree

- *A set of infected nodes $\mathcal{I}$:* An infected node is a node that involves in the diffusion process, e.g., a twitter user who retweeted a specific tweet, a computer infected by malware, etc. We assume $\mathcal{I}$ includes all infected nodes in the contagion. So $\mathcal{I}$ forms a connected subgraph of g. In the case $\mathcal{I}$ includes only a subset of infected nodes, our source localization algorithms rank the observed infected nodes according to their likelihood of being the earliest infected node. More discussion can be found in Section 3.5.

- *Partial timestamps $\boldsymbol{\tau}$:* $\boldsymbol{\tau}$ is a $|\mathcal{V}(g)|$-dimensional vector such that $\tau_v = \star$ if the timestamp is missing and otherwise, τ_v is the time at which Node v was infected. We remark that the time here is *the normal clock time, not the relative time with respect to the infection time of the source.* Note that in most cases, the infection time of the source is as difficult to know as the location of the source. In addition, we assume the observed timestamps are exact without any error or noise.

Figure 3.1a is a simple example showing the available information. The nodes in orange are the infected nodes. The time next to a node is the associated timestamp.

We define a spreading tree $\mathcal{P} = (T, \mathbf{t})$ to be a directed tree T with a $|T|$-dimensional vector $\mathbf{t}$. The directed tree T specifies the sequence of infection and the vector $\mathbf{t}$ specifies the time at which each infection occurs. We further require the time sequence $\mathbf{t}$ of a spreading tree to be *feasible* such that the infection time of a node is larger than its parent's, and to be *consistent* with the partial timestamps $\boldsymbol{\tau}$ such that $t_v = \tau_v$ if $\tau_v \neq \star$. Figure 3.1b shows a spreading tree that is feasible and consistent with the observation shown in Figure 3.1a. Note that, for simplicity, we omitted the date in the figure by assuming all events occur on the same day. The timestamps in black are the observed timestamps and the ones in blue are assigned by us. Denote by $\mathcal{F}(\mathcal{I}, \boldsymbol{\tau})$ the set of spreading trees that are both feasible and consistent with the partial timestamps.

3.1.1 Quadratic Cost and Sample Path Approach

Given a spreading tree $\mathcal{P} = (T, \mathbf{t}) \in \mathcal{F}(\mathcal{I}, \boldsymbol{\tau})$, we define the cost of the tree to be

$$C(\mathcal{P}) = \sum_{(v,w)\in T} (t_w - t_v - \eta)^2, \tag{3.2}$$

for some constant $\eta > 0$. This quadratic cost function is motivated by the following model.

The model is a continuous time SI model. Each node has two possible states: susceptible and infected. The infection propagates via edges. For each edge $(v, w) \in T$, assume that the time it takes for Node v to infect Node w follows a truncated Gaussian distribution with mean η and variance σ^2. Then given a spreading tree $\mathcal{P}$, the probability density associated with time sequence $\mathbf{t}$ is

$$f_{\mathcal{P}}(\mathbf{t}) = \prod_{(v,w)\in T} \frac{1}{Z\sqrt{2\pi}\sigma} \exp\left(-\frac{(t_w - t_v - \eta)^2}{2\sigma^2}\right), \tag{3.3}$$

where Z is the normalization constant. Note each node can be only infected by its parent when the spreading tree is given. Therefore, the log-likelihood is

$$\begin{aligned}&\log f_{\mathcal{P}}(\mathbf{t})\\=&-|\mathcal{E}(T)|\log(Z\sqrt{2\pi}\sigma)-\frac{1}{2\sigma^2}\sum_{(v,w)\in T}(t_w-t_v-\eta)^2,\end{aligned}$$

where $|\mathcal{E}(T)|$ is the number of edges in the tree. Therefore, given a tree T, the log-likelihood of time sequence $\mathbf{t}$ is inversely proportional to the quadratic cost defined in (3.2). The lower the cost, the more likely the time sequence occurs. While the quadratic cost is justified by the truncated Gaussian SI model, we remark that the algorithms based on the quadratic cost can be used on any diffusion model. We will evaluate the performance of the proposed algorithms under different diffusion models and networks in Section 3.4.

Now given an infected node in the network, the cost of the node is defined to be minimum cost among all spreading trees rooted at the node. Using $\mathcal{P}_v$ to denote a spreading tree rooted at Node v, the cost of Node v is

$$C(v)=\min_{\mathcal{P}_v\in\mathcal{F}(\mathcal{I},\boldsymbol{\tau})}C(\mathcal{P}_v). \tag{3.4}$$

After obtaining $C(v)$ for each infected node v, the infected nodes can be ranked according to either $C(v)$ or the timestamps of the minimum cost spreading tree. However, the calculation of $C(v)$ in a general graph is NP-hard as shown in the following theorem.

Theorem 7. *Problem (3.4) is an NP-hard problem.* □

Remark 1: This theorem is proved by showing that the longest-path problem can be solved by solving (3.4). The detailed analysis is presented in the appendix. Since computing the exact value of $C(v)$ is difficult, we present a greedy algorithm in the next section.

3.2 EIF: A Greedy Algorithm

In this section, we present a greedy algorithm, named Earliest-Infection-First (EIF), to solve problem (3.4). Note that if a node's observed infection time is larger than some other node's observed infection time, then it cannot be the source. So we only need to compute cost $C(v)$ for Node v such that $\tau_v = \star$ or $\tau_v = \min_{u:\tau_u \neq \star} \tau_u$. Furthermore, when all infected nodes are known, we can restrict the network to the subnetwork formed by the infected nodes to run the algorithm. We next present the algorithm, together with a simple example in Figure 3.2 for illustration. In the example, all edges are *bidirectional,* so the arrows are omitted, and the network in Figure 3.2 is the subnetwork formed by all infected nodes.

Earliest-Infection-First (EIF)

1. Step 1: The algorithm first estimates η from $\boldsymbol{\tau}$ using the average per-hop infection time. Let d_{vw} denote the length of the shortest path from Node v to Node w, then

$$\eta = \frac{\sum_{\tau_v \neq \star, \tau_w \neq \star, v \neq w} |\tau_v - \tau_w|}{\sum_{\tau_v \neq \star, \tau_w \neq \star, v \neq w} d_{vw}}.$$

   Example: Given the timestamps shown in Figure 3.2, $\eta = 36.94$ minutes.

2. Step 2: Sort the infected nodes in an ascending order according to the observed infection time $\boldsymbol{\tau}$. Let ι denote the ordered list such that ι_1 is the node with the earliest infection time.

   Example: Consider the example in Figure 3.2. The ordered list is

$$\iota = (6, 12, 13, 1).$$

3. Step 3: Construct the initial spreading tree T_0 that includes the root node only and set the cost to be zero.

Example: Assuming we want to compute the cost of Node 10 in Figure 3.2, we first have $T_0 = \{10\}$ and $C(10) = 0$.

4. Step 4: At the k^{th} iteration, Node ι_k is added to the spreading tree T_{k-1} using the following steps.

   Example: At the 3$^{\text{rd}}$ iteration, the current spreading tree is

   $$10 \to 6 \to 7 \to 8 \to 12,$$

   and the associated timestamps are given in Table 3.2. Note that these timestamps are assigned by EIF except those observed ones. The details can be found in the next step. In the 3$^{\text{rd}}$ iteration, Node 13 needs to be added to the spreading tree.

node ID	10	6	7	8	12
Timestamp	5:28	6:05	6:45	7:25	8:05

Table 3.2: The timestamps on the spreading tree in the 3$^{\text{rd}}$ iteration

   (a) For each node m on the spreading tree T_{k-1}, identify a modified shortest path from Node m to Node ι_k. The modified shortest path is a path that has the minimum number of hops among all paths from Node m to Node ι_k, which satisfy the following two conditions:

   – it does not include any nodes on the spreading tree T_{k-1}, except node m;

   – it does not include any nodes on list ι, except node ι_k.

   Example: The modified shortest path from Node 7 to Node 13 is

   $$7 \to 9 \to 13.$$

   There is no modified shortest path from Node 12 to Node 13 since all paths from 12 to 13 go through Node 8 that is on the spreading tree T_2.

(b) For the modified shortest path from Node m to Node ι_k, the cost of the path is defined to be

$$\gamma_m = \tilde{d}_{\iota_k m} \left(\frac{t_{\iota_k} - t_m}{\tilde{d}_{\iota_k m}} - \eta \right)^2 ,$$

where $\tilde{d}_{\iota_k m}$ denotes the length of the modified shortest path from m to ι_k. From all nodes on the spreading tree T_{k-1}, select Node m^* with the minimum cost i.e.,

$$m^* = \arg\min_m \gamma_m.$$

Example: The costs of the modified shortest paths to the nodes on the spreading tree

$$10 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow 12$$

are shown in Table 3.3. Node 7 has the smallest cost.

node ID	10	6	7	8	12
cost	15,640.00	∞	61.83	147.03	∞

Table 3.3: The costs of the modified shortest paths

(c) Construct a new spreading tree T_k by adding the modified shortest path from m^* to ι_k. Assume Node v on the newly added path is h_v hops from Node m^*, the infection time of Node v is set to be

$$t_v = t_{m^*} + (h_v - 1)\frac{t_{\iota_k} - t_{m^*}}{\tilde{d}_{m^* \iota_k}}. \tag{3.5}$$

The cost is updated to $C(v) = C(v) + \gamma_{m^*}$.

Example: At the 3rd iteration, the timestamp of Node 9 is set to be 7:28 PM, and the cost is updated to $C(10) = 89.92$.

5. Step 5: For those infected nodes that have not been added to the spreading tree, add these nodes by using a breadth-first search starting from the spreading tree T. When a new node (say Node w) is added to the spreading tree during the breadth-first search, the infection time of the node is set to be $t_{\text{par}_w} + \eta$, where par_w is the parent of Node w on the spreading tree. Note that the cost $C(v)$ does not change during this step because $t_w - t_{\text{par}_w} - \eta = 0$.

   Example: The final spreading tree and the associated timestamps are presented in Figure 3.2.

Remark 2: The timestamps of nodes on a newly added path are assigned according to Equation (3.5). This is because such an assignment is the minimum cost assignment in a line network in which only the timestamps of two end nodes are known.

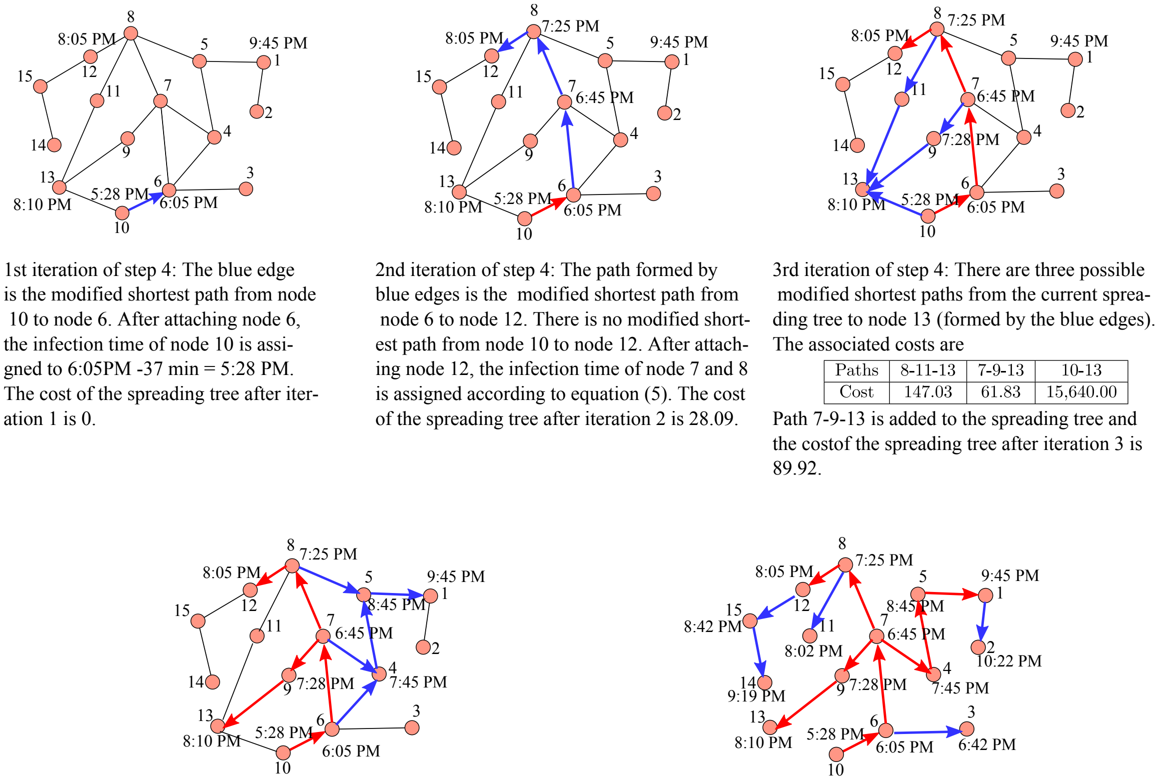
Lemma 8. *Consider a line network with n infected nodes. Assume the infection times of Node 1 and Node n are known and the infection times of the rest nodes are not. Furthermore, assume $\tau_1 < \tau_n$. The quadratic cost defined in (3.4) is minimized by setting*

$$t_k = \tau_1 + (k-1)\frac{\tau_n - \tau_1}{n-1} \tag{3.6}$$

for $1 < k < n$. □

Note that under the assignment above, the infection time, $\tau_{k+1} - \tau_k$, is the same for all edges, which is due to the quadratic form of the cost function. The detailed proof can be found in the appendix.

Remark 3: Note that in Step 4(a), we use the modified shortest path instead of the conventional shortest path. The purpose is to avoid inconsistence when assigning timestamps. For example, consider the 3$^{\text{rd}}$ iteration in Figure 3.2, and the paths



1st iteration of step 4: The blue edge is the modified shortest path from node 10 to node 6. After attaching node 6, the infection time of node 10 is assigned to 6:05PM -37 min = 5:28 PM. The cost of the spreading tree after iteration 1 is 0.

2nd iteration of step 4: The path formed by blue edges is the modified shortest path from node 6 to node 12. There is no modified shortest path from node 10 to node 12. After attaching node 12, the infection time of node 7 and 8 is assigned according to equation (5). The cost of the spreading tree after iteration 2 is 28.09.

3rd iteration of step 4: There are three possible modified shortest paths from the current spreading tree to node 13 (formed by the blue edges). The associated costs are

Paths	8-11-13	7-9-13	10-13
Cost	147.03	61.83	15,640.00

Path 7-9-13 is added to the spreading tree and the costof the spreading tree after iteration 3 is 89.92.

4th iteration of step 4: The costs of the three modified shortest paths to node 1 are listed below.

Paths	8-5-1	7-4-5-1	6-4-5-1
Cost	2,185.93	1,595.29	3,973.42

Path 7-4-5-1 is added to the spreading tree and the cost of the spreading tree after iteration 4 is 1,685.21.

Step 5: Expanding the spreading tree using the breadth-first search to include all infected nodes. The blue edges are newly added edges. The cost of this spreading tree (i.e., the cost of node 10) is 1,685.21.

Figure 3.2: An example for illustrating Step 4 and Step 5 of EIF. The paths formed by blue edges are modified shortest paths. The trees formed by red edges are the spreading trees at the beginning of each iteration.

from Node 7 to Node 2. There are two conventional shortest paths: 7 → 4 → 5 → 1 and 7 → 8 → 5 → 1. If we select path 7 → 8 → 5 → 1 and assign the timestamps according to (3.5), then the infection time of Node 8 is larger that of Node 7, which contradicts the current timestamps of Node 7 and Node 8. Therefore, 7 → 8 → 5 → 1 should not be selected.

Remark 4: A key step of EIF is the construction of the modified shortest paths from the nodes on T_{k-1} to Node ι_k. This can be done by constructing a modified breadth-first search tree starting from Node ι_k. In constructing the modified breadth-first search tree, we first reverse the direction of all edges as we want to construct paths from the nodes on T_{k-1} to Node ι_k. Then starting from Node ι_k, nodes are added in a breadth-first fashion. However, a branch of the tree terminates when the tree meets a node on T_{k-1} or Node ι_l for $l > k$. After obtainng the modified breadth-first search tree, if a leaf node is a node on T_{k-1}, say Node m, then the *reversed path* from Node ι_k to Node m on the modified breadth-first search tree is a modified shortest path from Node m to Node ι_k. If none of the leaf nodes is on T_{k-1}, then the cost of adding ι_k is claimed to be infinity. In Figure 3.2, the trees formed by the blue edges are the modified breadth-first trees at each iteration.

The pseudo code of the EIF algorithm is presented in Algorithm 3.

3.3 Cost-Based and Tree-Based Ranking

Denote by $\tilde{T}_v$ the spreading tree constructed under EIF for Node v, and $\tilde{C}(\tilde{T}_v)$ the corresponding cost computed by EIF. After constructing the spreading tree for each infected node and obtaining the corresponding cost, the nodes are ranked using the following two approaches.

Cost-Based Ranking (CR): Rank the infected nodes in an ascendent order according to $\tilde{C}(\tilde{T}_v)$.

Tree-Based Ranking (TR): Denote by $v^* = \arg\min_v \tilde{C}(\tilde{T}_v)$. Rank the infected nodes in an ascendent order according to the timestamps on $\tilde{T}_{v^*}$.

Theorem 9. *The complexity of CR and TR is $O(|\iota||\mathcal{I}||\mathcal{E}(g_i)|)$, where $|\iota|$ is the number of infected nodes with observed timestamps, $|\mathcal{I}|$ is the number of infected nodes, and $|\mathcal{E}(g_i)|$ is the number of edges in the subgraph formed by the infected nodes.* □

Algorithm 3: Earliest-Infection-First Algorithm

```
Input: $\boldsymbol{\tau}, g_i, v^\dagger$;
Output: $\tilde{C}(\tilde{T}_{v^\dagger})$ (Cost of $v^\dagger$), $\tilde{T}_{v^\dagger}$ (Spreading tree associated with $v^\dagger$);
Set
    $\eta = \frac{\sum_{\tau_v \neq \star, \tau_w \neq \star, v \neq w} |\tau_v - \tau_w|}{\sum_{\tau_v \neq \star, \tau_w \neq \star, v \neq w} d_{vw}}.$
Sort $\boldsymbol{\tau}$ in an ascending order. Denote by $\iota_i$ the $i$th node according to the order.
Set $T_0$ to be a tree that includes only $v^\dagger$ and set $\tilde{C} = 0$.
Set $N$ to be the length of $\boldsymbol{\tau}$.
for $k$ = 1 to $N$ do
    for Node $m$ in Tree $T_{k-1}$ do
        Identify the modified shortest path $\mathcal{P}_{m\iota_k}$ from $m$ to $\iota_k$.
        Compute
            $\gamma_m = |\mathcal{P}_{m\iota_k}| \left( \frac{t_{\iota_k} - t_m}{|\mathcal{P}_{m\iota_k}|} - \eta \right)^2.$
    Select $m^* \in \arg\min_m \gamma_m$.
    Set the infection time of Node $v \in \mathcal{P}_{m^*\iota_k}$ to be
        $t_v = t_{m^*} + (h_v - 1) \frac{t_{\iota_k} - t_{m^*}}{\tilde{d}_{m^*\iota_k}}$
    where $h_v$ is the number of hops from $m^*$ to $v$ on $\mathcal{P}_{m^*\iota_k}$.
    Add $\mathcal{P}_{m^*\iota_k}$ to $T_{k-1}$ to obtain $T_k$.
    Set $\tilde{C} = \tilde{C} + \gamma_{m^*}$.
Let $Q$ be an empty queue and enqueue all nodes on $T_N$.
while $Q$ is not empty do
    Dequeue $Q$. Let $m$ be the dequeued node.
    for All edges from $m$ to $v$ in $G_I$ do
        if $v$ is not in $T_N$ then
            Add edge $(m, v)$ to $T_N$.
            Set $t_v = t_m + \eta$.
            Enqueue $v$ to $Q$.
Set $\tilde{C}(\tilde{T}_{v^\dagger}) = \tilde{C}, \tilde{T}_{v^\dagger} = T_N$
return $\tilde{C}(\tilde{T}_{v^\dagger})$ and $\tilde{T}_{v^\dagger}$.
```

The proof is presented in the appendix.

CR and TR algorithms can be implemented in a distributed fashion where $\tilde{C}(\tilde{T}_v)$ could be computed parallelly for each node v.

3.4 Experimental Evaluation

In this section, we evaluate the performance of TR and CR using both synthetic data and real-world data. While both ranking algorithms (TR and CR) were justified by the sample path based approach based on the truncated Gaussian distribution, one important contribution of the two algorithms is that they are parameter-free and model-free and can be used for any diffusion model and network. In fact, the objective of our design is the development of such a general algorithm. Of course, the theoretical analysis can only be done for a specific model, but we conducted extensive simulations for different diffusion models including the IC model and SpikeM model and further under real social network data sets.

3.4.1 Performance of EIF on a Small Network

In the first set of simulations, we evaluated the performance of EIF of solving the minimum cost of the feasible and consistent spreading trees. Given an observation $\mathcal{I}$ and $\boldsymbol{\tau}$, denote by C^* the minimum cost of the feasible and consistent spreading trees. Then

$$C^* = \min_{\mathcal{P}\in\mathcal{F}(\mathcal{I},\boldsymbol{\tau})} C(\mathcal{P})$$

Denote by $\tilde{C}^*$ the minimum cost of the spreading trees obtained under EIF. We evaluated the approximation ratio $r = \frac{\tilde{C}^*}{C^*}$ on a small network — the Florentine families network (Breiger and Pattison, 1986) which has 15 nodes and 20 edges. Recall that the minimum cost problem is NP-hard, so the approximation ratio is evaluated over a small network only. To compute the actual minimum cost, we first enumerated all possible spanning trees using the algorithm in (Char, 1968), and then computed the minimum cost of each spanning tree by solving the quadratic programming problem.

In this experiment, we assumed the infection time of each edge follows a truncated Gaussian distribution with $\eta = 100$ and $\sigma = 100$. We evaluated the approximation ratio when the number of observed timestamps varied from 5 to 14. The results are shown in Figure 3.3, where each data point is an average of 500 runs. The error bar shows the mean ± standard deviations. Since the ratio can not be smaller than 1.0, the error bar is cut off at 1.0. The approximation ratio is 2.24 with 5 timestamps, 1.5 with 8 timestamps and becomes 1.08 when 14 timestamps are given. This experiment shows that EIF approximates the minimum cost solution reasonably well.

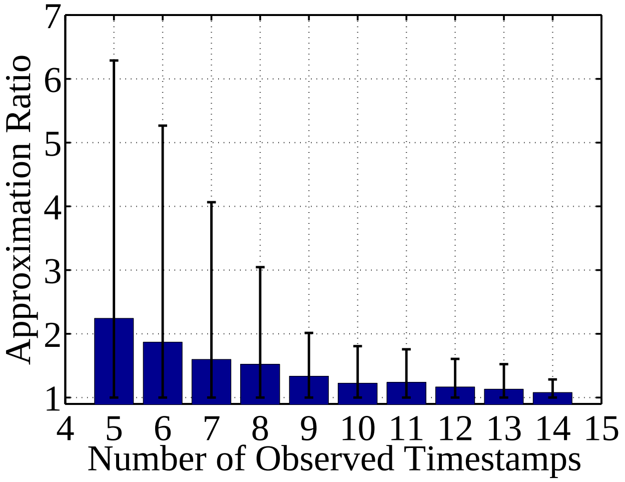


Figure 3.3: The approximation ratio of TR (error bar shows mean ± standard deviation)

3.4.2 Comparison with Other Algorithms

We first tested the algorithms using synthetic data on two real-world networks: the IAS network and the power grid network (PG)[2]:

[2] Available at `http://www-personal.umich.edu/~mejn/netdata/`

- The IAS network is a network of the Internet autonomous systems inferred from Oregon route-views on March, 31st, 2001. The network contains 10,670 nodes and 22,002 edges in the network. IAS is a small world network.

- The PG network is a network of Western States Power Grid of United States. The network contains 4,941 nodes and 6,594 edges. Compared to the IAS network, the PG network is locally tree-like.

We first compare CR and TR with the following four existing source localization algorithms.

- Rumor centrality (RUM): Rumor centrality was proposed in (Shah and Zaman, 2011), and is the maximum likelihood estimator on trees under the SI model. RUM ranks the infected nodes in an ascendent order according to nodes' rumor centrality.

- Infection eccentricity (ECCE): The infection eccentricity of a node is the maximum distance from the node to any infected node in the graph, where the distance is defined to be the length of the shortest path. The node with the smallest infection eccentricity, named Jordan infection center, is the optimal sample-path-based estimator on tree networks under the SIR model (Zhu and Ying, 2014a). ECCE ranks the infected nodes in a descendent order according to infection eccentricity.

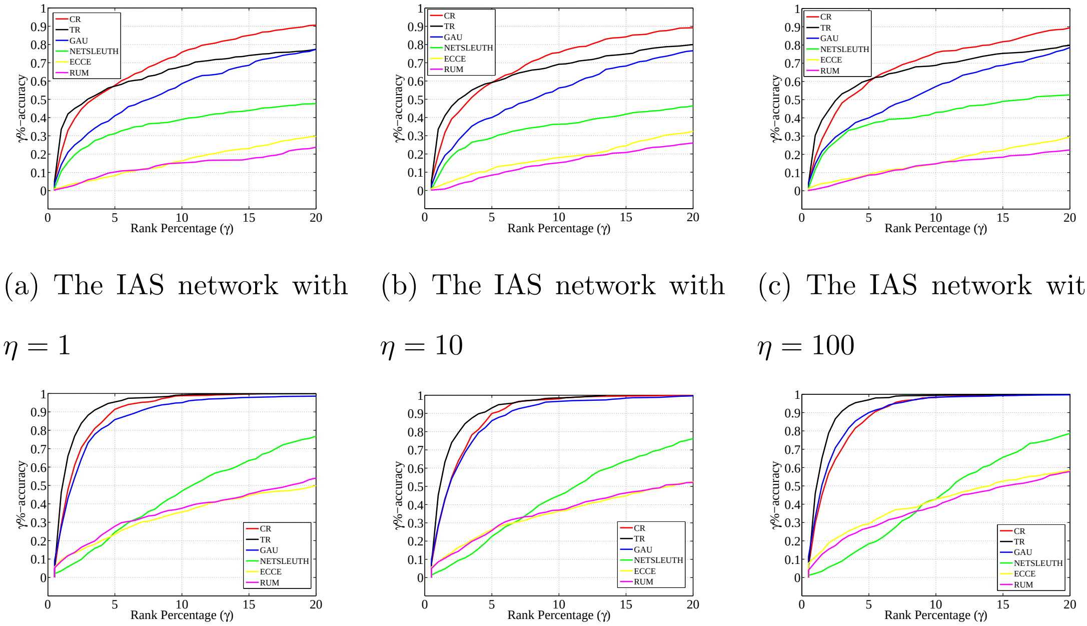
- NETSLEUTH: NETSLEUTH was proposed in (Prakash *et al.*, 2012). The algorithm constructs a submatrix of the infected nodes based on the graph Laplacian of the network and then ranks the infected nodes according to the eigenvector corresponding to the largest eigenvalue of the submatrix.

- Gaussian heuristic (GAU): Gaussian heuristic is an algorithm proposed in (Pinto *et al.*, 2012), which utilizes partial timestamp information. The algorithm is similar to CR in spirit, but uses the breadth-first search tree as the spreading tree for each infected node.

In the four algorithms above, RUM, ECCE, and NETSLEUTH only use topological information of the network, and do not exploit the timestamp information. GAU utilizes partial timestamp information.

In this set of experiments, we assume the infection time of each infection follows a truncated Gaussian distribution with $\eta = \{1, 10, 100\}$ and $\sigma = 100$. In each simulation, a source node was chosen uniformly across node degree to avoid the bias towards small degree nodes (In the IAS network, 3,720 out of the 10,670 nodes have degree one). In particular, the nodes were grouped into M bins such that the nodes in the m^{th} bin $(1 \leq m \leq M-1)$ have degree m and the nodes in the M^{th} bin have degree $\geq M$. In each simulation, we first randomly and uniformly picked a bin, and then randomly and uniformly pick a node from the selected bin. We simulated the diffusion process and terminated the process when having 200 infected nodes. For the IAS network, we chose $M = 20$; and for the PG network, we chose $M = 10$. Since there are less than 10 nodes with degree 21 and the total number of nodes with degree larger than 20 is 205 in the IAS network. Therefore, we use 20 bins to make sure there are enough nodes in each bins. On the other hand, the maximum degree of the PG network is only 19, so we use 10 bins in the PG network.

We selected 50% infected nodes (100 nodes) and revealed their infection time. The source node was always excluded from these 100 nodes so that the infection time of the source node was always unknown. We repeated the simulation 500 times to compute the average γ%-accuracy. Recall the γ%-accuracy is the probability with which the source is ranked among top γ percent.



(a) The IAS network with $\eta = 1$

(b) The IAS network with $\eta = 10$

(c) The IAS network with $\eta = 100$

(d) The PG network with $\eta = 1$

(e) The PG network with $\eta = 10$

(f) The PG network with $\eta = 100$

Figure 3.4: Comparison with existing algorithms with 50% timestamps

The results on the IAS and PG networks are presented in Figure 3.4 where the performance are consistent for different η values. Recall that RUM, ECCE and NETLEUTH only use topological information.

- **Observation 1:** In both networks, CR and TR perform much better than the other algorithms in the IAS network. In PG network, TR, CR and GAU have similar performance which dominates other algorithms due to the utilization of the timestamp information. In particular, in the IAS network, the 10%-accuracy of CR is 0.76 while 10%-accuracy of GAU and NETSLEUTH is 0.57 and 0.43, respectively when $\eta = 100$. In the PG network, the 10%-accuracy of TR is 0.99 while that of GAU and NETSLEUTH is 0.98 and 0.43, respectively.

- **Observation 2:** Most algorithms, except NETSLEUTH, have higher γ%-accuracy in the PG network than in the IAS network. We conjecture that it is because the IAS network has a small diameter and contains hub nodes while the PG network is more tree-like.

- **Observation 3:** NETSLEUTH dominates ECCE and RUM in the IAS network, but performs worse than ECCE and RUM in the PG network when $\gamma \leq 10$. Furthermore, while all other algorithms have higher γ-accuracy in IAS than in PG, NETSLEUTH has lower γ-accuracy in IAS than in PG when $\gamma < 10$. A similar phenomenon will be observed in a later simulation as well.

- **Observation 4:** CR performs better in the IAS network when $\gamma \geq 5$ while TR performs better in the PG network.

3.4.3 The Impact of Timestamp Distribution

In the previous set of simulations, the revealed timestamps were uniformly chosen from all timestamps except the timestamp of the source, which was always excluded. We call this *unbiased distribution.* In this set of experiments, we study the impact of the distribution of the timestamps. We compared the unbiased distribution with a distribution under which nodes with larger infection time are selected with higher probability. In particular, we selected nodes iteratively. Let $\mathcal{N}^k$ denote the set of remaining infected nodes after selecting k nodes, then the probability that Node i is selected in the next step is

$$p_i^{(k)} = \frac{t_i - t_s}{\sum_{j\in\mathcal{N}^k}(t_j - t_s)},$$

where t_s is the infection time of the source. We call this *time biased distribution.*

In this section, we evaluated the performance of our algorithms and GAU with different sizes of observed timestamps and different distributions of the observed timestamps. All the experiment setups are the same as in Section 3.4.2. We evaluate the algorithms with $\eta = \{1, 10, 100\}$ and the results of different number of timestamps are shown in Figure 3.5.

Note that the performance of RUM, ECCE and NETSLEUTH are independent of timestamp distribution and size, so we did not include these algorithms in the figures. From the figure, we have the following observations:

- **Observation 5:** We varied the size of observed timestamps from 10% to 90%. As we expected, the γ%-accuracy increases as the size increases under both CR and TR. Interestingly, in the IAS network, the 10%-accuracy of GAU is worse than TR and CR when more than 20% of the timestamps are observed. We conjecture this is because in small world networks such as the IAS network, the spreading tree is very different from the breadth-first search tree rooted at the source. Since GAU always uses the breadth-first search trees regardless of the size of timestamps, more timestamps do not result in a more accurate spreading tree. The spreading tree constructed by EIF, on the other hand, depends on the size of timestamps and is more accurate as the size of timestamps increases.

- **Observation 6:** In both networks, the time-biased distribution results in 5% to 15% reduction of the γ%-accuracy. This shows that earlier timestamps provide more valuable information for locating the source. However, the trends and relative performance of the three algorithms are similar to those in the unbiased case.

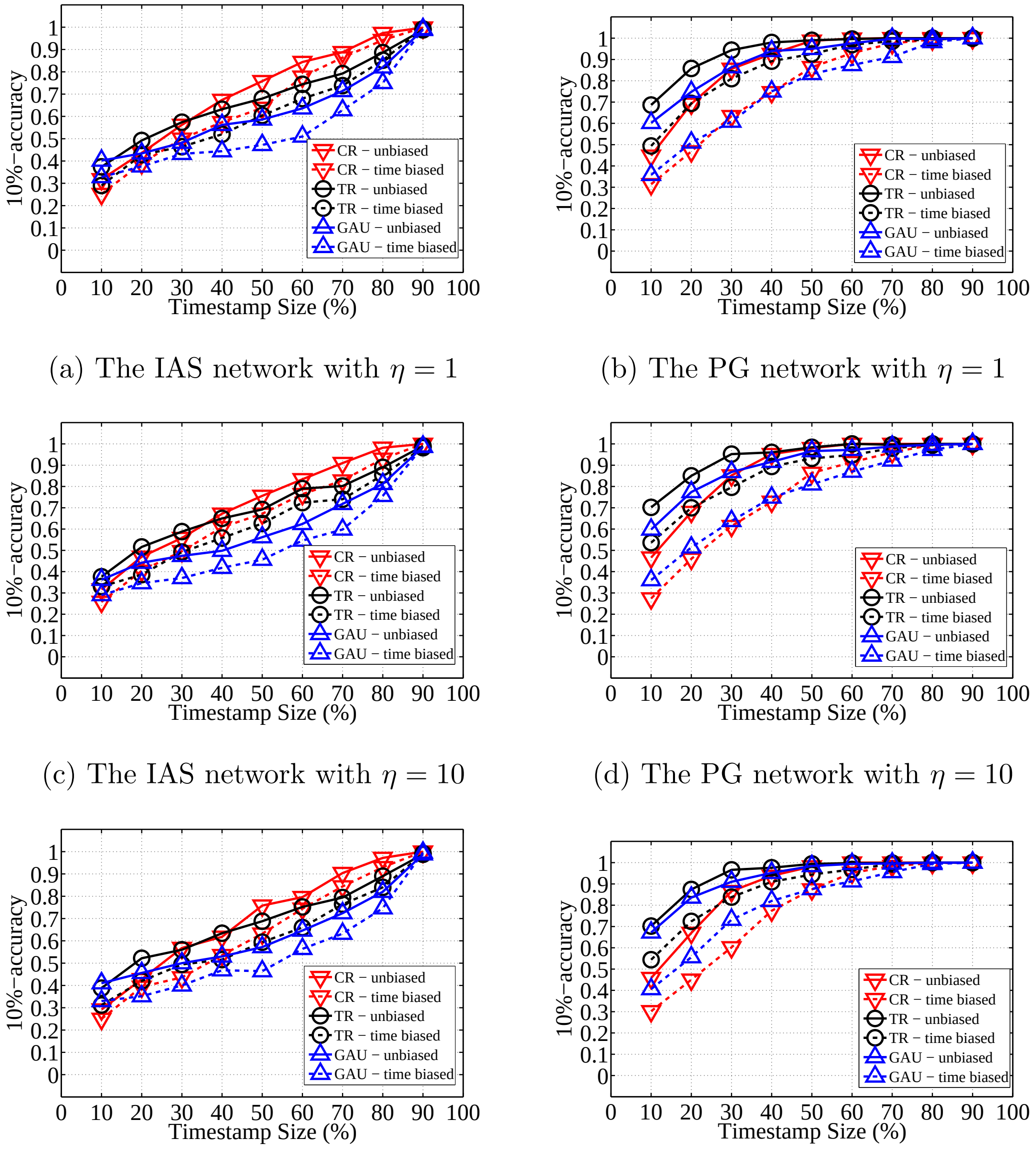
- **Observation 7:** CR performs better in the IAS network when the timestamp size is larger than 40%; and TR performs better in the PG network.

- **Observation 8:** The γ%-accuracy is much higher in the PG network than that in the IAS network under both the unbiased distribution and time-biased distribution. For example, with the time-biased distribution and 20% of timestamps, the 10%-accuracy of TR is 0.87 in PG and is only 0.52 in IAS when $\eta = 100$. This again confirms that the source localization problem is more difficult in networks with small diameters and hub nodes.

3.4.4 The Impact of the Diffusion Model

In all previous experiments, we used the truncated Gaussian model for diffusion. We now study the robustness of CR and TR to the diffusion models. We conducted the experiments using the IC model (Kempe *et al.*, 2003) and SpikeM model (Matsubara *et al.*, 2012) for diffusion. Both models are time slotted, so are very different from the truncated Gaussian model. In the IC model, each infected node has only one chance to infect each of its neighbors. If the infection failed, the node cannot make more attempts. In the experiments, the infection probability along each edge is selected with a uniform distribution over $(0, 1)$. SpikeM model has been shown to match the patterns of real-world information diffusion well. In the SpikeM model, infected nodes become less infectious as time increases. Furthermore, the activity level of a user in different time periods of a day varies to match the rise and fall patterns of information diffusion in the real world. In our experiments, we used the parameter set C5 in Table 3 of (Matsubara *et al.*, 2012) which was obtained based on MemeTracker dataset. The results are shown in Figure 3.6, where in each figure, the size of timestamps varies from 10% to 90%.

- **Observation 9:** Under both the IC and SpikeM models, the GAU algorithm has a better performance when less than 20% timestamps are observed in the IAS network. The performance of TR and CR dominate GAU when more than 20% timestamps are observed. For the PG network, the performances of TR



(a) The IAS network with $\eta = 1$

(b) The PG network with $\eta = 1$

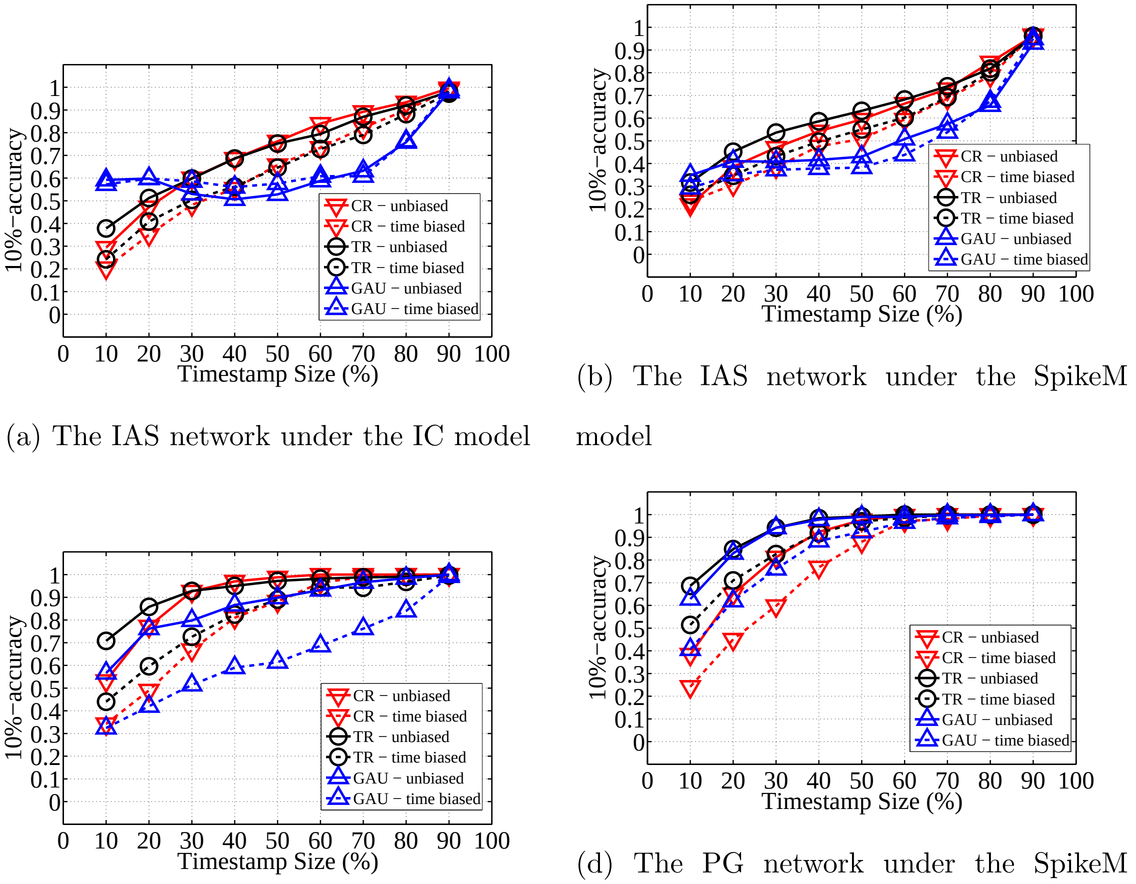
(c) The IAS network with $\eta = 10$

(d) The PG network with $\eta = 10$

(e) The IAS network with $\eta = 100$

(f) The PG network with $\eta = 100$

Figure 3.5: The impacts of the distribution and size of timestamps



(a) The IAS network under the IC model

(b) The IAS network under the SpikeM model

(c) The PG network under the IC model

(d) The PG network under the SpikeM model

Figure 3.6: The performance of CR, TR and GAU under different diffusion models

and CR are better than GAU under the IC model, and the performance of TR is better than GAU under the SpikeM model.

Remark 5: Another popular diffusion model is the Linear Threshold (LT) model (Kempe *et al.*, 2003). However, in the experiments, we found that it is difficult for a single source to infect more than 150 nodes under the LT model. Therefore, we only conducted experiments with the IC model.

3.4.5 The Impact of Network Topology

In the previous simulations, we have observed that locating the source in the PG network is easier than in the IAS network. We conjecture that it is because the IAS network is a small-world network while the PG network is more tree-like. To verify this conjecture, we removed edges from the IAS network to observe the change of γ%-accuracy as the number of removed edges increases. For each removed edge, we randomly picked one edge and removed it if the network remains to be connected after the edge is removed. We used the truncated Gaussian model and all other settings are the same as those in Section 3.4.2. The results are shown in Figure 3.7.

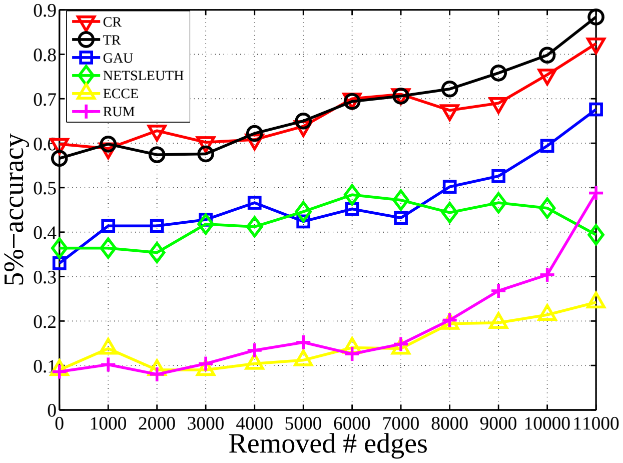
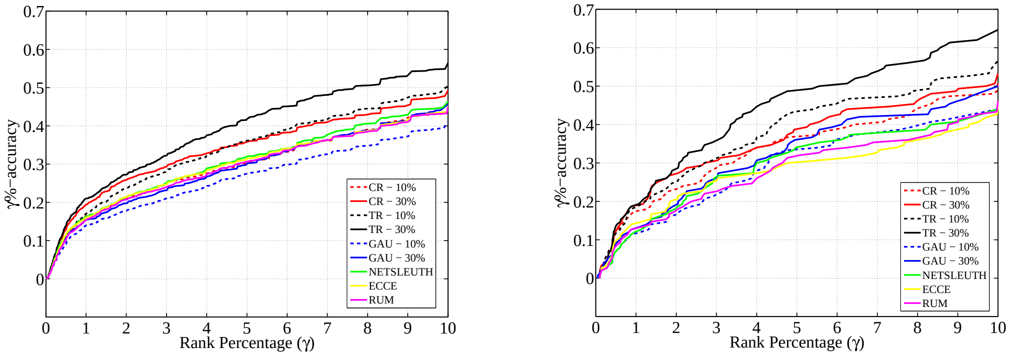


Figure 3.7: The γ%-accuracy as the number of removed edges increases

- **Observation 10:** After removing 11,000 edges, the ratio of the number of edges to the number of nodes is $11,002/10,670 = 1.03$, so the network is tree-like. As showed in Figure 3.7, the 5%-accuracy of all algorithms, except NETSLEUTH, improves as the number of the removed edges increases, which confirms our conjecture. The 5%-accuracy of NETSLEUTH starts to decrease when the number of removed edges is more than $6,000$. This is consistent with the observation we



(a) All tweets

(b) Resample by degree

Figure 3.8: Performance on Weibo data

had in Figure 3.4, in which the 5% accuracy of NETSLUETH in PG is worse than that in IAS.

3.4.6 Weibo Data Evaluation

In this section, we evaluated the performance of our algorithms with real-world network and real-world information spreading. The dataset is the Sina Weibo[3] data, provided by the WISE 2012 challenge[4]. Sina Weibo is the Chinese version of Twitter, and the dataset includes a friendship graph and a set of tweets.

The friendship graph is a directed graph with 265,580,802 edges and 58,655,849 nodes. The tweet dataset includes 369,797,719 tweets. Each tweet includes the user ID and post time of the tweet. If the tweet is a retweet of some tweet, it includes the tweet ID of the original tweet, the user who post the original tweet, the post time of the original tweet, and the retweet path of the tweet which is a sequence of user IDs. For example, the retweet path $a \to b \to c$ means that user b retweeted user a's tweet, and user c retweeted user b's.

[3] `http://www.weibo.com/`

[4] `http://www.wise2012.cs.ucy.ac.cy/challenge.html`

Average Tweet cascade size (number of nodes)	332.19
Average diameter (longest shortest path)	6.86
Average out degree	3.60

Table 3.4: Statistics of extracted tweet cascades

We selected the tweets with more than 1,500 retweets. For each tweet, all users who retweet the tweet are viewed as infected nodes and we extracted the subnetwork induced by these users. We also added those edges on the retweet paths to the subnetwork if they are not present in the friendship graph, by treating them as missing edges in the friendship network. The user who posts the original tweet is regarded as the source. If there does not exist a path from the source to an infected node along which the post time is increasing, the node was removed from the subnetwork. In addition, to make sure we have enough timestamps, we remove the samples with less than 30% timestamps.

After the above preprocessing, we have 1,170 tweets with at least 30% observed timestamps. Some statistics of the extracted tweet cascades are listed in Table 3.4.

Similar to Section 3.4.2 in the chapter, we grouped the tweets into five bins according the degree of the source in the friendship graph. In the k^{th} bin (for $k = 1, 2, 3, 4$), the degree of the source is between $8000(k-1)$ to $8000k - 1$. In the 5^{th} bin, the degree of the source is at least $32,000$. The number of tweets in the bins are $[568 \quad 147 \quad 70 \quad 68 \quad 317]$. From each bin, we draw 30 samples without replacement. For completeness, we also evaluated the performance with all 1,170 tweets. The results are summarized in Figure 3.8. Figure 3.8a shows the performance with all tweets samples and Figure 3.8b shows the performance if we resample the tweets by the above degree bins. The observed timestamps are uniformly selected

Tweet cascade size Number of samples	[10,200) 285	[200,400) 126	[400,600) 106	[600,800) 76	[800,∞) 145
CR-30%	0.87	0.82	0.71	0.55	0.63
CR-10%	0.92	0.70	0.50	0.47	0.60
TR-30%	**0.95**	**0.91**	**0.84**	**0.79**	**0.86**
TR-10%	0.94	0.79	0.71	0.64	0.69
GAU-30%	0.93	0.73	0.55	0.47	0.57
GAU-10%	0.91	0.67	0.41	0.41	0.43
NETSLEUTH	0.92	0.76	0.58	0.55	0.55
ECCE	0.91	0.68	0.55	0.57	0.56
RUM	0.94	0.64	0.63	0.53	0.48

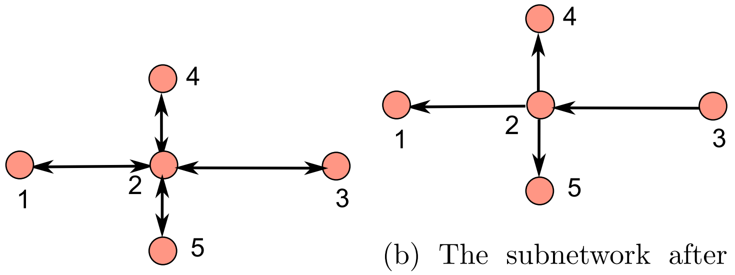
Table 3.5: 10%-accuracy for different tweet cascade sizes

from the available timestamps and the source node is excluded. We also investigate the 10%$-$accuracy for different tweet cascade sizes. The results are shown in Table 3.5. The reason that the first tweet cascade size bin is $[10, 200)$ is that the samples with <10 nodes will always have zero 10%-accuracy.

- **Observation 11:** Figure 3.8 shows that CR and TR dominates GAU with both 10% and 30% of timestamps. In particular for the resample by degree case, TR performs very well and dominates all other algorithms with a large margin. The 10%-accuracy of TR with 30% timestamps is around 0.64 while that of CR is 0.53 and that of NETSLEUTH is only 0.4.

- **Observation 12:** As shown in Table 3.5, for small cascade sizes, all methods have similar accuracy. When the cascade size increases, the performance of our TR algorithm with 30% timestamps dominates all other algorithms. In particular, with same amount of timestamps, TR is much better than GAU which again demonstrated the effectiveness of our algorithm.

Summary: From the synthetic data and real data evaluations, we have seen that



(a) The subnetwork before modification

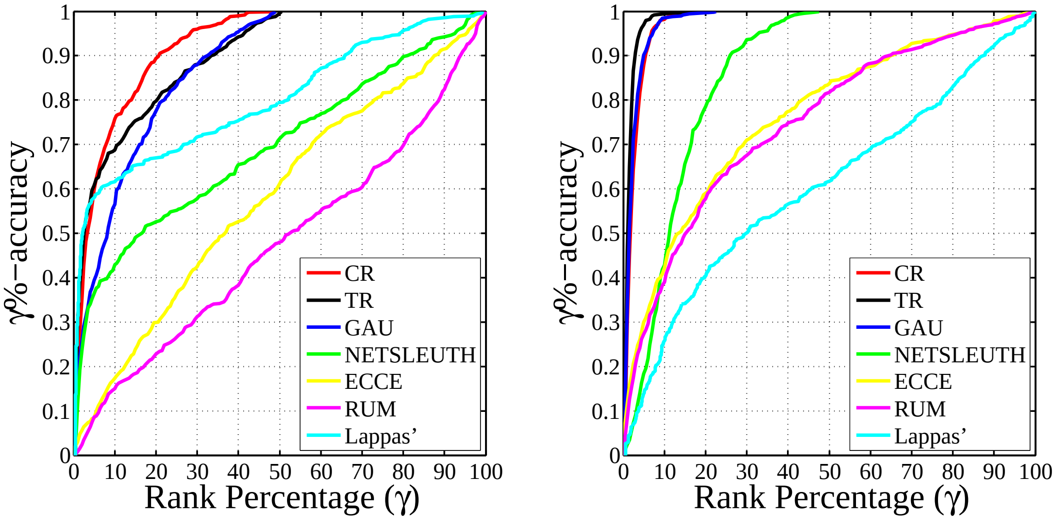
(b) The subnetwork after including information 2 → 3

Figure 3.9: An example of extensions with direction information

both TR and CR perform better than existing algorithms, and are robust to diffusion models and timestamp distributions. Furthermore, TR performs better than CR in most cases. CR performs better than TR only in the IAS network when the sample size is large ($\geq 30\%$ under the truncated Gaussian diffusion, $\geq 50\%$ under the IC model and $\geq 70\%$ under the SpikeM model). More simulation results can be found in Appendix 3.6.

3.5 Extensions

In some practical scenarios, we may have other side information than timestamps such as *who infected whom.* This side information can be incorporated in the algorithm by modifying the network g. Consider the example in Figure 3.9a. If we know that Node 2 was infected by Node 3, then we can removed all incoming edges to Node 2, except $3 \to 2$, and the edge $2 \to 3$ to obtain a modified g as shown in Figure 3.9b. We can then apply CR and TR on the modified graph to rank the observed infected nodes.



(a) γ%-accuracy in the IAS network

(b) γ%-accuracy in the PG network

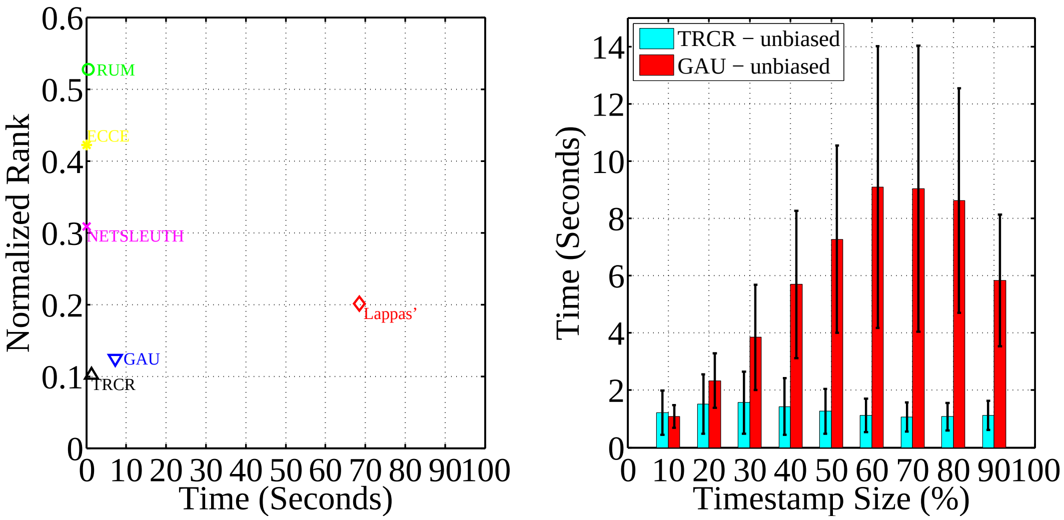
Figure 3.10: The performance comparison to the Lappas' algorithm

3.6 Additional Experimental Evaluation

In this section, we present additional experiments we conducted, including the comparison to Lappas' algorithm under the IC model, the evaluation of the algorithms' scalability and the evaluation using normalized rank.

3.6.1 Comparison to Lappas' Algorithm (Lappas et al.*, 2010)*

In this section, we evaluate the performance of the algorithm in (Lappas *et al.*, 2010) (Lappas' algorithm). Lappas' algorithm was developed for the IC model and requires the infection probabilities of the IC model. Therefore, we only compared the algorithm in (Lappas *et al.*, 2010) on the IC model and the results are shown in Figure 3.10. The experiments settings are the same as those in Section 3.4.2. We assume 50% timestamps are observed for the TR, CR and GAU algorithms. As shown in Figure 3.10, the γ%-accuracy of Lappas' algorithm on the IAS network is significantly smaller than the TR and CR algorithms when $\gamma \geq 10$. In the PG



(a) Normalized rank versus computation time (50 % timestamps observed)

(b) Timestamp size versus computation time

Figure 3.11: Execution time in the IAS network under the IC model

network, the TR and CR algorithms dominates Lappas' algorithm for all γ.

3.6.2 Scalability

We measured the execution time of the algorithms as shown in Figure 3.11. The experiments are conducted on a Intel Core i5-3210M CPU with four cores and 8G RAM with a Windows 7 Professional 64 bit system. All algorithms are implemented with python 2.7. All the other settings are the same as those in Section 5.2 with $\eta = 100$. As shown in Figure 3.11, CR and TR are more than six times faster than GAU when 50% timestamps are observed. Although some other algorithms which do not use timestamps are faster, their performances are worse than TR, CR and GAU. Lappas' algorithm is significantly slower than all the algorithms since Lappas' algorithm is based on the full network while other algorithms are only based on the network with infected nodes or the neighbors of the infected nodes. In addition, as shown in Figure 3.11b, the mean and the standard deviation of the running time of TR and CR are much smaller than those of GAU when the available timestamps are more than 10%. Furthermore, the running time of TR and CR remains roughly the same as the number of timestamps increases while the running time of GAU increases significantlty initially and then decreases a little bit. The decrease is because when more timestamps are observed, only the infected nodes with unobserved timestamps and the node which has the earliest observed timestamps could be the source which reduces the number of candidates hence the total running time.

3.6.3 Normalized Rank

In addition to the γ%-accuracy, we further evaluated the performance of the algorithms using the normalized rank, which is defined to be the ratio between the rank of the actual source and the total number of infected nodes. The observations are similar to the γ%-accuracy except that CR performs better in the IAS network than TR in most cases and TR performs better in the PG network. The difference between GAU and TR & CR are smaller. The results show TR and CR not only achieve much better "accuracy-at-the-top", but also improve the normalized rank in most cases. We next present a short summary for each set of simulations.

The Impact of Timestamp Distribution

Table 3.6, 3.7, 3.8, 3.9, 3.10 and 3.11 show the normalized rank for the truncated Gaussian model for the IAS network and the PG network. The settings of the experiments are same as those in Section 3.4.3. In the IAS network, the CR algorithm yields the smallest normalized ranks and standard deviations when there are more than 10% of timestamps are observed. In the PG network, TR yields the smallest normalized ranks and standard deviations.

The Impact of the Diffusion Model

Table 3.12, 3.13, 3.14 and 3.15 show the normalized rank under the IC model and SpikeM model. The settings are the same as that in Section 3.4.4. GAU has better or similar performance as TR and CR when the fraction of observed timestamps is small, but yields a larger normalized rank when the number of observed timestamps increases.

The Impact of Network Topology

Table 3.16 shows the normalized rank when we remove the edges from the IAS network. The settings are the same as that in Section 3.4.5 and CR dominates in this case.

Weibo Data Evaluation

Table 3.17 shows the normalized rank for the Weibo data. The settings are the same as that in Section 3.4.6. We observed that the CR algorithm with 30% timestamps has the minimum normalized rank for all tweet cascades sizes.

Timestamp Size	CR	TR	GAU	CR (Biased)	TR (Biased)	GAU (Biased)
10%	0.29 ± 0.25	0.31 ± 0.29	0.25 ± 0.25	0.32 ± 0.24	0.36 ± 0.29	0.29 ± 0.25
20%	0.18 ± 0.18	0.23 ± 0.25	0.21 ± 0.21	0.22 ± 0.20	0.27 ± 0.26	0.25 ± 0.22
30%	0.14 ± 0.15	0.17 ± 0.20	0.18 ± 0.18	0.17 ± 0.17	0.21 ± 0.22	0.21 ± 0.19
40%	0.11 ± 0.13	0.14 ± 0.17	0.14 ± 0.16	0.13 ± 0.13	0.17 ± 0.18	0.18 ± 0.16
50%	0.07 ± 0.09	0.11 ± 0.14	0.13 ± 0.13	0.10 ± 0.11	0.13 ± 0.15	0.15 ± 0.14
60%	0.06 ± 0.07	0.08 ± 0.10	0.10 ± 0.10	0.07 ± 0.07	0.10 ± 0.12	0.13 ± 0.11
70%	0.04 ± 0.05	0.06 ± 0.08	0.07 ± 0.07	0.05 ± 0.05	0.07 ± 0.08	0.09 ± 0.08
80%	0.03 ± 0.03	0.04 ± 0.05	0.05 ± 0.05	0.03 ± 0.03	0.04 ± 0.05	0.06 ± 0.05
90%	0.02 ± 0.01	0.02 ± 0.02	0.03 ± 0.03	0.02 ± 0.02	0.03 ± 0.02	0.04 ± 0.03

Table 3.6: Normalized rank (mean ± standard deviation) for different distributions and sizes of timestamps on the IAS network when $\eta = 1$

Timestamp Size	CR	TR	GAU	CR (Biased)	TR (Biased)	GAU (Biased)
10%	0.27 ± 0.23	0.30 ± 0.28	0.26 ± 0.24	0.31 ± 0.24	0.34 ± 0.30	0.30 ± 0.26
20%	0.18 ± 0.18	0.23 ± 0.26	0.21 ± 0.22	0.21 ± 0.20	0.27 ± 0.25	0.26 ± 0.23
30%	0.14 ± 0.15	0.17 ± 0.20	0.19 ± 0.19	0.16 ± 0.16	0.21 ± 0.22	0.23 ± 0.20
40%	0.10 ± 0.12	0.13 ± 0.17	0.16 ± 0.16	0.13 ± 0.13	0.16 ± 0.18	0.19 ± 0.17
50%	0.08 ± 0.09	0.10 ± 0.14	0.13 ± 0.13	0.10 ± 0.10	0.13 ± 0.15	0.16 ± 0.13
60%	0.05 ± 0.06	0.07 ± 0.10	0.10 ± 0.10	0.07 ± 0.07	0.09 ± 0.10	0.13 ± 0.11
70%	0.04 ± 0.05	0.06 ± 0.08	0.08 ± 0.08	0.05 ± 0.06	0.07 ± 0.08	0.10 ± 0.08
80%	0.02 ± 0.02	0.04 ± 0.05	0.06 ± 0.05	0.04 ± 0.04	0.04 ± 0.05	0.07 ± 0.05
90%	0.02 ± 0.01	0.02 ± 0.02	0.03 ± 0.03	0.02 ± 0.02	0.03 ± 0.02	0.04 ± 0.03

Table 3.7: Normalized rank (mean ± standard deviation) for different distributions and sizes of timestamps on the IAS network when $\eta = 10$

Timestamp Size	CR	TR	GAU	CR (Biased)	TR (Biased)	GAU (Biased)
10%	0.29 ± 0.23	0.31 ± 0.29	0.24 ± 0.23	0.32 ± 0.24	0.35 ± 0.29	0.29 ± 0.25
20%	0.19 ± 0.18	0.22 ± 0.25	0.20 ± 0.20	0.22 ± 0.19	0.26 ± 0.25	0.25 ± 0.22
30%	0.14 ± 0.16	0.18 ± 0.21	0.17 ± 0.18	0.18 ± 0.16	0.21 ± 0.22	0.21 ± 0.19
40%	0.11 ± 0.11	0.13 ± 0.17	0.15 ± 0.16	0.13 ± 0.13	0.17 ± 0.18	0.17 ± 0.16
50%	0.08 ± 0.09	0.10 ± 0.13	0.12 ± 0.12	0.10 ± 0.10	0.14 ± 0.15	0.16 ± 0.13
60%	0.06 ± 0.07	0.08 ± 0.10	0.10 ± 0.10	0.07 ± 0.07	0.10 ± 0.11	0.12 ± 0.11
70%	0.04 ± 0.04	0.06 ± 0.07	0.08 ± 0.08	0.05 ± 0.05	0.07 ± 0.08	0.09 ± 0.08
80%	0.03 ± 0.03	0.04 ± 0.05	0.05 ± 0.05	0.04 ± 0.03	0.05 ± 0.05	0.06 ± 0.05
90%	0.02 ± 0.01	0.02 ± 0.02	0.03 ± 0.03	0.02 ± 0.02	0.02 ± 0.02	0.04 ± 0.03

Table 3.8: Normalized rank (mean ± standard deviation) for different distributions and sizes of timestamps on the IAS network when $\eta = 100$

Timestamp Size	CR	TR	GAU	CR (Biased)	TR (Biased)	GAU (Biased)
10%	0.17 ± 0.14	0.10 ± 0.12	0.12 ± 0.12	0.21 ± 0.16	0.17 ± 0.17	0.19 ± 0.16
20%	0.09 ± 0.09	0.06 ± 0.08	0.08 ± 0.10	0.14 ± 0.11	0.09 ± 0.10	0.14 ± 0.13
30%	0.06 ± 0.05	0.04 ± 0.04	0.06 ± 0.07	0.10 ± 0.08	0.06 ± 0.07	0.11 ± 0.11
40%	0.04 ± 0.04	0.03 ± 0.03	0.04 ± 0.04	0.07 ± 0.06	0.05 ± 0.05	0.08 ± 0.08
50%	0.03 ± 0.02	0.02 ± 0.02	0.03 ± 0.04	0.06 ± 0.05	0.04 ± 0.04	0.06 ± 0.06
60%	0.02 ± 0.01	0.02 ± 0.02	0.02 ± 0.02	0.04 ± 0.04	0.03 ± 0.03	0.05 ± 0.05
70%	0.01 ± 0.01	0.01 ± 0.01	0.02 ± 0.02	0.03 ± 0.03	0.02 ± 0.02	0.04 ± 0.04
80%	0.01 ± 0.01	0.01 ± 0.00	0.02 ± 0.01	0.03 ± 0.02	0.02 ± 0.02	0.03 ± 0.03
90%	0.01 ± 0.00	0.01 ± 0.00	0.01 ± 0.01	0.02 ± 0.01	0.02 ± 0.01	0.02 ± 0.02

Table 3.9: Normalized rank (mean ± standard deviation) for different distributions and sizes of timestamps on the PG network when $\eta = 1$

Timestamp Size	CR	TR	GAU	CR (Biased)	TR (Biased)	GAU (Biased)
10%	0.16 ± 0.14	0.09 ± 0.11	0.12 ± 0.13	0.22 ± 0.17	0.14 ± 0.14	0.19 ± 0.16
20%	0.09 ± 0.09	0.05 ± 0.07	0.08 ± 0.09	0.14 ± 0.11	0.10 ± 0.11	0.14 ± 0.13
30%	0.06 ± 0.05	0.03 ± 0.04	0.05 ± 0.06	0.10 ± 0.08	0.07 ± 0.07	0.11 ± 0.11
40%	0.04 ± 0.03	0.03 ± 0.03	0.04 ± 0.04	0.08 ± 0.07	0.05 ± 0.05	0.08 ± 0.08
50%	0.03 ± 0.02	0.02 ± 0.02	0.03 ± 0.04	0.05 ± 0.05	0.04 ± 0.04	0.07 ± 0.07
60%	0.02 ± 0.01	0.01 ± 0.01	0.03 ± 0.03	0.05 ± 0.04	0.03 ± 0.03	0.05 ± 0.05
70%	0.02 ± 0.01	0.01 ± 0.01	0.02 ± 0.02	0.04 ± 0.03	0.03 ± 0.02	0.04 ± 0.04
80%	0.01 ± 0.01	0.01 ± 0.01	0.02 ± 0.01	0.03 ± 0.02	0.02 ± 0.02	0.03 ± 0.03
90%	0.01 ± 0.00	0.01 ± 0.00	0.01 ± 0.01	0.02 ± 0.01	0.02 ± 0.01	0.02 ± 0.02

Table 3.10: Normalized rank (mean ± standard deviation) for different distributions and sizes of timestamps on the PG network when $\eta = 10$

Timestamp Size	CR	TR	GAU	CR (Biased)	TR (Biased)	GAU (Biased)
10%	0.15 ± 0.14	0.09 ± 0.11	0.10 ± 0.10	0.21 ± 0.15	0.14 ± 0.15	0.17 ± 0.15
20%	0.09 ± 0.09	0.05 ± 0.06	0.06 ± 0.07	0.14 ± 0.11	0.09 ± 0.09	0.12 ± 0.11
30%	0.05 ± 0.05	0.03 ± 0.04	0.04 ± 0.05	0.10 ± 0.08	0.06 ± 0.07	0.08 ± 0.08
40%	0.04 ± 0.03	0.03 ± 0.03	0.03 ± 0.03	0.07 ± 0.06	0.04 ± 0.04	0.07 ± 0.06
50%	0.03 ± 0.02	0.02 ± 0.02	0.03 ± 0.03	0.05 ± 0.04	0.04 ± 0.04	0.05 ± 0.05
60%	0.02 ± 0.01	0.01 ± 0.01	0.02 ± 0.02	0.04 ± 0.03	0.03 ± 0.03	0.04 ± 0.04
70%	0.01 ± 0.01	0.01 ± 0.01	0.02 ± 0.01	0.03 ± 0.02	0.02 ± 0.02	0.03 ± 0.03
80%	0.01 ± 0.01	0.01 ± 0.01	0.01 ± 0.01	0.02 ± 0.02	0.02 ± 0.01	0.03 ± 0.02
90%	0.01 ± 0.00	0.01 ± 0.00	0.01 ± 0.01	0.02 ± 0.01	0.02 ± 0.01	0.02 ± 0.01

Table 3.11: Normalized rank (mean ± standard deviation) for different distributions and sizes of timestamps on the PG network when $\eta = 100$

Timestamp Size	CR	TR	GAU	CR (Biased)	TR (Biased)	GAU (Biased)
10%	0.33 ± 0.26	0.32 ± 0.29	0.18 ± 0.24	0.39 ± 0.27	0.39 ± 0.29	0.18 ± 0.22
20%	0.22 ± 0.23	0.22 ± 0.25	0.16 ± 0.20	0.28 ± 0.24	0.27 ± 0.26	0.16 ± 0.20
30%	0.16 ± 0.19	0.17 ± 0.21	0.16 ± 0.18	0.20 ± 0.20	0.21 ± 0.22	0.15 ± 0.18
40%	0.11 ± 0.15	0.12 ± 0.17	0.16 ± 0.16	0.16 ± 0.18	0.17 ± 0.19	0.14 ± 0.15
50%	0.08 ± 0.11	0.08 ± 0.13	0.13 ± 0.13	0.12 ± 0.14	0.12 ± 0.16	0.12 ± 0.13
60%	0.05 ± 0.08	0.06 ± 0.10	0.11 ± 0.10	0.08 ± 0.10	0.08 ± 0.12	0.10 ± 0.10
70%	0.04 ± 0.06	0.04 ± 0.07	0.08 ± 0.08	0.05 ± 0.07	0.05 ± 0.08	0.09 ± 0.08
80%	0.02 ± 0.04	0.02 ± 0.04	0.06 ± 0.05	0.03 ± 0.04	0.03 ± 0.05	0.06 ± 0.05
90%	0.01 ± 0.02	0.01 ± 0.02	0.03 ± 0.03	0.02 ± 0.02	0.02 ± 0.02	0.03 ± 0.03

Table 3.12: Normalized rank (mean ± standard deviation) for different distributions and sizes of timestamps on the IAS network under the IC model

Timestamp Size	CR	TR	GAU	CR (Biased)	TR (Biased)	GAU (Biased)
10%	0.35 ± 0.26	0.34 ± 0.29	0.27 ± 0.26	0.36 ± 0.27	0.36 ± 0.29	0.31 ± 0.26
20%	0.24 ± 0.22	0.26 ± 0.26	0.24 ± 0.23	0.29 ± 0.23	0.31 ± 0.27	0.25 ± 0.22
30%	0.20 ± 0.19	0.20 ± 0.23	0.21 ± 0.20	0.23 ± 0.20	0.24 ± 0.23	0.23 ± 0.20
40%	0.15 ± 0.16	0.17 ± 0.20	0.19 ± 0.17	0.18 ± 0.17	0.19 ± 0.19	0.19 ± 0.16
50%	0.13 ± 0.13	0.13 ± 0.16	0.17 ± 0.14	0.15 ± 0.14	0.15 ± 0.16	0.18 ± 0.14
60%	0.09 ± 0.10	0.09 ± 0.12	0.13 ± 0.11	0.11 ± 0.11	0.11 ± 0.12	0.14 ± 0.11
70%	0.07 ± 0.08	0.07 ± 0.09	0.10 ± 0.09	0.08 ± 0.08	0.08 ± 0.09	0.11 ± 0.09
80%	0.05 ± 0.05	0.05 ± 0.06	0.08 ± 0.06	0.06 ± 0.05	0.05 ± 0.06	0.07 ± 0.06
90%	0.03 ± 0.03	0.03 ± 0.03	0.04 ± 0.03	0.03 ± 0.03	0.03 ± 0.03	0.05 ± 0.03

Table 3.13: Normalized rank (mean ± standard deviation) for different distributions and sizes of timestamps on the IAS network under the SpikeM model

Timestamp Size	CR	TR	GAU	CR (Biased)	TR (Biased)	GAU (Biased)
10%	0.13 ± 0.13	0.10 ± 0.13	0.13 ± 0.14	0.19 ± 0.15	0.18 ± 0.18	0.22 ± 0.18
20%	0.07 ± 0.08	0.06 ± 0.09	0.09 ± 0.12	0.13 ± 0.11	0.12 ± 0.13	0.17 ± 0.15
30%	0.04 ± 0.04	0.04 ± 0.07	0.07 ± 0.08	0.09 ± 0.08	0.09 ± 0.11	0.13 ± 0.12
40%	0.03 ± 0.03	0.03 ± 0.07	0.05 ± 0.06	0.06 ± 0.05	0.06 ± 0.08	0.11 ± 0.10
50%	0.02 ± 0.02	0.02 ± 0.04	0.04 ± 0.05	0.05 ± 0.04	0.05 ± 0.07	0.10 ± 0.09
60%	0.01 ± 0.01	0.02 ± 0.03	0.04 ± 0.04	0.04 ± 0.03	0.04 ± 0.05	0.09 ± 0.08
70%	0.01 ± 0.01	0.01 ± 0.02	0.03 ± 0.03	0.03 ± 0.02	0.03 ± 0.04	0.07 ± 0.06
80%	0.01 ± 0.01	0.01 ± 0.02	0.02 ± 0.02	0.02 ± 0.02	0.02 ± 0.03	0.06 ± 0.04
90%	0.01 ± 0.00	0.01 ± 0.01	0.02 ± 0.01	0.02 ± 0.01	0.02 ± 0.01	0.03 ± 0.02

Table 3.14: Normalized rank (mean ± standard deviation) for different distributions and sizes of timestamps on the PG network under the IC model

Timestamp Size	CR	TR	GAU	CR (Biased)	TR (Biased)	GAU (Biased)
10%	0.18 ± 0.15	0.10 ± 0.12	0.11 ± 0.11	0.24 ± 0.16	0.15 ± 0.15	0.17 ± 0.14
20%	0.10 ± 0.09	0.06 ± 0.07	0.06 ± 0.07	0.14 ± 0.10	0.09 ± 0.08	0.11 ± 0.10
30%	0.06 ± 0.06	0.03 ± 0.04	0.04 ± 0.04	0.10 ± 0.08	0.06 ± 0.06	0.07 ± 0.07
40%	0.04 ± 0.04	0.03 ± 0.02	0.03 ± 0.03	0.07 ± 0.05	0.04 ± 0.04	0.05 ± 0.05
50%	0.03 ± 0.03	0.02 ± 0.02	0.02 ± 0.02	0.05 ± 0.04	0.04 ± 0.03	0.04 ± 0.04
60%	0.02 ± 0.02	0.02 ± 0.01	0.02 ± 0.02	0.04 ± 0.03	0.03 ± 0.02	0.03 ± 0.03
70%	0.02 ± 0.01	0.02 ± 0.01	0.02 ± 0.01	0.03 ± 0.02	0.02 ± 0.02	0.03 ± 0.02
80%	0.02 ± 0.01	0.01 ± 0.00	0.01 ± 0.01	0.03 ± 0.02	0.02 ± 0.01	0.02 ± 0.02
90%	0.01 ± 0.00	0.01 ± 0.00	0.01 ± 0.01	0.02 ± 0.01	0.02 ± 0.01	0.02 ± 0.01

Table 3.15: Normalized rank (mean ± standard deviation) for different distributions and sizes of timestamps on the PG network under the SpikeM model

Edges Removed	CR	TR	GAU	NETSLEUTH	ECCE	RUM
0	0.08 ± 0.09	0.10 ± 0.13	0.12 ± 0.12	0.31 ± 0.32	0.42 ± 0.30	0.53 ± 0.32
1000	0.08 ± 0.10	0.10 ± 0.13	0.13 ± 0.13	0.29 ± 0.31	0.41 ± 0.30	0.52 ± 0.33
2000	0.07 ± 0.09	0.11 ± 0.14	0.13 ± 0.13	0.30 ± 0.31	0.42 ± 0.29	0.54 ± 0.32
3000	0.07 ± 0.09	0.11 ± 0.14	0.13 ± 0.13	0.25 ± 0.30	0.42 ± 0.29	0.52 ± 0.33
4000	0.07 ± 0.08	0.09 ± 0.13	0.12 ± 0.12	0.26 ± 0.30	0.42 ± 0.30	0.49 ± 0.34
5000	0.07 ± 0.08	0.09 ± 0.12	0.12 ± 0.12	0.25 ± 0.29	0.39 ± 0.29	0.48 ± 0.33
6000	0.06 ± 0.08	0.08 ± 0.12	0.11 ± 0.12	0.21 ± 0.26	0.35 ± 0.29	0.41 ± 0.31
7000	0.06 ± 0.08	0.08 ± 0.12	0.12 ± 0.12	0.21 ± 0.27	0.34 ± 0.27	0.39 ± 0.31
8000	0.06 ± 0.08	0.07 ± 0.12	0.10 ± 0.11	0.21 ± 0.26	0.33 ± 0.28	0.38 ± 0.32
9000	0.06 ± 0.08	0.06 ± 0.11	0.10 ± 0.11	0.19 ± 0.25	0.32 ± 0.30	0.35 ± 0.32
10000	0.05 ± 0.06	0.05 ± 0.09	0.08 ± 0.10	0.18 ± 0.23	0.34 ± 0.29	0.32 ± 0.32
11000	0.05 ± 0.07	0.03 ± 0.07	0.07 ± 0.10	0.14 ± 0.21	0.33 ± 0.29	0.29 ± 0.35

Table 3.16: Normalized rank (mean ± standard deviation) as the number of removed edges increases in the IAS network

Tweet cascade size	[10,200)	[200,400)	[400,600)	[600,800)	$[800,\infty)$
Number of samples	285	126	106	76	145
CR-30%	**0.05** ± 0.05	**0.04** ± 0.07	**0.07** ± 0.08	**0.10** ± 0.08	**0.08** ± 0.08
CR-10%	0.21 ± 0.29	0.08 ± 0.10	0.12 ± 0.11	0.14 ± 0.12	0.10 ± 0.10
TR-30%	0.06 ± 0.11	0.08 ± 0.19	0.10 ± 0.19	0.17 ± 0.25	0.10 ± 0.17
TR-10%	0.23 ± 0.30	0.15 ± 0.24	0.21 ± 0.29	0.24 ± 0.30	0.23 ± 0.32
GAU-30%	0.06 ± 0.06	0.06 ± 0.08	0.11 ± 0.11	0.12 ± 0.10	0.12 ± 0.12
GAU-10%	0.06 ± 0.06	0.09 ± 0.11	0.14 ± 0.12	0.15 ± 0.11	0.14 ± 0.12
NETSLEUTH	0.36 ± 0.30	0.43 ± 0.35	0.37 ± 0.30	0.35 ± 0.28	0.36 ± 0.27
ECCE	0.06 ± 0.06	0.08 ± 0.10	0.11 ± 0.10	**0.10** ± 0.10	0.11 ± 0.11
RUM	**0.05** ± 0.05	0.09 ± 0.11	0.10 ± 0.11	0.11 ± 0.10	0.13 ± 0.11

Table 3.17: Normalized rank for different tweet cascade sizes (mean ± standard deviation) on the Weibo dataset

Chapter 4

CONCLUSIONS

In this dissertation, we derived the MAP estimator of the information source on tree networks under the IC model. The SFT algorithm for general networks has been proposed based on the MAP estimator. We proved the SFT algorithm identifies the information source with probability one asymptotically in the ER random graphs when the observation time $t \leq \frac{2}{3}t_u$, which is the first theoretical guarantee on non-tree networks to our best knowledge. We evaluated the performance of SFT on tree networks, the ER random graphs and the IAS networks.

In addition, we proposed two ranking algorithms, CR and TR which utilize the partial timestamps to improve the accuracy. Experimental evaluations on synthetic and real world data demonstrated that CR and TR improved the ranking accuracy significantly compared with existing algorithms under different diffusion models, and perform well in the real-world dataset.

All the algorithms discussed in this dissertation assume a single diffusion source and the set of infected nodes are known. The following directions are worthy exploring: 1) It would be interesting to extend the definition of the Jordan infection center to multiple sources and provide similar theoretical performance guarantees; 2) An efficient and accurate algorithm which utilizes the partial timestamps to detect multiple information sources is another interesting future work; 3) One major assumption of the dissertation is that all the infected nodes are observed. Another possible future work is to extend the SFT algorithm to incomplete observations and to understand the performance of the algorithm with incomplete observations.

REFERENCES

Agaskar, A. and Y. M. Lu, “A fast monte carlo algorithm for source localization on graphs”, in “SPIE Optical Engineering and Applications”, (2013).

Boyd, S., C. Cortes, M. Mohri and A. Radovanovic, “Accuracy at the top”, in “Advances in Neural Information Processing Systems”, pp. 962–970 (2012).

Boyd, S. and L. Vandenberghe, *Convex Optimization* (Cambridge Unversity Press, New York, NY, 2004).

Breiger, R. L. and P. E. Pattison, “Cumulated social roles: The duality of persons and their algebras”, Social Networks **8**, 3, 215 – 256 (1986).

Char, J., “Generation of trees, two-trees, and storage of master forests”, IEEE Trans. Circuit Theory **15**, 3, 228–238 (1968).

Chen, W., C. Wang and Y. Wang, “Scalable influence maximization for prevalent viral marketing in large-scale social networks”, in “Proc. Ann. ACM SIGKDD Conf. Knowledge Discovery and Data Mining (KDD)”, pp. 1029–1038 (2010).

Chen, W., Y. Wang and S. Yang, “Efficient influence maximization in social networks”, in “Proc. Ann. ACM SIGKDD Conf. Knowledge Discovery and Data Mining (KDD)”, pp. 199–208 (2009).

Chen, Z., K. Zhu and L. Ying, “Detecting multiple information sources in networks under the SIR model”, in “Proc. IEEE Conf. Information Sciences and Systems (CISS)”, (Princeton, NJ, 2014).

Dong, W., W. Zhang and C. W. Tan, “Rooting out the rumor culprit from suspects”, in “Proc. IEEE Int. Symp. Information Theory (ISIT)”, pp. 2671–2675 (Istanbul, Turkey, 2013).

Draief, M. and L. Massouli, *Epidemics and rumours in complex networks* (Cambridge University Press, 2010).

Erdos, P. and A. Renyi, “On random graphs I”, Publ. Math. Debrecen **6**, 290–297 (1959).

Garey, M. R. and D. S. Johnson, *Computers and Intractibility: A guide to the theory of NP-completeness* (Macmillan Higher Education, 1979).

Goldenberg, J., B. Libai and E. Muller, “Talk of the network: A complex systems look at the underlying process of word-of-mouth”, Marketing Letters **12**, 3, 211–223 (2001).

Goyal, A., W. Lu and L. V. S. Lakshmanan, “Simpath: An efficient algorithm for influence maximization under the linear threshold model”, in “IEEE Int. Conf. Data Mining (ICDM)”, pp. 211–220 (IEEE Computer Society, 2011).

Gruhl, D., R. Guha, D. Liben-Nowell and A. Tomkins, “Information diffusion through blogspace”, in “Proc. Int. Conf. World Wide Web (WWW)”, pp. 491–501 (New York, NY, 2004).

Harary, F., *Graph theory* (Addison-Wesley, 1991).

Karamchandani, N. and M. Franceschetti, “Rumor source detection under probabilistic sampling”, in “Proc. IEEE Int. Symp. Information Theory (ISIT)”, (Istanbul, Turkey, 2013).

Kempe, D., J. Kleinberg and E. Tardos, “Maximizing the spread of influence through a social network”, in “Proc. Ann. ACM SIGKDD Conf. Knowledge Discovery and Data Mining (KDD)”, pp. 137–146 (Washington DC, 2003).

Lappas, T., E. Terzi, D. Gunopulos and H. Mannila, “Finding effectors in social networks”, in “Proc. Ann. ACM SIGKDD Conf. Knowledge Discovery and Data Mining (KDD)”, pp. 1059–1068 (2010).

Lokhov, A. Y., M. Mézard, H. Ohta and L. Zdeborová, “Inferring the origin of an epidemic with a dynamic message-passing algorithm”, Phys. Rev. E **90**, 012801 (2014).

Luo, W. and W. P. Tay, “Estimating infection sources in a network with incomplete observations”, in “Proc. IEEE Global Conference on Signal and Information Processing (GlobalSIP)”, pp. 301–304 (Austin, TX, 2013a).

Luo, W. and W. P. Tay, “Finding an infection source under the SIS model”, in “Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)”, (Vancouver, BC, 2013b).

Luo, W., W. P. Tay and M. Leng, “Identifying infection sources and regions in large networks”, IEEE Trans. Signal Process. **61**, 2850–2865 (2013).

Matsubara, Y., Y. Sakurai, B. A. Prakash, L. Li and C. Faloutsos, “Rise and fall patterns of information diffusion: model and implications”, in “Proc. Ann. ACM SIGKDD Conf. Knowledge Discovery and Data Mining (KDD)”, pp. 6–14 (Beijing, China, 2012).

Milling, C., C. Caramanis, S. Mannor and S. Shakkottai, “Network forensics: Random infection vs spreading epidemic”, in “Proc. Ann. ACM SIGMETRICS Conf.”, pp. 223–234 (2012).

Mitzenmacher, M. and E. Upfal, *Probability and Computing: Randomized Algorithms and Probabilistic Analysis* (Cambridge University Press, Cambridge, 2005).

Myers, S. A., C. Zhu and J. Leskovec, “Information diffusion and external influence in networks”, in “Proc. Ann. ACM SIGKDD Conf. Knowledge Discovery and Data Mining (KDD)”, pp. 33–41 (Beijing, China, 2012).

Pinto, P. C., P. Thiran and M. Vetterli, “Locating the source of diffusion in large-scale networks”, Phys. Rev. Lett. **109**, 6, 068702 (2012).

Prakash, B. A., J. Vreeken and C. Faloutsos, “Spotting culprits in epidemics: How many and which ones?”, in “IEEE Int. Conf. Data Mining (ICDM)”, pp. 11–20 (Brussels, Belgium, 2012).

Sadikov, E., M. Medina, J. Leskovec and H. Garcia-Molina, “Correcting for missing data in information cascades”, in “Proc. of the Fourth ACM Int. Conf. on Web Search and Data Mining”, pp. 55–64 (2011).

Shah, D. and T. Zaman, “Rumors in a network: Who’s the culprit?”, IEEE Trans. Inf. Theory **57**, 5163–5181 (2011).

Shah, D. and T. Zaman, “Rumor centrality: A universal source detector”, ACM SIGMETRICS Performance Evaluation Review **40**, 1, 199–210 (2012).

Shakarian, P. and V. S. Subrahmanian, *Geospatial Abduction: Principles and Practice* (Springer, 2011).

Shakarian, P., V. S. Subrahmanian and M. L. Sapino, “GAPs: Geospatial abduction problems”, ACM Trans. Intell. Syst. Technol. **3**, 1, 1–27 (2011).

Snow, J., “The cholera near Golden-square, and at Deptford”, Medical Times and Gazette (1854).

Subramanian, V. G. and R. Berry, “Spotting trendsetters: Inference for network games”, in “Proc. Annu. Allerton Conf. Communication, Control and Computing”, (2012).

Wang, Z., W. Dong, W. Zhang and C. W. Tan, “Rumor source detection with multiple observations: fundamental limits and algorithms”, in “Proc. Ann. ACM SIGMETRICS Conf.”, (Austin, TX, 2014).

Zejnilovic, S., J. Gomes and B. Sinopoli, “Network observability and localization of the source of diffusion based on a subset of nodes”, in “Proc. Annu. Allerton Conf. Communication, Control and Computing”, (Monticello, IL, 2013).

Zhu, K. and L. Ying, “Information source detection in the SIR model: A sample path based approach”, IEEE/ACM Trans. Netw. DOI: 10.1109/TNET.2014.2364972 (2014a).

Zhu, K. and L. Ying, “A robust information source estimator with sparse observations”, in “Proc. IEEE Int. Conf. Computer Communications (INFOCOM)”, (Toronto, Canada, 2014b).

APPENDIX A

PROOFS OF CHAPTER 2

A.1 Proof of Theorem 1

Proof. We first analyze the computational complexity of the node ID broadcasting phase of Algorithm 1. The running time of the algorithm is equal to the minimum infection eccentricity and the number of messages each node receives/sends during each time slot is bounded by its degree. To implement Algorithm 1, each node maintains an array of integers of size $|\mathcal{V}(g_i)|$ and an integer counter. We assign an integer index to each infected node. The values of that integer array are the distances from the node to the infected nodes. The integer counter records the number of distinct indexes received. A message only contains the index of an infected node. At each iteration, each node broadcasts the new indexes to its neighbors. When a node receives a new message, it checks the specific index to see whether the index has been received. If not, it updates the value at the corresponding location of the array with the current iteration number, which equals to the distance from the current node to the infected node with the received index. Otherwise, the message is discarded. Then, the node increases the value of its integer counter by one and checks whether the value is $|\mathcal{V}(g_i)|$. All operations mentioned above have constant complexity. Therefore, the complexity of processing one message is $O(1)$. Each edge is used to transmit at most $|\mathcal{V}(g_i)|$ messages in one direction. Hence there are at most $2|\mathcal{V}(g_i)||\mathcal{E}(g_i)|$ messages to be handled. Therefore, the worse case complexity of the node ID broadcasting phase of Algorithm 1 is $O(|\mathcal{V}(g_i)||\mathcal{E}(g_i)|)$.

The complexity of Algorithm 2 is $O(\deg(\mathcal{I}))$ since the number of boundary nodes is bounded by $|\mathcal{I}|$. In addition, Algorithms 2 are called at most $|\mathcal{S}|$ times in Algorithm 1 and $|\mathcal{S}| \leq |\mathcal{V}(g_i)|$. Therefore, the complexity of Algorithm 1 is

$$O(|\mathcal{V}(g_i)||\mathcal{E}(g_i)| + |\mathcal{V}(g_i)|\deg(\mathcal{I}))$$

Note $\mathcal{V}(g_i) = \mathcal{I}$ and $|\mathcal{E}(g_i)| \leq \deg(\mathcal{I})$. The complexity becomes

$$O(|\mathcal{I}|\deg(\mathcal{I})).$$

□

A.2 Proof of Theorem 2

First, we prove the following lemma for neighboring nodes.

Lemma 10. Neighboring nodes inequality *Consider nodes u, v on tree $\tilde{T}$ satisfying the following conditions:*

- $(u, v) \in \mathcal{E}(\tilde{T})$.
- *The observation time follows a distribution such that* $\Pr(t) \geq \Pr(t+1)$ *for all* t.
- *The source is uniformly chosen among all nodes, i.e.,* $\Pr(u) = \Pr(v)$.
- $e(v, \mathcal{I}) > e(u, \mathcal{I})$.

We have

$$\Pr(v|\mathcal{O}) \leq \Pr(u|\mathcal{O})$$

Proof. Consider nodes u, v on a tree $\tilde{T}$ where $(u, v) \in \mathcal{E}(\tilde{T})$. Let t_u, t_v be the observation times associated with u, v. We will show that when $e(v, \mathcal{I}) > e(u, \mathcal{I})$,

$$\Pr(\mathcal{O}|v, t_v = t+1) \leq \Pr(\mathcal{O}|u, t_u = t), \tag{A.1}$$

where $\Pr(\mathcal{O}|v, t_v = t)$ is the probability of the snapshot $\mathcal{O}$ given v is the source and the observation time is at time slot t. .

We adopt an equivalent view of the IC model called *live edge* model (Kempe *et al.*, 2003). In the IC model, after u becomes infected, it attempts to infect its neighbor w with probability q_{uw} once. Therefore, we can assume that a biased coin with parameter q_{uw} is flipped for edge $(u, w) \in \mathcal{E}(g)$ when u tries to infect w in the IC model. Note that the probability of node w is infected by node u remains the same whether the coin is flipped at the moment when node u attempts to infect w or prior to the infection but is revealed for the attempt. Assume the coins of all edges are flipped at the beginning of the infection process. When one node attempts to infect one of its neighbors, we check the stored coin realization to determine whether the infection succeeds. This process is called live edge model and it is equivalent to the IC model since we only change the time of the coin flippings, so the probability of an infection trace remains the same. In the live edge model, the infection process consists of two steps. First, each edge (u, w) flips a biased coin with probability q_{uw} to be a *live edge* prior to the infection starts. After all coin flippings, the graph formed by live edges is called the *live edge graph.* In the second step, the infection spreads over all live edges deterministically, starting from the source. We now analyze SFT under the live-edge model.

Denote by $\mathcal{T}$ the set of all live edge graphs of $\tilde{T}$, i.e.,

$$\mathcal{T} = \{T | \mathcal{E}(T) \subset \mathcal{E}(\tilde{T}), \mathcal{V}(T) = \mathcal{V}(\tilde{T})\}$$

Note there are no loops in $T \in \mathcal{T}$ since T is a subgraph of tree $\tilde{T}$.

Denote by $\mathcal{K}(\mathcal{O}, v, t_v)$ the set of all live edge graphs on which the observation $\mathcal{O}$ is feasible if the source is v and the observation time is t_v. All infected nodes must be within t_v hops from the source and all the observed healthy nodes must be more than t_v hops away from the source in a feasible live edge graph. Formally, we have

$$\mathcal{K}(\mathcal{O}, v, t_v) = \{T \in \mathcal{T} | \forall w \in \mathcal{I}, d^T_{vw} \leq t_v, \forall w \in \mathcal{H}, d^T_{vw} > t_v\}.$$

$\Pr(\mathcal{O}|v, t_v)$ equals to the probability a live edge graph is in set $\mathcal{K}(\mathcal{O}, v, t_v)$ due to the equivalence between the IC model and the live-edge model. The probability of a specific live edge graph is the product of edge live/dead probabilities.

Hence we have

$$\Pr(\mathcal{O}|v, t_v) = \sum_{T \in \mathcal{K}(\mathcal{O}, v, t_v)} \Pr(T),$$

To prove the lemma, we will prove the following claim,

$$\mathcal{K}(\mathcal{O}, v, t_v = t+1) \subset \mathcal{K}(\mathcal{O}, u, t_u = t).$$

To simplify notation, we next assume $t_u = t_v - 1 = t \geq e(u, \mathcal{I})$, and ignore t in the equations. Note that we only consider $t_u \geq e(u, \mathcal{I})$ because $\Pr(\mathcal{O}|u, t_u) = 0$ otherwise.

Consider $T \in \mathcal{K}(\mathcal{O}, v, t_v)$. Denote by T_v^{-u} the tree rooted at v without the subtree starting from u where $(u, v) \in \mathcal{E}(T)$. Note, if $\mathcal{I} \cap \tilde{T}_u^{-v} = \emptyset$, we have $e(v, \mathcal{I}) \leq e(u, \mathcal{I}$ since all infected nodes are on $\tilde{T}_v^{-u}$ which is contradict to the fact that $e(v, \mathcal{I}) > e(u, \mathcal{I})$. Therefore, there exists $w^\dagger \in \mathcal{I} \cap \tilde{T}_u^{-v}$. If $(v, u) \notin \mathcal{E}(T)$, we have $d_{vw^\dagger}^T = \infty > t_v$ which is a contradiction to $T \in \mathcal{K}(\mathcal{O}, v, t_v)$. Hence, we have $(v, u) \in \mathcal{E}(T)$.

- Consider $\tilde{T}_u^{-v}$ part.
  - For any $w \in \tilde{T}_u^{-v} \cap \mathcal{I}$, we have
  $$d_{vw}^T \leq e(v, \mathcal{I}) \leq t_v.$$
  Since $(v, u) \in \mathcal{E}(g)$ and T have no loops, we have
  $$d_{uw}^T = d_{vw}^T - 1 \leq t_v - 1 = t_u.$$
  - For any $w \in \tilde{T}_u^{-v} \cap \mathcal{H}$, we have
  $$d_{vw}^T > e(v, \mathcal{I}) \geq t_v.$$
  Since $(v, u) \in \mathcal{E}(g)$ and T has no loops, we have
  $$d_{uw}^T = d_{vw}^T - 1 > t_v - 1 = t_u.$$
- Consider $\tilde{T}_v^{-u}$ part.
  - For any $w \in \tilde{T}_v^{-u} \cap \mathcal{I}$, if $d_{vw}^{\tilde{T}} \geq e(u, \mathcal{I})$, we have $d_{uw}^{\tilde{T}} = d_{vw}^{\tilde{T}} + 1 \geq e(u, \mathcal{I}) + 1$ which contradicts the definition of infection eccentricity. Therefore, we have
  $$d_{vw}^{\tilde{T}} \leq e(u, \mathcal{I}) - 1.$$
  There is only one path $\mathcal{P}_{vw}$ from v to w in tree $\tilde{T}$. If $\mathcal{P}_{vw} \not\subset \mathcal{E}(T)$, v, w are disconnected in T which contradicts the fact that $d_{vw}^T \leq t_v$. Hence $\mathcal{P}_{vw} \subset \mathcal{E}(T)$. In addition, we have $(v, u) \subset \mathcal{E}(T)$. Hence
  $$d_{uw}^T = d_{vw}^T + 1 = d_{vw}^{\tilde{T}} + 1 \leq e(u, \mathcal{I}) \leq t_u.$$
  - For any $w \in \tilde{T}_v^{-u} \cap \mathcal{H}$, there is only one path $\mathcal{P}_{vw}$ from v to w in tree $\tilde{T}$. If $\mathcal{P}_{vw} \not\subset \mathcal{E}(T)$, we have
  $$d_{uw}^T = \infty > t_u.$$
  If $\mathcal{P}_{vw} \subset \mathcal{E}(T)$,
  $$d_{uw}^T = d_{uw}^{\tilde{T}} = d_{vw}^{\tilde{T}} + 1 = d_{vw}^T + 1 > t_v + 1 > t_u$$
  where $d_{vw}^T > t_v$ because $T \in \mathcal{K}(\mathcal{O}, v, t_v)$.

As a summary, for any $T \in \mathcal{K}(\mathcal{O}, v, t_v)$, we have

$$\forall w \in \mathcal{I}, d_{uw}^T \leq t_u, \forall w \in \mathcal{H}, d_{uw}^T > t_u$$

Therefore, $T \in \mathcal{K}(\mathcal{O}, u, t_u)$.

Hence, we proved

$$\mathcal{K}(\mathcal{O}, v, t_v = t+1) \subset \mathcal{K}(\mathcal{O}, u, t_u = t)$$

which implies

$$\begin{aligned}
\Pr(\mathcal{O}|v, t_v = t+1) &= \sum_{T \in \mathcal{K}(\mathcal{O}, v, t_v = t+1)} \Pr(T) \\
&\leq \sum_{T \in \mathcal{K}(\mathcal{O}, u, t_u = t)} \Pr(T) \\
&= \Pr(\mathcal{O}|u, t_u = t).
\end{aligned}$$

Hence, we proved Inequality (A.1).

Denote by $\Pr(v)$ the probability that v is the source, $\Pr(t)$ the probability that the observation time is t, and $\Pr(\mathcal{O}|v, t)$ is the probability of snapshot $\mathcal{O}$ given v is the source and t is the observation time. Since the observation time t is independent of the source node, we obtain

$$\begin{aligned}
\Pr(v|\mathcal{O}) &= \frac{1}{\Pr(\mathcal{O})} \Pr(v, \mathcal{O}) \\
&= \frac{1}{\Pr(\mathcal{O})} \sum_{t \geq e(v,\mathcal{I})} \Pr(v, t, \mathcal{O}) \\
&= \frac{1}{\Pr(\mathcal{O})} \sum_{t \geq e(v,\mathcal{I})} \Pr(\mathcal{O}|v, t) \Pr(v, t) \\
&= \frac{\Pr(v)}{\Pr(\mathcal{O})} \sum_{t \geq e(v,\mathcal{I})} \Pr(\mathcal{O}|v, t) \Pr(t) \\
&\leq_{\text{(a)}} \frac{\Pr(v)}{\Pr(\mathcal{O})} \sum_{t \geq e(v,\mathcal{I})} \Pr(\mathcal{O}|u, t-1) \Pr(t) \\
&=_{\text{(b)}} \frac{\Pr(u)}{\Pr(\mathcal{O})} \sum_{t \geq e(u,\mathcal{I})} \Pr(\mathcal{O}|u, t) \Pr(t+1) \\
&\leq_{\text{(c)}} \frac{\Pr(u)}{\Pr(\mathcal{O})} \sum_{t \geq e(u,\mathcal{I})} \Pr(\mathcal{O}|u, t) \Pr(t) \\
&= \Pr(u|\mathcal{O})
\end{aligned}$$

(a) is due to Inequality (A.1), (b) is based on $\Pr(u) = \Pr(v)$ and $e(v, \mathcal{I}) = e(u, \mathcal{I}) + 1$, and (c) is based on $\Pr(t) \geq \Pr(t+1)$. □

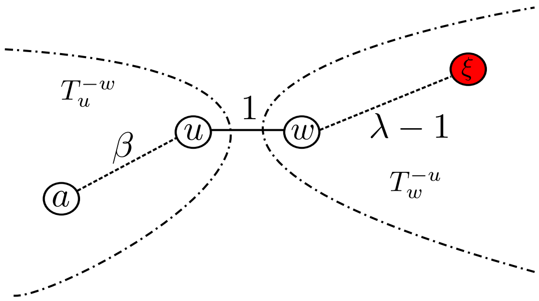


Figure A.1: A pictorial description of the positions of nodes a, u, w and ξ.

Next, we present the essential lemma which is needed to prove Theorem 2.

Lemma 11. *For any $a \in \mathcal{V}(\tilde{T})$ which is not a Jordan infection center, there exists a path from a to one Jordan infection center along which the infection eccentricity strictly decreases.*

Proof. We assume the tree has two Jordan infection centers: w and u, and assume $e(w, \mathcal{I}) = e(u, \mathcal{I}) = \lambda$. The same argument works for the case where the tree has only one Jordan infection center.

According to (Harary, 1991), there exist at most two Jordan centers. When the network has two Jordan centers, the two must be neighbors. Therefore w and u must be adjacent. We will show for any $a \in \mathcal{V}(\tilde{T}) \backslash \{w, u\}$, there exists a path from a to u (or w) along which the infection eccentricity strictly decreases.

Step 1: First, it is easy to see from Figure A.1 that $d_{\gamma w} \leq \lambda - 1 \ \forall \gamma \in \tilde{T}_w^{-u} \cap \mathcal{I}$. We next show that there exists a node ξ such that the equality holds.

Suppose that $d_{\gamma w} \leq \lambda - 2$ for any $\gamma \in \tilde{T}_w^{-u} \cap \mathcal{I}$, which implies

$$d_{\gamma u} \leq \lambda - 1 \quad \forall \gamma \in \tilde{T}_w^{-u} \cap \mathcal{I}.$$

Since w and u are both Jordan infection centers, we have $\forall \gamma \in \tilde{T}_u^{-w} \cap \mathcal{I}$,

$$d_{\gamma w} \leq \lambda$$
$$d_{\gamma u} \leq \lambda - 1.$$

In a summary, $\forall \gamma \in \mathcal{I}$,

$$d_{\gamma u} \leq \lambda - 1.$$

This contradicts the fact that $e(w, \mathcal{I}) = e(u, \mathcal{I}) = \lambda$. Therefore, there exists $\xi \in \tilde{T}_w^{-u} \cap \mathcal{I}$ such that

$$d_{\xi w} = \lambda - 1.$$

Step 2: Similarly, $\forall \gamma \in \tilde{T}_u^{-w} \cap \mathcal{I}$,

$$d_{\gamma u} \leq \lambda - 1,$$

and there exists a node such that the equality holds.

Step 3: Next we consider $a \in \mathcal{V}(\tilde{T})\backslash\{w,u\}$, and assume $a \in \tilde{T}_u^{-w}$ and $d(a,u)=\beta$. Then for any $\gamma \in \tilde{T}_w^{-u} \cap \mathcal{I}$, we have

$$\begin{aligned} d_{a\gamma} &= d_{au} + d_{uw} + d_{w\gamma} \\ &\leq \beta + 1 + \lambda - 1 \\ &= \lambda + \beta, \end{aligned}$$

and there exists $\xi \in \tilde{T}_w^{-u} \cap \mathcal{I}$ such that the equality holds. On the other hand, $\forall \gamma \in \tilde{T}_u^{-w} \cap \mathcal{I}$.

$$\begin{aligned} d_{a\gamma} &\leq d_{au} + d_{u\gamma} \\ &\leq \beta + \lambda - 1. \end{aligned}$$

Therefore, we conclude that

$$e(a,\mathcal{I}) = \lambda + \beta,$$

so the infection eccentricity decreases along the path from a to u.

□

Based on Lemma 11, there exists a path from any node to a Jordan infection center in the tree network such that the infection eccentricity strictly decreases along the path. By repeatedly applying Lemma 10, we conclude that a MAP estimator must be a Jordan infection center.

Recall $\mathcal{J}$ is the set of Jordan infection centers. Next, we will show that the MAP estimator , say node v, has the maximum $\sum_{(v,w)\in\mathcal{F}'_u} |\log(1-q_{vw})|$ among all nodes in $\mathcal{J}$.

Define an edge set

$$\mathcal{F} = \{(v,w)|(v,w) \in \mathcal{E}(\tilde{T}), v \in \mathcal{I}, w \in \mathcal{H}\}.$$

We call the edges in $\mathcal{F}$ the frontier edges since they are the edges between $\mathcal{I}$ and $\mathcal{H}$.

Define another edge set

$$\mathcal{B} = \{(v,w)|(v,w) \in \mathcal{E}(\tilde{T}), v, w \in \mathcal{I}\}.$$

The edges in $\mathcal{B}$ are the edges between infected nodes.

In addition, for any $u \in \mathcal{J}$ define

$$\mathcal{F}_u(t_u) = \{(v,w)|(v,w) \in \mathcal{E}(\tilde{T}), v \in \mathcal{I}, w \in \mathcal{H}, d_{uw} \leq t_u\}.$$

$\mathcal{F}_u(t_u)$ is set of edges which cannot be live edges when u is the source and t_u is the observation time.

For a complete observation, we have

$$\Pr(\mathcal{O}|u,t_u) = \prod_{(v,w)\in\mathcal{B}} q_{vw} \prod_{(v,w)\in\mathcal{F}_u(t_u)} (1-q_{vw}) \tag{A.2}$$

Denote by e^* the minimum infection eccentricity, i.e.,

$$\forall v \in \mathcal{J}, e(v, \mathcal{I}) = e^*$$

Intuitively, when $t_u > e^*$, none of frontier edges should be a live edge in a feasible live edge graph to make sure healthy nodes are not infected. So when $t_u > e^*$, we have

$$\mathcal{F}_u(t_u) = \mathcal{F}.$$

Hence,

$$\Pr(\mathcal{O}|u, t_u) = \prod_{(v,w)\in\mathcal{B}} q_{vw} \prod_{(v,w)\in\mathcal{F}} (1 - q_{vw}) \triangleq C,$$

which is not a function of either t_u or u. Substituting into Equation (A.2), we have

$$\Pr(\mathcal{O}|u, e^*) = \frac{C}{\prod_{(v,w)\in\mathcal{F}\backslash\mathcal{F}_u(e^*)}(1 - q_{vw})} \tag{A.3}$$

Follow a similar procedure in Lemma 10, for an Jordan infection center u, we have

$$\begin{aligned}\Pr(u|\mathcal{O}) &= \frac{\Pr(u)}{\Pr(\mathcal{O})} \sum_t \Pr(\mathcal{O}|u, t) \Pr(t) \\ &= \frac{\Pr(u)}{\Pr(\mathcal{O})} \Big(\Pr(\mathcal{O}|u, e^*) \Pr(t = e^*) + \sum_{t>e^*} \Pr(\mathcal{O}|u, t) \Pr(t) \Big) \\ &= \frac{\Pr(u)}{\Pr(\mathcal{O})} \left(\Pr(\mathcal{O}|u, e^*) \Pr(t = e^*) + \Pr(t > e^*) C \right)\end{aligned}$$

Therefore,

$$\begin{aligned} & \arg\max_u \Pr(u|\mathcal{O}) && \text{(A.4)} \\ &= \arg\max_{u\in\mathcal{J}} \Pr(u|\mathcal{O}) && \text{(A.5)} \\ &= \arg\max_{u\in\mathcal{J}} \frac{\Pr(u)}{\Pr(\mathcal{O})} \big(\Pr(\mathcal{O}|u, e^*) \Pr(t = e^*) + \Pr(t > e^*) C \big) && \text{(A.6)} \\ &= \arg\max_{u\in\mathcal{J}} \Pr(\mathcal{O}|u, e^*) && \text{(A.7)} \\ &= \arg\min_{u\in\mathcal{J}} \prod_{(v,w)\in\mathcal{F}\backslash\mathcal{F}_u(e^*)} (1 - q_{vw}). && \text{(A.8)}\end{aligned}$$

Note we have

$$\mathcal{F}\backslash\mathcal{F}_u(e^*) = \{(v, w)|(v, w) \in \mathcal{E}(\tilde{T}), v \in \mathcal{I}, w \in \mathcal{H}, d_{vw} > e^*\}$$

Since e^* is the minimum eccentricity and u is the Jordan infection center, we have $d_{uv} \leq e^*$ for all $v \in \mathcal{I}$. Hence for all $w \in \mathcal{H}$ which have at least one edge to the infected nodes, we have $d_{uw} \leq e^* + 1$. Therefore, we have

$$\mathcal{F}\backslash\mathcal{F}_u(e^*) = \{(v, w)|(v, w) \in \mathcal{E}(\tilde{T}), v \in \mathcal{I}, w \in \mathcal{H}, d_{vw} = e^* + 1\} = \mathcal{F}'_u.$$

Based on equations A.7 and A.8, we conclude

$$\arg\min_{u\in\mathcal{J}} \prod_{(v,w)\in\mathcal{F}'_u} (1-q_{vw}) = \arg\max_{u\in\mathcal{J}} \sum_{(v,w)\in\mathcal{F}'_u} |\log(1-q_{vw})| = \arg\max_u \Pr(u|\mathcal{O}).$$

Remark: Theorem 2 contains two important properties for the MAP estimator on tree networks: 1) the MAP estimator is a Jordan infection center; 2) the Jordan infection center with minimum $\prod_{(v,w)\in\mathcal{F}'_u}(1-q_{vw})$ is the MAP estimator. The short-fat tree algorithm is designed based on these properties, which identifies the Jordan infection centers first and then selects the one with maximum $\sum_{(v,w)\in\mathcal{F}'_u}|\log(1-q_{vw})|$.

A.3 Proof of Theorem 3

We first introduce and recall some necessary notations. Consider an ER random graph g.

- Denote by s the actual source.
- A node v is said to locate on level k if $d_{sv}=k$. Denote by $\mathcal{L}_k$ the set of nodes from level 0 to level k and $l_k=|\mathcal{L}_k|$.
- The *descendants* of node v in a tree are all the nodes in the subtree rooted at v and v is the *ancestor* of these nodes.
- The offsprings of a node on level k (say v) are the nodes which are on level $k+1$ and have edges to v. Denote by $\Phi(v)$ the offspring set of v and $\phi(v)=|\Phi(v)|$.
- Denote by p the wiring probability in the ER random graph.
- Denote by n the total number of nodes.
- Denote by $\mu=np$.
- Recall that $\mathrm{Bi}(n,p)$ is the binomial distribution with n number of trials and each trial succeeds with probability p.
- Denote by q the minimum infection probability of all the edges, i.e., $q=\min_{e\in\mathcal{E}(g)} q_e$.

For simplicity, we use $d_{vu}=d^g_{vu}$.

We first elaborate the construction of the BFS tree. Denote by v_{ij} the jth saturated node on level i of the BFS tree from the source. $v_{01}=s$ is the first node on level zero. Denote by b_i the number of nodes on level i of the BFS tree starting from the source. We start with an empty graph $T^\dagger$. Initially, we add v_{01} to the tree. Starting from v_{01}, we explore all neighbors $v_{11},v_{12},\cdots,v_{1b_1}$ of v_{01}, mark v_{01} as saturated and add the edges from v_{01} to $v_{11},v_{12},\cdots,v_{1b_1}$ to $T^\dagger$. Then we explore all neighbors $v_{21},v_{22},\cdots,v_{2r_1}$ of v_{11} in the set $\mathcal{V}(g)\backslash\{v_{01},v_{11},\cdots,v_{1b_1}\}$, mark v_{11} as saturated and add the edges from v_{11} to $v_{21},v_{22},\cdots,v_{2r_1}$ to $T^\dagger$. Then we explore all neighbors $v_{2r_1+1},v_{2r_1+2},\cdots,v_{2r_1+r_2}$ of v_{12} in the set $\mathcal{V}(g)\backslash\{v_{01},v_{11},\cdots,v_{1b_1},v_{21},v_{22},\cdots,v_{2r_1}\}$ and add the corresponding edges. Only after all nodes on level i are saturated, we explore nodes on level $i+1$. The exploration terminates after all nodes on level $t-1$ are saturated. The resulting tree $T^\dagger$ is the BFS tree.

We further introduce some notations for the BFS tree.

- Denote by $\Phi'(v)$ the set of offsprings of node v on $T^\dagger$ and $\phi'(v) = |\Phi'(v)|$.
- Denote by g_t the subgraph induced by all nodes within t hops from s on the ER graph. The *collision edges* are the edges which are not in $T^\dagger$ but in g_t, i.e., $e \in \mathcal{E}(g_t) \backslash \mathcal{E}(T^\dagger)$. A node who is an end node of a collision edge is called a *collision node.* Denote by $\mathcal{R}_k$ the set of collision edges whose end nodes are within level k and $R_k = |\mathcal{R}_k|$.
- Denote by $\mathcal{Z}_j^i(v)$ the set of nodes that are infected at time slot i, on level j and the descendants of node v in the BFS tree $T^\dagger$. In addition, denote by $Z_j^i(v) = |\mathcal{Z}_j^i(v)|$. We often use $\mathcal{Z}_j^i = \mathcal{Z}_j^i(s)$ and $Z_j^i = Z_j^i(s)$ for simplicity.

We first define the probability space of the problem. Define the sample space Ω to be the set of live edge subgraphs of all ER graphs. The probability measure of a live-edge graph is defined by edge generations. Edge (v, w) exists in a live edge subgraph with probability pq_{vw}.

To prove that s is the *only* Jordan infection center, we consider the following asymptotically high probability events.

- **Offsprings of each node.** Define

  $$E_1 = \{\forall v \in \mathcal{L}_{t-1}, \phi'(v) \in ((1-\delta)\mu, (1+\delta)\mu)\}.$$

  E_1, when occurs, provides upper and lower bounds for the number of offsprings of each node in $\mathcal{L}_{t-1}$.

- **Collision edges.** We define event E_2 when the following upper bound on the collision edges holds

  $$R_j \begin{cases} = 0 & \text{if } 0 < j \leq \lfloor m^- \rfloor, \\ \leq 8\mu & \text{if } \lfloor m^- \rfloor < j < \lceil m^+ \rceil, \\ \leq \frac{4[(1+\delta)\mu]^{2j+1}}{n} & \text{if } \lceil m^+ \rceil \leq j \leq \frac{\log n}{(1+\alpha)\log \mu}. \end{cases}$$

  where $m^+ = \frac{\log n}{2\log[(1+\delta)\mu]}$ and $m^- = \frac{\log n - 2\log \mu - \log 8}{2\log[(1+\delta)\mu]}$. E_2 provides the upper bounds for collision edges at different levels. Note that a subgraph with diameter $\leq m^-$ is a tree with high probability since there is no collision edge.

- **Infected nodes.** Define

  $$E_3 = \{Z_1^1 \geq (1-\delta)^2\mu q\} \cap \{\forall v \in \mathcal{Z}_1^1, \cap_{i=2}^t Z_i^i(v) \geq (1-\delta)^2 \mu q Z_{i-1}^{i-1}(v)\}.$$

  Level 1 has at least $(1-\delta)^2\mu q$ infected nodes and the number of nodes grows exponentially by each level with a factor of $(1-\delta)^2\mu q$. One immediate consequence of event E_3 is that

  $$\forall v \in \mathcal{Z}_1^1, Z_t^t(v) \geq [(1-\delta)^2\mu q]^{t-1},$$

  i.e., there are at least $[(1-\delta)^2\mu q]^{t-1}$ infected descendants on level t in $T^\dagger$ for each infected node on level 1.

Based on Lemma 12, 13 and 14, for any $\epsilon > 0$, with the union bound, we have that when $t \leq \frac{\log n}{(1+\alpha)\log \mu}$ and n is sufficiently large,

$$\Pr(E_1 \cap E_2 \cap E_3) \geq \Pr(E_1)\left(1 - \Pr(\bar{E}_2|E_1) - \Pr(\bar{E}_3|E_1)\right) \geq 1 - \epsilon$$

Next, we show that s is the only Jordan infection center when E_1, E_2, E_3 occur.

For $t \leq \lfloor m^- \rfloor$, the nodes within t hops from the source form a tree because there is no collision edge (due to event E_2). When event E_3 occurs, we have $\forall v \in \mathcal{Z}_1^1, Z_t^t(v) \geq [(1-\delta)^2\mu q]^{t-1}$ which means there exists at least one observed infected node on t level for each subtree rooted on level 1. Consider infected node s'. Recall that $a(s')$ is the ancestor of s' on level 1 of $T^\dagger$. Consider node $u \in \mathcal{Z}_1^1$ such that $u \neq a(s')$ and node $w \in \mathcal{Z}_t^t(u)$. We have $d_{s'w}^{T^\dagger} = d_{s's}^{T^\dagger} + d_{sw}^{T^\dagger} > t$. Hence the infection eccentricity of s' is larger than t. Therefore, s is the only Jordan infection center. The positions of $s, s', a(s'), u$ and w are illustrated in Figure 2.3.

Consider the case $t > \lfloor m^- \rfloor$ and an infected node s' on level $k \in [1, t]$. In the rest of the proof, we show that there exists node $v \in \mathcal{I}$ such that $d_{s'v} > t$, which means that s' cannot be the Jordan infection center.

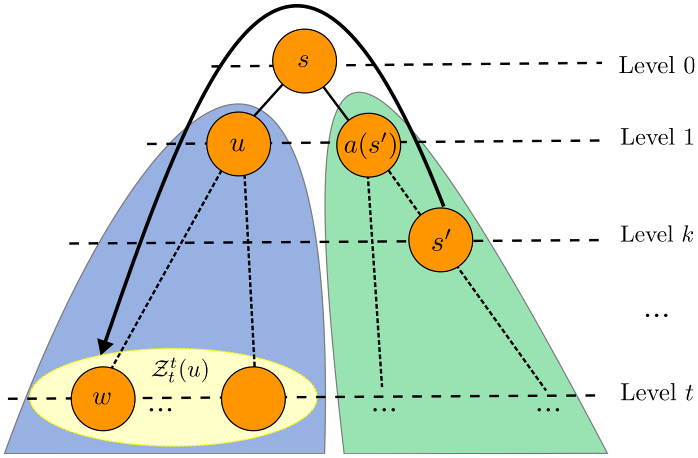


Figure A.2: A pictorial example of $\mathcal{Z}_t^t(u)$ in BFS tree $T^\dagger$

Consider node $u \in \mathcal{Z}_1^1, u \neq a(s')$ (the existence of u is guaranteed since $Z_1^1 \geq (1-\delta)\mu q \geq 2$). For the convenience of the reader, we copied Figure 2.3 as Figure A.2 which shows the relative positions of $s', a(s'), u$, and $\mathcal{Z}_t^t(u)$. The distance between a node in $\mathcal{Z}_t^t(u)$ and s' on the tree $T^\dagger$ is $k+t$. Therefore, if s' is the Jordan infection center, there exists at least one collision node on the path between s' and each node in $\mathcal{Z}_t^t(u)$ to make the distance $\leq t$.

Define H to be the total number of nodes each of which has the shortest path to s' within t hops and containing at least one collision node on g_t. If $H < Z_t^t(u)$, there exists a node $v \in \mathcal{Z}_t^t(u)$ such that $d_{s'v} > t$. Therefore, s' can not be the Jordan infection center and the theorem is proved.

In the rest of the proof, we will show that $H < Z_t^t(u)$. We first have the lower bound on $Z_t^t(u)$ according to E_3,

$$Z_t^t(u) \geq [(1-\delta)^2\mu q]^{t-1} \tag{A.9}$$

The upper bound of H is computed in Lemma 15.

$$H \leq c[(1+\delta)\mu]^{\frac{3}{4}t+\frac{1}{2}} + c[(1+\delta)\mu]^{(\frac{5}{4}-\frac{\alpha}{2})t+2},$$

Since $\frac{1}{2} < \alpha < 1$, we have $\alpha = \frac{1}{2} + \alpha'$ where $0 < \alpha' < \frac{1}{2}$ is a constant. Based on Lemma 15, we have

$$\begin{aligned}
\frac{H}{Z_t^t(u)} &\leq \frac{c[(1+\delta)\mu]^{\frac{3}{4}t+\frac{1}{2}} + c[(1+\delta)\mu]^{(\frac{5}{4}-\frac{\alpha}{2})t+2}}{[(1-\delta)^2\mu q]^{t-1}} \\
&\leq \frac{c[(1+\delta)\mu]^{\frac{3}{4}t+\frac{1}{2}}}{[(1-\delta)^2\mu q]^{t-1}} + \frac{c[(1+\delta)\mu]^{(\frac{5}{4}-\frac{\alpha}{2})t+2}}{[(1-\delta)^2\mu q]^{t-1}} \\
&= \frac{c}{\mu}\left(\frac{(1+\delta)^{\frac{3}{4}+\frac{1}{2t}}}{[(1-\delta)^2 q]^{1-\frac{1}{t}}\mu^{\frac{1}{4}-\frac{5}{2t}}}\right)^t + \frac{c}{\mu}\left(\frac{(1+\delta)^{\frac{5}{4}-\frac{\alpha}{2}+\frac{2}{t}}}{[(1-\delta)^2 q]^{1-\frac{1}{t}}\mu^{\frac{\alpha}{2}-\frac{1}{4}-\frac{4}{t}}}\right)^t \\
&\leq \frac{c}{\mu}\left(\frac{(1+\delta)^{\frac{3}{4}+\frac{1}{2t}}}{[(1-\delta)^2 q]^{1-\frac{1}{t}}\mu^{\frac{1}{4}-\frac{4}{t}}}\right)^t + \frac{c}{\mu}\left(\frac{(1+\delta)^{1-\frac{\alpha'}{2}+\frac{2}{t}}}{[(1-\delta)^2 q]^{1-\frac{1}{t}}\mu^{\frac{\alpha'}{2}-\frac{4}{t}}}\right)^t
\end{aligned}$$

For $t > 16/\alpha'$ we have

$$\frac{H}{Z_t^t(u)} \leq \frac{2c}{\mu}\left(\frac{(1+\delta)}{[(1-\delta)^2 q]\mu^{\frac{\alpha'}{4}}}\right)^t.$$

Since $\mu > 3\log n$ and δ, q, α' are constants, we have

$$\frac{(1+\delta)}{[(1-\delta)^2 q]\mu^{\frac{\alpha'}{4}}} < 1$$

when

$$n > \exp\left(\frac{1}{2}\left(\frac{(1+\delta)}{(1-\delta)^2 q}\right)^{\frac{4}{\alpha'}}\right).$$

Therefore, we have

$$\frac{H}{Z_t^t(u)} \leq \frac{2c}{\mu} \leq \epsilon',$$

where $\epsilon' \in (0,1)$ is a constant and the inequality holds for sufficiently large n. Therefore, there are at least $(1-\epsilon')Z_t^t(u)$ nodes which cannot be reached from s' on level k with time t. Hence we have $e(s', \mathcal{O}) > t, \forall s' \neq s$.

A.3.1 Bounds on the Number of Offsprings of Each Node

Lemma 12. *Assume the conditions in Theorem 3 hold, for any* $\epsilon > 0$, *we have*

$$\Pr(E_1) \geq 1 - \epsilon$$

for sufficient large n.

Proof. Consider $\delta \in (0,1)$. Since $t \leq \frac{\log n}{(1+\alpha)\log \mu}$, we have for sufficiently large n,

$$\sum_{i=0}^{t}[(1+\delta)\mu]^i \leq 2[(1+\delta)\mu]^t \leq \delta' n,$$

where $\delta' \in (0,1)$ is a constant which can be arbitrarily close to 0. This condition shows that the t hop neighborhood of node s includes at most a constant fraction of the total number of nodes.

Denote by $\mathcal{E}(\mathcal{V}_1, \mathcal{V}_2)$ the set of edges between node set $\mathcal{V}_1$ and $\mathcal{V}_2$. Recall that v_{ij} is the jth nodes on level i to be explored in the BFS tree starting from the source and b_i is the number of nodes on level i.

Define the edge set from v_{01} to all other nodes in the ER graph to be

$$\Psi(v_{01}) = \mathcal{E}(\{v_{01}\}, \mathcal{V}(g)\backslash\{v_{01}\}),$$

which is the set of edges between v_{01} and all other nodes in the graph.

Define

$$\Phi'(v_{01}) = \{v|(v, v_{01}) \in \Psi(v_{01})\}.$$

Define

$$\Psi(v_{01}, v_{11}) = \mathcal{E}(\{v_{11}\}, \mathcal{V}(g)\backslash(\Phi'(v_{01}) \cup \{v_{01}\})),$$

which is the set of edges from node v_{11} to all nodes that are not already included in the BFS tree and

$$\Phi'(v_{01}, v_{11}) = \{v|(v, v_{11}) \in \Psi(v_{01}, v_{11})\},$$

which is the set of offsprings of v_{11}.

For simplicity, we use $\Psi(v_{ij})$ to denote

$$\Psi(v_{01}, v_{11}, \cdots, v_{1b_1}, \cdots, v_{i1}, \cdots, v_{ij}).$$

and use $\Phi'(v_{ij})$ to denote

$$\Phi'(v_{01}, v_{11}, \cdots, v_{1b_1}, \cdots, v_{i1}, \cdots, v_{ij}).$$

Iteratively, we define

$$\Psi(v_{ij}) \triangleq \mathcal{E}(\{v_{ij}\}, \mathcal{V}(g)\backslash(\{v_{01}\} \cup \Phi'(v_{01}) \cup \cdots \cup \Phi'(v_{ij-1})))$$

and

$$\Phi'(v_{ij}) \triangleq \{v|(v, v_{ij}) \in \Psi(v_{ij})\}$$

which is the set of offsprings of node v_{ij} in the BFS tree from the source. Define $\phi'(v_{ij}) = |\Phi'(v_{ij})|$ and $\psi(v_{ij}) = |\Psi(v_{ij})|$.

Note that $\Psi(v_{ij})$ uniquely determines $\Phi'(v_{ij})$ and vice versa. In addition, according to the definition, $\Psi(v_{ij})$ for any i and j are pairwise disjoint.

Define

$$\Lambda(v_{ij}) = \{\Phi'(v_{ij})|\phi'(v_{ij}) \in (\mu(1-\delta), \mu(1+\delta))\}.$$

which is the set of $\Phi'(v_{ij})$ which satisfies the given bounds on the number of offsprings.

Therefore, we have

$$\begin{aligned}
&\Pr(E_1)\\
&= \Pr\left(\forall v \in \mathcal{L}_{t-1}, \phi'(v) \in (\mu(1-\delta), \mu(1+\delta))\right)\\
&= \sum_{\Phi'(v_{01}) \in \Lambda(v_{01})} \Pr\Big(\Phi'(v_{01}), \phi'(v) \in (\mu(1-\delta), \mu(1+\delta)), \forall v \in \mathcal{L}_{t-1}\backslash\{v_{01}\}\Big)\\
&= \sum_{\Phi'(v_{01}) \in \Lambda(v_{01})} \Pr(\Phi'(v_{01})) \Pr\big(\forall v \in \mathcal{L}_{t-1}\backslash\{v_{01}\}, \phi'(v) \in (\mu(1-\delta), \mu(1+\delta)) | \Phi'(v_{01})\big)
\end{aligned}$$

Given $\Phi'(v_{01})$, the order of the nodes to be explored during the construction of the BFS tree on the next level is determined and we have

$$\begin{aligned}
&\Pr\left(\forall v \in \mathcal{L}_{t-1}, \phi'(v) \in (\mu(1-\delta), \mu(1+\delta))\right)\\
&= \sum_{\Phi'(v_{01}) \in \Lambda(v_{01})} \sum_{\Phi'(v_{11}) \in \Lambda(v_{11})} \Pr(\Phi'(v_{01}), \Phi'(v_{11}))\\
&\times \Pr\big(\forall v \in \mathcal{L}_{t-1}\backslash\{v_{01}, v_{11}\}, \phi'(v) \in (\mu(1-\delta), \mu(1+\delta)) | \Phi'(v_{01}), \Phi'(v_{11})\big)
\end{aligned}$$

Iteratively, we have

$$\begin{aligned}
&\Pr\left(\forall v \in \mathcal{L}_{t-1}, \phi'(v) \in (\mu(1-\delta), \mu(1+\delta))\right) && \text{(A.10)}\\
&= \sum_{\Phi'(v_{01}) \in \Lambda(v_{01})} \cdots \sum_{\Phi'(v_{t-1 b_{t-1}-1}) \in \Lambda(v_{t-1 b_{t-1}-1})} && \text{(A.11)}\\
&\Pr(\Phi'(v_{01}), \cdots, \Phi'(v_{t-1 b_{t-1}-1})) && \text{(A.12)}\\
&\times \Pr\Big(\phi'(v_{t-1 b_{t-1}}) \in (\mu(1-\delta), \mu(1+\delta)) && \text{(A.13)}\\
&|\Phi'(v_{01}), \cdots, \Phi'(v_{t-1 b_{t-1}-1})\Big) && \text{(A.14)}
\end{aligned}$$

Next, we focus on the last term in Equation (A.14). Note, $\Psi(v_{ij})$ uniquely determines $\Phi'(v_{ij})$ and vice versa. Therefore,

$$\begin{aligned}
&\Pr\Big(\phi'(v_{t-1 b_{t-1}}) \in (\mu(1-\delta), \mu(1+\delta)) | \Phi'(v_{01}), \cdots, \Phi'(v_{t-1 b_{t-1}-1})\Big) && \text{(A.15)}\\
&= \Pr\Big(\phi'(v_{t-1 b_{t-1}}) \in (\mu(1-\delta), \mu(1+\delta)) | \Psi(v_{01}), \cdots, \Psi(v_{t-1 b_{t-1}-1})\Big) && \text{(A.16)}
\end{aligned}$$

Since $\Psi(v_{t-1 b_{t-1}})$ is disjoint with $\Psi(v_{01}), \cdots, \Psi(v_{t-1 b_{t-1}-1})$ and each edge is generated independently in the ER graph. Therefore, conditioned on $\Psi(v_{01}), \cdots, \Psi(v_{t-1 b_{t-1}-1})$, we have $\phi'(v_{t-1 b_{t-1}})$ follows

$$\mathrm{Bi}(n - \sum_{i=0}^{t-2} \sum_{j=1}^{b_i} \phi'(v_{ij}) - \sum_{j=1}^{b_{t-1}-1} \phi'(v_{t-1j}) - 1, p).$$

Note, $\phi(v_{01}), \cdots, \phi(v_{t-1b_{t-1}-1})$ are in $(\mu(1-\delta), \mu(1+\delta))$ according to the condition in Equation (A.11). Hence

$$\sum_{i=0}^{t-2}\sum_{j=1}^{b_i}\phi'(v_{ij}) + \sum_{j=1}^{b_{t-1}-1}\phi'(v_{t-1j}) + 1 \leq \sum_{i=0}^{t}[\mu(1+\delta))]^i$$

Therefore, $\phi'(v_{t-1b_{t-1}})$ stochastically dominates $\mathrm{Bi}(n - \sum_{i=0}^{t}[\mu(1+\delta))]^i, p)$ and is stochastically dominated by $\mathrm{Bi}(n,p)$ which implies

$$\begin{aligned}&\Pr\left(\phi'(v_{t-1b_{t-1}}) \in (\mu(1-\delta), \mu(1+\delta))|\Phi'(v_{01}), \cdots, \Phi'(v_{t-1b_{t-1}-1})\right)\\ &\geq 1 - \Pr\left(\mathrm{Bi}\left(n - \sum_{i=0}^{t}[\mu(1+\delta))]^i, p\right) \leq (1-\delta)\mu\right) - \Pr\left(\mathrm{Bi}\left(n,p\right) \geq \mu(1+\delta)\right)\end{aligned}$$

Note $\sum_{i=0}^{t}[(1+\delta)\mu]^i \leq \delta' n$. Therefore, we have

$$\Pr\left(\phi'(v_{t-1b_{t-1}}) \in (\mu(1-\delta), \mu(1+\delta))|\Phi'(v_{01}), \cdots, \Phi'(v_{t-1b_{t-1}-1})\right) \quad (A.17)$$

$$\geq 1 - \Pr\left(\mathrm{Bi}\left((1-\delta')n, p\right) \leq (1-\delta)\mu\right) - \Pr\left(\mathrm{Bi}\left(n,p\right) \geq \mu(1+\delta)\right) \quad (A.18)$$

By using the Chernoff bound in Lemma 19, we have

$$\Pr\left(\mathrm{Bi}\left((1-\delta')n, p\right) \leq \mu(1-\delta)\right) \leq \exp\left(-\frac{(\delta-\delta')^2\mu}{2(1-\delta')}\right),$$

and

$$\Pr\left(\mathrm{Bi}\left(n,p\right) \geq \mu(1+\delta)\right) \leq \exp\left(-\frac{\delta^2\mu}{2+\delta}\right)$$

Substitute into Inequality (A.18), we obtain

$$\Pr\left(\phi'(v_{t-1b_{t-1}}) \in (\mu(1-\delta), \mu(1+\delta))|\Phi'(v_{01}), \cdots, \Phi'(v_{t-1b_{t-1}-1})\right) \quad (A.19)$$

$$\geq 1 - \exp\left(-\frac{(\delta-\delta')^2\mu}{2(1-\delta')}\right) - \exp\left(-\frac{\delta^2\mu}{2+\delta}\right) \triangleq \Delta \quad (A.20)$$

Substitute Inequality (A.20) into Equation (A.14), we obtain

$$\Pr\left(\forall v \in \mathcal{L}_{t-1}, \phi'(v) \in (\mu(1-\delta), \mu(1+\delta))\right) \tag{A.21}$$

$$\geq \sum_{\Phi'(v_{01})\in\Lambda(v_{01})} \cdots \sum_{\Phi'(v_{t-1b_{t-1}-1})\in\Lambda(v_{t-1b_{t-1}-1})} \tag{A.22}$$

$$\Pr(\Phi'(v_{01}), \cdots, \Phi'(v_{t-1b_{t-1}-1})) \times \Delta \tag{A.23}$$

$$= \Delta \sum_{\Phi'(v_{01})\in\Lambda(v_{01})} \cdots \sum_{\Phi'(v_{t-1b_{t-1}-2})\in\Lambda(v_{t-1b_{t-1}-2})} \tag{A.24}$$

$$\Bigg(\sum_{\Phi'(v_{t-1b_{t-1}-1})\in\Lambda(v_{t-1b_{t-1}-1})} \Pr(\Phi'(v_{01}), \cdots, \Phi'(v_{t-1b_{t-1}-1}))\Bigg) \tag{A.25}$$

$$= \Delta \sum_{\Phi'(v_{01})\in\Lambda(v_{01})} \cdots \sum_{\Phi'(v_{t-1b_{t-1}-2})\in\Lambda(v_{t-1b_{t-1}-2})} \tag{A.26}$$

$$\Pr(\Phi'(v_{01}), \cdots, \Phi'(v_{t-1b_{t-1}-2})) \tag{A.27}$$

$$\Pr\big(\phi'(v_{t-1b_{t-1}-1}) \in (\mu(1-\delta), \mu(1+\delta)) \tag{A.28}$$

$$|\Phi'(v_{01}), \cdots, \Phi'(v_{t-1b_{t-1}-2})\big) \tag{A.29}$$

$$= \Delta^2 \sum_{\Phi'(v_{01})\in\Lambda(v_{01})} \cdots \sum_{\Phi'(v_{t-1b_{t-1}-2})\in\Lambda(v_{t-1b_{t-1}-2})} \Pr(\Phi'(v_{01}), \cdots, \Phi'(v_{t-1b_{t-1}-2})) \tag{A.30}$$

Applying Equation (A.30) iteratively, we have

$$\begin{aligned}&\Pr\left(\forall v \in \mathcal{L}_{t-1}, \phi'(v) \in (\mu(1-\delta), \mu(1+\delta))\right) \\ \geq& \Delta^{\sum_{i=0}^{t-1}[(1+\delta)\mu]^i} \\ \geq& \left(1 - \exp\left(-\frac{(\delta-\delta')^2\mu}{2(1-\delta')}\right) - \exp\left(-\frac{\delta^2\mu}{2+\delta}\right)\right)^{\sum_{i=0}^{t-1}[(1+\delta)\mu]^i}\end{aligned}$$

When $\delta' \to 0$, we have

$$\frac{(\delta-\delta')^2\mu}{2(1-\delta')} \to \frac{\delta^2\mu}{2} > \frac{\delta^2\mu}{2+\delta}$$

Therefore, we can choose a sufficiently small δ' such that

$$\begin{aligned}
&\Pr\left(\forall v \in \mathcal{L}_{t-1}, \phi'(v) \in (\mu(1-\delta), \mu(1+\delta))\right) \\
\geq &\left(1 - 2\exp\left(-\frac{\delta^2\mu}{2+\delta}\right)\right)^{\sum_{i=0}^{t-1}[(1+\delta)\mu]^i} \\
\geq &\left(1 - 2\exp\left(-\frac{\delta^2\mu}{2+\delta}\right)\right)^{2[(1+\delta)\mu]^{t-1}} \\
\geq_{(a)} &\exp\left(-8[(1+\delta)\mu]^{t-1}\exp\left(-\frac{\delta^2\mu}{2+\delta}\right)\right) \\
\geq &\exp\left(-8\exp\left(-\frac{\delta^2\mu}{2+\delta} + (t-1)\log[(1+\delta)\mu]\right)\right),
\end{aligned}$$

where (a) is based on Lemma 20 and holds when μ is sufficiently large (i.e., when n is sufficiently large). To make the above bound greater than $1-\epsilon$, we need

$$t \leq \frac{\frac{\delta^2\mu}{2+\delta} - \log 8 + \log\log\left(\frac{1}{1-\epsilon}\right)}{\log(1+\delta) + \log\mu} + 1.$$

When $\mu > \frac{2+\delta}{\delta^2}\log n$, we have

$$\begin{aligned}
t &\leq \frac{\log n}{(1+\alpha)\log\mu} \\
&< \frac{\log n - \log 8 + \log\log\left(\frac{1}{1-\epsilon}\right)}{\log(1+\delta) + \log\mu} + 1.
\end{aligned}$$

for sufficiently large n.

Note $\frac{2+\delta}{\delta^2} \to 3$ when $\delta \to 1$ which matches the condition that $\mu > 3\log n$. Therefore, we prove the lemma.

□

A.3.2 Bounds on the Number of Collision Edges

Next, we analyze the number of collision edges on different levels. We have the following lemma.

Lemma 13. *If the conditions in Theorem 3 hold, for any $\epsilon > 0$,*

$$\Pr(E_2|E_1) \geq 1 - \epsilon$$

for sufficiently large n.

Proof. We have

$$
\begin{aligned}
\Pr(E_2|E_1) \geq 1 - &\Pr(R_{\lfloor m^- \rfloor} \neq 0|E_1) \\
&- \sum_{j=\lfloor m^- \rfloor+1}^{\lceil m^+ \rceil-1} \Pr(R_j > 8\mu|E_1) \\
&- \sum_{j=\lceil m^+ \rceil}^{t} \Pr\left(R_j > \frac{4[(1+\delta)\mu]^{2j+1}}{n}|E_1\right)
\end{aligned}
$$

- **No collision edge at the first $\lfloor m^- \rfloor$ levels.**

  We will show that

  $$\Pr\left(R_{\lfloor m^- \rfloor} \neq 0|E_1\right) \leq 1 - \exp\left(-\frac{1}{\mu}\right) \leq \frac{1}{\mu}$$

  when n is sufficiently large.

  Conditioning on E_1, we have R_j is stochastically dominated by $\mathrm{Bi}(l_j^2, p)$. Since $l_j \leq 2[(1+\delta)\mu]^j$, R_j is stochastically dominated by $\mathrm{Bi}(4[(1+\delta)\mu]^{2j}, p)$. We have for sufficiently large n,

  $$
  \begin{aligned}
  \Pr\left(R_j = 0\middle|E_1\right) &\geq (1-p)^{\left(2[(1+\delta)\mu]^j\right)^2} \\
  &= \left(1 - \frac{\mu}{n}\right)^{4[(1+\delta)\mu]^{2j}} \\
  &\geq_{(a)} \exp\left(-8[(1+\delta)\mu]^{2j}\frac{\mu}{n}\right) \\
  &\geq_{(b)} \exp\left(-\frac{1}{\mu}\right)
  \end{aligned}
  $$

  Inequality (a) is based on Lemma 20. To obtain Inequality (b), note

  $$j \leq m^- = \frac{\log n - 2\log\mu - \log 8}{2\log[(1+\delta)\mu]}.$$

  We have

  $$8[(1+\delta)\mu]^{2j}\frac{\mu}{n} \leq \frac{1}{\mu}$$

  which explains (b).

- **The number of collision edges at levels between $\lfloor m^- \rfloor + 1$ and $\lceil m^+ \rceil - 1$.**
  We will show that

  $$\Pr\left(R_j > 8\mu|E_1\right) \leq \exp\left(-\frac{4}{3}\mu\right)$$

  when n is sufficiently large.

Conditioned on event E_1, R_j is stochastically dominated by $\mathrm{Bi}(l_j^2, p)$. Since $l_j \leq 2[(1+\delta)\mu]^j$, R_j is stochastically dominated by $\mathrm{Bi}(4[(1+\delta)\mu]^{2j}, p)$. Then

$$\begin{aligned}
&\Pr\left(R_j \leq \frac{4[(1+\delta)\mu]^{2j+1}}{n} | E_1\right) \\
&\geq \Pr\left(\mathrm{Bi}(4[(1+\delta)\mu]^{2j}, p) \leq \frac{4[(1+\delta)\mu]^{2j+1}}{n}\right) \\
&\geq \Pr\left(\mathrm{Bi}(4[(1+\delta)\mu]^{2j}, p) \leq (1+\delta)4[(1+\delta)\mu]^{2j}p\right) \\
&\geq 1 - \exp\left(-\frac{\delta^2}{2+\delta}4[(1+\delta)\mu]^{2j}\frac{\mu}{n}\right) \\
&\geq_{(a)} 1 - \exp\left(-\frac{\delta^2}{2+\delta}4\mu\right)
\end{aligned}$$

From $j \geq \lceil m^+ \rceil \geq \frac{\log n}{2\log[(1+\delta)\mu]}$, we obtain

$$n \leq [(1+\delta)\mu]^{2j}. \tag{A.31}$$

we obtain Inequality (a) by substituting Inequality (A.31).

- **The number of collision edges at levels between $\lceil m^+ \rceil$ and $\frac{\log n}{(1+\alpha)\log\mu}$.** We will show

$$\Pr\left(R_j > \frac{4[(1+\delta)\mu]^{2j+1}}{n}\middle| E_1\right) \leq \exp\left(-\frac{4\delta^2}{2+\delta}\mu\right)$$

when n is sufficiently large.

Let

$$\delta' = \frac{2n}{[(1+\delta)\mu]^{2j}} - 1$$

Since $j \leq m^+ = \frac{\log n}{2\log[(1+\delta)\mu]}$, we have $n \geq [(1+\delta)\mu]^{2j}$. Hence

$$\delta' \geq 1$$

Conditioned on event E_1, R_j is stochastically dominated by $\mathrm{Bi}(4[(1+\delta)\mu]^{2j}, p)$.

Using the Chernoff bound in Lemma 19, we have,

$$\begin{aligned}
&\Pr\left(R_j \leq (1+\delta')4[(1+\delta)\mu]^{2j}p \middle| E_1\right)\\
&\geq \Pr\left(\mathrm{Bi}(4[(1+\delta)\mu]^{2j}, p) \leq (1+\delta')4[(1+\delta)\mu]^{2j}p\right)\\
&\geq 1-\exp\left(-\frac{\delta'^2}{2+\delta'}4[(1+\delta)\mu]^{2j}\frac{\mu}{n}\right)\\
&\geq 1-\exp\left(-\frac{\delta'}{2+\delta'}4(2n-[(1+\delta)\mu]^{2j})\frac{\mu}{n}\right)\\
&\geq_{(a)} 1-\exp\left(-\frac{\delta'}{2+\delta'}4\mu\right)\\
&\geq_{(b)} 1-\exp\left(-\frac{4}{3}\mu\right)
\end{aligned}$$

Note (a) is due to $n > [(1+\delta)\mu]^{2j}$ and (b) is due to $\delta' \geq 1$. Note,

$$\begin{aligned}
&(1+\delta')4[(1+\delta)\mu]^{2j}p\\
=&\left(1+\frac{2n}{[(1+\delta)\mu]^{2j}}-1\right)4[(1+\delta)\mu]^{2j}p\\
=&\frac{2n}{[(1+\delta)\mu]^{2j}}4[(1+\delta)\mu]^{2j}\frac{\mu}{n}\\
=&8\mu
\end{aligned}$$

Since $m^+ - m^- < 2$ we have

$$\begin{aligned}
\Pr(E_2|E_1) \geq& 1-\frac{1}{\mu}-(m^+-m^-)\exp\left(-\frac{4}{3}\mu\right)-\sum_{j=\lceil m^+\rceil}^{t}\exp\left(-\frac{4\delta^2}{2+\delta}\mu\right)\\
\geq& 1-\frac{1}{\mu}-2\exp\left(-\frac{4}{3}\mu\right)-t\exp\left(-\frac{4\delta^2}{2+\delta}\mu\right)
\end{aligned}$$

Note we have $\mu \geq 3\log n$ and $t \leq \frac{\log n}{(1+\alpha)\log\mu}$, therefore, for n sufficiently large, we have

$$\Pr(E_2|E_1) \geq 1-\epsilon.$$

□

A.3.3 Bounds on the Number of Infected Nodes

Lemma 14. *Assume the conditions in Theorem 3 hold, for any* $\epsilon > 0$, *we have*

$$\Pr\left(E_3|E_1\right) \geq 1-\epsilon$$

for sufficiently large n.

Proof.

- We first show that for any $\epsilon > 0$,

  $$\Pr(Z_1^1 \geq (1-\delta)^2 \mu q | E_1) \geq 1 - \epsilon$$

  for sufficient large n. Z_1^1 is lower bounded by a binomial distribution $\mathrm{Bi}((1-\delta)\mu, q)$. Hence using Chernoff bound, we have

  $$\Pr(Z_1^1 \geq (1-\delta)(1-\delta)\mu q | E_1) \geq 1 - \exp\left(-\frac{\delta^2}{2}(1-\delta)\mu q\right)$$

  Note $\mu \to \infty$ while all other parameters are constants, for any $\epsilon > 0$, we have

  $$\Pr(Z_1^1 \geq (1-\delta)^2 \mu q | E_1) \geq 1 - \epsilon$$

- We show that for any $\epsilon > 0$,

  $$\Pr\left(\{\forall v \in \mathcal{Z}_1^1, Z_t^t(v) \geq [(1-\delta)^2 \mu q]^{t-1}\} | E_1\right) \geq 1 - \epsilon$$

  for sufficiently large n. Define

  $$E_4 = \{(1-\delta)^2 \mu q \leq Z_1^1 \leq (1+\delta)\mu\}$$

  Note when n is sufficiently large, the following holds

  $$\Pr(Z_1^1 \geq (1-\delta)^2 \mu q | E_1) \geq 1 - \frac{\epsilon}{4}.$$

  When E_1 occurs, we have $Z_1^1 \leq (1+\delta)\mu$. Therefore, we have

  $$\Pr(E_4 | E_1) \geq 1 - \frac{\epsilon}{4}$$

  Define

  $$\begin{aligned} \mathcal{S}_2^t(v) = \{&(\mathcal{Z}_2^2(v), \cdots, \mathcal{Z}_t^t(v)) | \\ &\cap_{i=2}^t Z_i^i(v) \geq (1-\tilde{\delta})(1-\delta)\mu q Z_{i-1}^{i-1}(v).\} \end{aligned}$$

  We have

  $$\Pr\left(\cap_{i=2}^t Z_i^i(v) \geq (1-\tilde{\delta})(1-\delta)\mu q Z_{i-1}^{i-1}(v) | E_4, E_1\right) \tag{A.32}$$
  $$= \Pr\left(Z_t^t(v) \geq (1-\tilde{\delta})(1-\delta)\mu q Z_{t-1}^{t-1}(v),\right. \tag{A.33}$$
  $$\left.\cap_{i=2}^{t-1} Z_i^i(v) \geq (1-\tilde{\delta})(1-\delta)\mu q Z_{i-1}^{i-1}(v) | E_4, E_1\right) \tag{A.34}$$
  $$= \sum_{\mathcal{Z}_2^2(v), \cdots, \mathcal{Z}_{t-1}^{t-1}(v) \in \mathcal{S}_2^{t-1}(v)} \Pr\left(Z_t^t(v) \geq\right. \tag{A.35}$$
  $$\left.(1-\tilde{\delta})(1-\delta)\mu q Z_{t-1}^{t-1}(v) | \mathcal{Z}_2^2(v), \cdots, \mathcal{Z}_{t-1}^{t-1}(v), E_4, E_1\right) \tag{A.36}$$
  $$\times \Pr(\mathcal{Z}_2^2(v), \cdots, \mathcal{Z}_{t-1}^{t-1}(v) | E_4, E_1) \tag{A.37}$$

Conditioned on E_1 and E_4, we have $Z_1^1 \neq 0$. For any $v \in \mathcal{Z}_1^1$, $Z_i^i(v)$ stochastically dominates $\mathrm{Bi}((1-\delta)\mu Z_{i-1}^{i-1}(v), q)$ given $\mathcal{Z}_{i-1}^{i-1}(v)$. Therefore, denote by $\tilde{\delta} \in (0,1)$, we have

$$\begin{aligned}
&\Pr\big(Z_t^t(v) \geq (1-\tilde{\delta})(1-\delta)\mu q Z_{t-1}^{t-1}(v) \\
&|\mathcal{Z}_2^2(v), \cdots, \mathcal{Z}_{t-1}^{t-1}(v), E_4, E_1\big) \\
\geq& \Pr(Z_t^t(v) \geq (1-\tilde{\delta})(1-\delta)\mu q Z_{t-1}^{t-1}(v) | \mathcal{Z}_{t-1}^{t-1}(v), E_4, E_1) \\
\geq& 1 - \exp\left(-\frac{\tilde{\delta}^2(1-\delta)\mu q Z_{t-1}^{t-1}(v)}{2+\tilde{\delta}}\right) \\
\geq& \exp\left(-2\exp\left(-\frac{\tilde{\delta}^2(1-\delta)\mu q Z_{t-1}^{t-1}(v)}{2+\tilde{\delta}}\right)\right)
\end{aligned}$$

Since we have $\mathcal{Z}_2^2(v), \cdots, \mathcal{Z}_{t-1}^{t-1}(v) \in \mathcal{S}_2^{t-1}(v)$, therefore,

$$Z_{t-1}^{t-1}(v) \geq [(1-\tilde{\delta})(1-\delta)\mu q]^{t-2}$$

Hence, we have

$$\begin{aligned}
&\Pr\big(Z_t^t(v) \geq (1-\tilde{\delta})(1-\delta)\mu q Z_{t-1}^{t-1}(v) \\
&|\mathcal{Z}_2^2(v), \cdots, \mathcal{Z}_{t-1}^{t-1}(v), E_4, E_1\big) \\
\geq& \exp\left(-2\exp\left(-\frac{\tilde{\delta}^2(1-\delta)\mu q[(1-\tilde{\delta})(1-\delta)\mu q]^{t-2}}{2+\tilde{\delta}}\right)\right) \\
\geq& \exp\left(-2\exp\left(-\frac{\tilde{\delta}^2[(1-\tilde{\delta})(1-\delta)\mu q]^{t-1}}{(2+\tilde{\delta})(1-\tilde{\delta})}\right)\right)
\end{aligned}$$

Substituting back to Equation (A.37), we obtain

$$\Pr\left(\cap_{i=2}^t Z_i^i(v) \geq (1-\tilde{\delta})(1-\delta)\mu q Z_{i-1}^{i-1}(v) | E_4, E_1\right) \tag{A.38}$$

$$\geq \sum_{\mathcal{Z}_2^2(v),\cdots,\mathcal{Z}_{t-1}^{t-1}(v) \in \mathcal{S}_2^{t-1}(v)} \exp\Big(-2\exp\big(-\frac{\tilde{\delta}^2[(1-\tilde{\delta})(1-\delta)\mu q]^{t-1}}{(2+\tilde{\delta})(1-\tilde{\delta})}\big)\Big) \tag{A.39}$$

$$\times \Pr(\mathcal{Z}_2^2(v), \cdots, \mathcal{Z}_{t-1}^{t-1}(v) | E_4, E_1) \tag{A.40}$$

$$= \exp\left(-2\exp\left(-\frac{\tilde{\delta}^2[(1-\tilde{\delta})(1-\delta)\mu q]^{t-1}}{(2+\tilde{\delta})(1-\tilde{\delta})}\right)\right) \tag{A.41}$$

$$\times \sum_{\mathcal{Z}_2^2(v),\cdots,\mathcal{Z}_{t-1}^{t-1}(v) \in \mathcal{S}_2^{t-1}(v)} \Pr(\mathcal{Z}_2^2(v), \cdots, \mathcal{Z}_{t-1}^{t-1}(v) | E_4, E_1) \tag{A.42}$$

$$= \exp\left(-2\exp\left(-\frac{\tilde{\delta}^2[(1-\tilde{\delta})(1-\delta)\mu q]^{t-1}}{(2+\tilde{\delta})(1-\tilde{\delta})}\right)\right) \tag{A.43}$$

$$\times \Pr(\cap_{i=2}^{t-1} Z_i^i(v) \geq (1-\tilde{\delta})(1-\delta)\mu q Z_{i-1}^{i-1}(v) | E_4, E_1) \tag{A.44}$$

Use Equation (A.44) iteratively on all levels, we obtain

$$\begin{aligned}
&\Pr\left(\cap_{i=2}^{t} Z_i^i(v) \geq (1-\tilde{\delta})(1-\delta)\mu q Z_{i-1}^{i-1}(v) | E_4, E_1\right) \\
\geq &\prod_{i=2}^{t} \exp\left(-2\exp\left(-\frac{\tilde{\delta}^2[(1-\tilde{\delta})(1-\delta)\mu q]^{i-1}}{(2+\tilde{\delta})(1-\tilde{\delta})}\right)\right) \\
= &\exp\left(-2\sum_{i=2}^{t} \exp\left(-\frac{\tilde{\delta}^2[(1-\tilde{\delta})(1-\delta)\mu q]^{i-1}}{(2+\tilde{\delta})(1-\tilde{\delta})}\right)\right) \\
\geq &\exp\left(-2(t-1)\exp\left(-\frac{\tilde{\delta}^2(1-\delta)\mu q}{2+\tilde{\delta}}\right)\right) \\
\geq &1-2(t-1)\exp\left(-\frac{\tilde{\delta}^2(1-\delta)\mu q}{2+\tilde{\delta}}\right)
\end{aligned}$$

Using the union bound for all $v \in \mathcal{Z}_1^1$, we have

$$\begin{aligned}
&\Pr\Bigg(\forall v \in \mathcal{Z}_1^1, \cap_{i=2}^{t} Z_i^i(v) \\
&\quad \geq (1-\tilde{\delta})(1-\delta)\mu q Z_{i-1}^{i-1}(v) | E_4, E_1\Bigg) \\
\geq &1-2(1+\delta)\mu t \exp\left(-\frac{\tilde{\delta}^2(1-\delta)\mu q}{2+\tilde{\delta}}\right)
\end{aligned}$$

Note $t \leq \frac{\log n}{(1+\alpha)\log \mu}$ and $\mu > 3\log n$. We have $t \leq \log n \leq \mu$, and

$$\begin{aligned}
&\Pr\Bigg(\forall v \in \mathcal{Z}_1^1, \cap_{i=1}^{t} Z_i^i(v) \\
&\geq (1-\tilde{\delta})(1-\delta)\mu q Z_{i-1}^{i-1}(v) | E_4, E_1\Bigg) \\
&\geq 1-2(1+\delta)\mu^2 \exp\left(-\frac{\tilde{\delta}^2(1-\delta)\mu q}{2+\tilde{\delta}}\right) \\
&\geq 1-\frac{\epsilon}{2}
\end{aligned}$$

for sufficiently large n.

Define

$$E_5 = \{\forall v \in \mathcal{Z}_1^1, \cap_{i=2}^{t} Z_i^i(v) \geq (1-\delta)^2 \mu q Z_{i-1}^{i-1}(v)\}$$

We further have,

$$\begin{aligned}\Pr\left(E_5|E_1\right) \geq & \Pr\left(E_5|E_4, E_1\right)\\ & \times \Pr(E_4|E_1)\\ \geq & (1-\frac{\epsilon}{4})(1-\frac{\epsilon}{4})\\ \geq & 1-\frac{\epsilon}{2}.\end{aligned}$$

Choosing $\tilde{\delta} = \delta$, we have

$$\begin{aligned}&\Pr\left(\forall v \in \mathcal{Z}_1^1, \cap_{i=2}^t Z_i^i(v) \geq (1-\delta)^2 \mu q Z_{i-1}^{i-1}(v) | E_1\right)\\ \geq & 1-\frac{\epsilon}{2}.\end{aligned}$$

Note $E_3 = E_4 \cap E_5$. We have

$$\begin{aligned}\Pr(E_3|E_1) &= \Pr(E_4 \cap E_5|E_1)\\ &\geq 1 - \Pr(\bar{E}_4|E_1) - \Pr(\bar{E}_5|E_1)\\ &\geq 1-\epsilon.\end{aligned}$$

where $\bar{E}_4, \bar{E}_5$ are the complement of event E_4, E_5.

□

Lemma 15. *When E_1 and E_2 occurs, if $\lfloor m^- \rfloor < t \leq \frac{\log n}{(1+\alpha)\log \mu}$, we have*

$$H \leq c[(1+\delta)\mu]^{\frac{3}{4}t+\frac{1}{2}} + c[(1+\delta)\mu]^{(\frac{5}{4}-\frac{\alpha}{2})t+2},$$

where c is a constant.

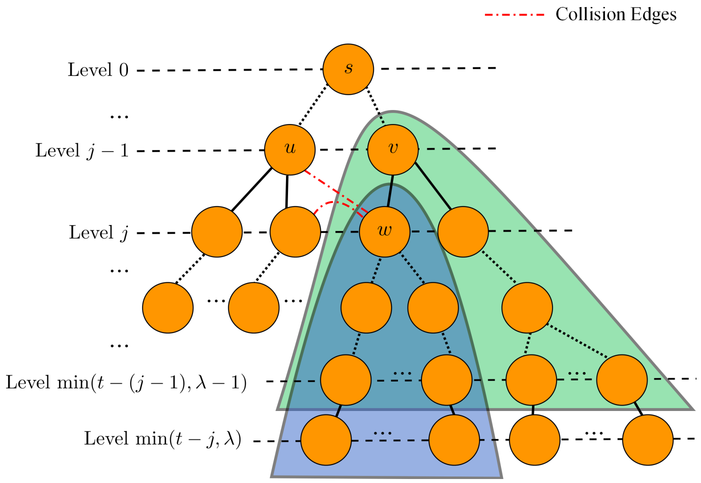


Figure A.3: A pictorial example of upper bounds of H

Proof. Define a *collision removed* breadth-first search tree to be a BFS tree on the graph with all collision nodes removed. Denote by $\mathcal{U}^h(v)$ the set of nodes in the **collision removed** BFS tree from node v with h hops in g_t and $U^h(v) = |\mathcal{U}^h(v)|$. Denote by $\tilde{\mathcal{U}}^h(v)$ the set of nodes that are within h hops from node v on g_t. Recall that a node v is said to locate on level j if $d_{sv} = j$. Denote by $\mathcal{C}_j$ the set of collision nodes on level j in g_t. Therefore, we have

$$H \leq \left| \cup_{j=0}^{t} \cup_{v \in \mathcal{C}_j} \tilde{\mathcal{U}}^{t-d_{s'v}}(v) \right|,$$

where $\tilde{\mathcal{U}}^{t-d_{s'v}}(v)$ is the set of nodes that can be reached from s' within t hops via the collision node v. Next, we will prove that

$$\cup_{j=0}^{t} \cup_{v \in \mathcal{C}_j} \tilde{\mathcal{U}}^{t-d_{s'v}}(v) = \cup_{j=0}^{t} \cup_{v \in \mathcal{C}_j} \mathcal{U}^{t-d_{s'v}}(v).$$

Since $\mathcal{U}^{t-d_{s'v}}(v) \subset \tilde{\mathcal{U}}^{t-d_{s'v}}(v)$, we have

$$\cup_{j=0}^{t} \cup_{v \in \mathcal{C}_j} \tilde{\mathcal{U}}^{t-d_{s'v}}(v) \supseteq \cup_{j=0}^{t} \cup_{v \in \mathcal{C}_j} \mathcal{U}^{t-d_{s'v}}(v).$$

We only need to show that

$$\cup_{j=0}^{t} \cup_{v \in \mathcal{C}_j} \tilde{\mathcal{U}}^{t-d_{s'v}}(v) \subseteq \cup_{j=0}^{t} \cup_{v \in \mathcal{C}_j} \mathcal{U}^{t-d_{s'v}}(v).$$

For any node $w \in \cup_{j=0}^{t} \cup_{v \in \mathcal{C}_j} \tilde{\mathcal{U}}^{t-d_{s'v}}(v)$, we consider the following cases.

- If w is a collision node, we have $w \in \mathcal{U}^{t-d_{s'w}}(w)$. Hence $w \in \cup_{j=0}^{t} \cup_{v \in \mathcal{C}_j} \mathcal{U}^{t-d_{s'v}}(v)$.
- If w is not a collision node, there exists v' such that $w \in \tilde{\mathcal{U}}^{t-d_{s'v'}}(v')$.
  - If the shortest path from w to v' does not contain any other collision nodes, we have $w \in \mathcal{U}^{t-d_{s'v'}}(v')$.
  - If the shortest path from w to v' contains other collision nodes, denote by u the collision node on that path which is the closest to node w. Therefore, there is no collision node on the shortest path from node u to node w. We have
    $$d_{uw} = d_{wv'} - d_{uv'}$$
    Note we have $w \in \tilde{\mathcal{U}}^{t-d_{s'v'}}(v')$, therefore $d_{wv'} \leq t - d_{s'v'}$. Hence, we have
    $$d_{uw} = d_{wv'} - d_{uv'} \leq t - d_{s'v'} - d_{uv'} \leq_{(a)} t - d_{s'u},$$
    where (a) is due to the triangle inequality. Therefore, we have
    $$d_{uw} \leq t - d_{s'u},$$
    and the shortest path from node u to node w contains no collision nodes. Hence,
    $$w \in \mathcal{U}^{t-d_{s'u}}(u)$$

As a summary, we proved

$$\cup_{j=0}^{t} \cup_{v\in\mathcal{C}_j} \tilde{\mathcal{U}}^{t-d_{s'v}}(v) = \cup_{j=0}^{t} \cup_{v\in\mathcal{C}_j} \mathcal{U}^{t-d_{s'v}}(v)$$

Now we can use the collision removed BFS tree to bound H since the branch of collision nodes of traditional BFS tree are already counted in the collision removed BFS tree rooted at these collision nodes. For example, consider the collision removed BFS tree from node w in Figure A.3. We ignore the presence of node u since the branch of node u are already considered in the collision removed BFS tree rooted at node u.

Hence, we have

$$\begin{aligned} H &\leq \left|\cup_{j=0}^{t} \cup_{v\in\mathcal{C}_j} \tilde{\mathcal{U}}^{t-d_{s'v}}(v)\right| \\ &= \left|\cup_{j=0}^{t} \cup_{v\in\mathcal{C}_j} \mathcal{U}^{t-d_{s'v}}(v)\right| \\ &\leq \sum_{j=0}^{t} \sum_{v\in\mathcal{C}_j} U^{t-d_{s'v}}(v) \end{aligned}$$

Since $U^{\lambda}(u)$ is an increasing function of λ and $d_{s'v} \geq |k-j|$ for node v on level j, we have

$$H \leq \sum_{j=0}^{t} \sum_{v\in\mathcal{C}_j} U^{t-|k-j|}(v).$$

We next establish the lemma using the following steps.

- **Step 1: Upper bound on** $U^{t-|k-j|}(w)$. Denote by $\text{par}(w)$ the parent of w on the BFS tree from the source and denote by $\text{par}^i(w)$ the ith ancestor of w on the BFS tree from the source. For example, v is the first ancestor of w ($\text{par}(w) = v$) and $\text{par}(v)$ is the second ancestor of w ($\text{par}^2(w) = \text{par}(v)$) as shown in Figure A.3. Denote by $\sigma^h(v)$ the number of nodes in the collision removed BFS subtree rooted in node v with height h without branch of $\text{par}(v)$.

  Consider node w in Figure A.3 and ignore the presence of the collision nodes. Firstly, we remove the branch of the parent of w. The remaining nodes on the tree are below the level of w. This is because the level of a node w's neighbor can differ from node w's level by at most one and those neighbors that are at the same or higher levels must be collision nodes. The height of the tree is no larger than the total number of hops λ. On the other hand, the tree only contains the nodes within t hops from the actual source s since the tree is based on the infection subnetwork. Since w locates on level j, the height of the tree must be no larger than $t-j$. Therefore, the maximum height of the tree is $\min(t-j,\lambda)$ and denote by $\sigma^{\min(t-j,\lambda)}(w)$ the total number of nodes in the tree as shown in Figure A.3.

  Next, we consider the branch of the parent of w ($v = \text{par}(w)$) in Figure A.3. Note w has only one parent v (all other parents are collision nodes thus removed). Since we considered the λ hops of the removed collision BFS tree rooted at w and it takes one hop from w to v, the branch of node v in the collision removed BFS tree is contained in $\mathcal{U}^{\lambda-1}(v)$.

Therefore, we have

$$U^{\lambda}(w) \leq \sigma^{\min(t-j,\lambda)}(w) + U^{\lambda-1}(v) \tag{A.45}$$

$$= \sigma^{\min(t-j,\lambda)}(w) + U^{\lambda-1}(\text{par}(w)) \tag{A.46}$$

Repeatedly using Equation (A.46), we have

$$U^{\lambda}(w) \leq \sum_{i=0}^{\min(\lambda,j)} \sigma^{\min(t-(j-i),\lambda-i)}(\text{par}^{i}(w)). \tag{A.47}$$

Note the maximum number of hops upward is no larger than λ and the total number of levels above w is no larger than j. Therefore, we only need to consider $\min(\lambda, j)$ levels above w in Equation (A.47).

Intuitively, the upper bound on $U^{\lambda}(w)$ is a collection of trees rooted at level $j-i$ with height $\min(t-(j-i), \lambda-i), \forall i \leq j$. For example, in Figure A.3, the blue area shows the tree rooted in level j with height $\min(t-j, \lambda)$ and the green area shows in tree rooted in level $j-1$ with height $\min(t-(j-1), \lambda-1)$. In this example, we consider the removed collision BFS tree rooted at w. The blue area is the collision removed BFS tree from w after further removing the branch from v and the green area is the collision removed BFS tree from v by further removing the branch from $\text{par}(v)$. The height is no larger than $t-(j-1)$ since node v locates on level $j-1$ and we consider the t hop neighborhood of s. In addition, the height is no larger than $\lambda-1$ since it takes one hop from node w to node v and the total number of possible hops from node w is λ.

According to E_1, we have

$$\sigma^{l}(v) \leq \sum_{h=0}^{l} [(1+\delta)\mu]^{h}, \forall v \in \mathcal{V}(g_t).$$

Hence,

$$U^{\lambda}(w) \leq \sum_{i=0}^{\min(\lambda,j)} \sum_{h=0}^{\min(\lambda-i,t-(j-i))} [(1+\delta)\mu]^{h}. \tag{A.48}$$

Consider $\lambda \in (0, t)$ and $j \in [1, t]$. We obtain an upper bound on $U^{\lambda}(w)$ by analyzing different ranges of λ.

- $\lambda < j$. We have $\min(\lambda, j) = \lambda$.

  - $*$ $\lambda < t - j$. We have $\min(\lambda - i, t - (j - i)) = \lambda - i$. Hence,

$$U^\lambda(w) \leq \sum_{i=0}^{\lambda} \sum_{h=0}^{\lambda - i} [(1+\delta)\mu]^h \tag{A.49}$$

$$\leq \sum_{i=0}^{\lambda} 2[(1+\delta)\mu]^{\lambda - i} \tag{A.50}$$

$$\leq 4[(1+\delta)\mu]^{\lambda}. \tag{A.51}$$

  - $*$ $\lambda \geq t - j$. When $i \leq \frac{\lambda - t + j}{2}$, we have $\lambda - i > t - (j - i)$. Therefore, $\min(\lambda - i, t - (j - i)) = t - (j - i)$. Hence,

$$U^\lambda(w) \tag{A.52}$$

$$\leq \sum_{i=0}^{\left\lfloor \frac{\lambda - t + j}{2} \right\rfloor} \sum_{h=0}^{t - j + i} [(1+\delta)\mu]^h + \sum_{i=\left\lceil \frac{\lambda - t + j}{2} \right\rceil}^{\lambda} \sum_{h=0}^{\lambda - i} [(1+\delta)\mu]^h \tag{A.53}$$

$$\leq \sum_{i=0}^{\left\lfloor \frac{\lambda - t + j}{2} \right\rfloor} 2[(1+\delta)\mu]^{t - j + i} + \sum_{i=\left\lceil \frac{\lambda - t + j}{2} \right\rceil}^{\lambda} 2[(1+\delta)\mu]^{\lambda - i} \tag{A.54}$$

$$\leq 4[(1+\delta)\mu]^{t - j + \left\lfloor \frac{\lambda - t + j}{2} \right\rfloor} + 4[(1+\delta)\mu]^{\lambda - \left\lceil \frac{\lambda - t + j}{2} \right\rceil} \tag{A.55}$$

$$\leq 4[(1+\delta)\mu]^{t - j + \frac{\lambda - t + j}{2}} + 4[(1+\delta)\mu]^{\lambda - \frac{\lambda - t + j}{2}} \tag{A.56}$$

$$\leq 8[(1+\delta)\mu]^{\frac{\lambda + t - j}{2}} \tag{A.57}$$

- $\lambda \geq j$. We have $\min(\lambda, j) = j$.
  - $*$ $\lambda < t - j$. We have $\min(\lambda - i, t - (j - i)) = \lambda - i$. Hence,

$$U^\lambda(w) \leq \sum_{i=0}^{j} \sum_{h=0}^{\lambda - i} [(1+\delta)\mu]^h \tag{A.58}$$

$$\leq \sum_{i=0}^{j} 2[(1+\delta)\mu]^{\lambda - i} \tag{A.59}$$

$$\leq 4[(1+\delta)\mu]^{j}. \tag{A.60}$$

∗ $\lambda \geq t-j$. When $i \leq \frac{\lambda-t+j}{2}$, we have $\lambda - i > t-(j-i)$. Hence,

$$\begin{aligned}
&U^{\lambda}(w) && \text{(A.61)}\\
\leq &\sum_{i=0}^{\lfloor\frac{\lambda-t+j}{2}\rfloor}\sum_{h=0}^{t-j+i}[(1+\delta)\mu]^h + \sum_{i=\lceil\frac{\lambda-t+j}{2}\rceil}^{j}\sum_{h=0}^{\lambda-i}[(1+\delta)\mu]^h && \text{(A.62)}\\
\leq &\sum_{i=0}^{\lfloor\frac{\lambda-t+j}{2}\rfloor} 2[(1+\delta)\mu]^{t-j+i} + \sum_{i=\lceil\frac{\lambda-t+j}{2}\rceil}^{j} 2[(1+\delta)\mu]^{\lambda-i} && \text{(A.63)}\\
\leq &4[(1+\delta)\mu]^{t-j+\lfloor\frac{\lambda-t+j}{2}\rfloor} + 4[(1+\delta)\mu]^{\lambda-\lceil\frac{\lambda-t+j}{2}\rceil} && \text{(A.64)}\\
\leq &4[(1+\delta)\mu]^{t-j+\frac{\lambda-t+j}{2}} + 4[(1+\delta)\mu]^{\lambda-\frac{\lambda-t+j}{2}} && \text{(A.65)}\\
\leq &8[(1+\delta)\mu]^{\frac{\lambda+t-j}{2}} && \text{(A.66)}
\end{aligned}$$

As a summary, we have

$$U^{\lambda}(w) \leq \begin{cases} 4[(1+\delta)\mu]^{\min(\lambda,j)}. & \text{if } \lambda < t-j \\ 8[(1+\delta)\mu]^{\frac{\lambda+t-j}{2}} & \text{if } \lambda \geq t-j. \end{cases}$$

For simplicity, we define U_j^{λ} to be the upper bound on $U^{\lambda}(w)$ for w on level j. We have

$$U_j^{\lambda} = \begin{cases} 4[(1+\delta)\mu]^{\min(\lambda,j)}. & \text{if } \lambda < t-j \\ 8[(1+\delta)\mu]^{\frac{\lambda+t-j}{2}} & \text{if } \lambda \geq t-j. \end{cases} \quad \text{(A.67)}$$

and $U^{\lambda}(w) \leq U_j^{\lambda}$ where the subscript means the level of the nodes and the superscript means the number of hops.

Hence, we have

$$H \leq \sum_{j=0}^{t}\sum_{v\in\mathcal{C}_j} U^{t-|k-j|}(v) \leq \sum_{j=0}^{t} |\mathcal{C}_j| U_j^{t-|k-j|}$$

- **Step 2: Upper bound on $|\mathcal{C}_j|$.** Recall that $\mathcal{C}_j$ is the set of collision nodes on level j. Note one collision edge may connect two nodes on the same level or connect one node on level j and one node on level $j-1$. Therefore, we have

$$|\mathcal{C}_j| \leq 2R_{j+1}, \quad \forall j \leq t-1$$

For $j = t$, since we only consider the t hop neighborhood of the actual source s, we have

$$|\mathcal{C}_t| \leq 2R_t$$

Therefore, we have

$$H \leq 2R_t U_t^{t-|k-t|} + \sum_{j=0}^{t-1} 2R_{j+1} U_j^{t-|k-j|} \tag{A.68}$$

$$= \underbrace{2R_t U_t^k}_{(a)} + \underbrace{\sum_{j=1}^{t} 2R_j U_{j-1}^{t-|k-(j-1)|}}_{(b)} \tag{A.69}$$

- **Step 3** We analyze part (a) and part (b) in equation (A.69) separately.
  - **Step 3.a: Upper bound on part** (a) **in Equation (A.69)**
    Define
    $$\alpha' = \frac{\alpha}{2} + \frac{1}{4}.$$
    and we have $\alpha' \in (1/2, 3/4)$. Since $t \leq \frac{\log n}{(1+\alpha)\log \mu}$, $\alpha < 1$ and $\delta < 1$, we have when n is sufficiently large,
    $$t \leq \frac{\log n}{(1+\alpha)\log \mu} \tag{A.70}$$
    $$t \leq \frac{\log n}{(1+\alpha')\log[(1+\delta)\mu]} \tag{A.71}$$
    $$[(1+\delta)\mu]^{(1+\alpha')t} \leq n, \tag{A.72}$$
    According to Equation (A.67), since $k \geq t - t = 0$, we have
    $$U_t^k = 8[(1+\delta)\mu]^{\frac{k+t-t}{2}} = 8[(1+\delta)\mu]^{\frac{k}{2}}$$
    Based on event E_2, we have
    $$2R_t U_t^k \leq \frac{64}{n}[(1+\delta)\mu]^{2t+1+\frac{k}{2}}$$
    Since $[(1+\delta)\mu]^{(1+\alpha')t} \leq n$,
    $$2R_t U_t^k \leq 64[(1+\delta)\mu]^{(1-\alpha')t+1+\frac{k}{2}}$$
    Since $k \leq t$, we have
    $$2R_t U_t^k \leq 64[(1+\delta)\mu]^{(\frac{3}{2}-\alpha')t+1} \tag{A.73}$$
  - **Step 3.b: Upper bound on part** (b) **in Equation (A.69)**
    Based on event E_3, we have
    $$\sum_{j=1}^{t} 2R_j U_{j-1}^{t-|k-(j-1)|} = \sum_{j=\lfloor m^- \rfloor+1}^{t} 2R_j U_{j-1}^{t-|k-(j-1)|}$$

Therefore, we only consider the cases when $j \geq \lfloor m^- \rfloor + 1$. We will show that

$$t - |k - (j-1)| \geq t - (j-1),$$

when $j \geq \lfloor m^- \rfloor + 1$. As a consequence, we have

$$\begin{aligned} U_{j-1}^{t-|k-(j-1)|} &= 8[(1+\delta)\mu]^{\frac{t-|k-(j-1)|+t-(j-1)}{2}} \\ &= 8[(1+\delta)\mu]^{t-\frac{(j-1)+|k-(j-1)|}{2}} \end{aligned}$$

Based on $t \leq \frac{\log n}{(1+\alpha)\log \mu}$, we have

$$\begin{aligned} \frac{t}{2} &\leq \frac{\log n}{2(1+\alpha)\log \mu} & \text{(A.74)} \\ &\leq \lfloor m^- \rfloor & \text{(A.75)} \end{aligned}$$

for sufficiently large n. Therefore, we have $j - 1 \geq \lfloor m^- \rfloor > \frac{t}{2}$.
When $j-1 \leq k \leq t$, we have $|k-(j-1)| \leq \frac{t}{2} \leq j-1$. When $0 \leq k < j-1$, we have $|k-(j-1)| = (j-1) - k \leq j-1$. We have

$$|k-(j-1)| \leq j-1, \forall k \in [0, t].$$

Hence,

$$t - |k-(j-1)| \geq t - (j-1).$$

Therefore, for all the discussions in Step 3.b, based on Equation A.69, we have

$$U_{j-1}^{t-|k-(j-1)|} = 8[(1+\delta)\mu]^{t-\frac{(j-1)+|k-(j-1)|}{2}}$$

Based on event E_2, we have

$$\begin{aligned} &\sum_{j=1}^{t} 2R_j U_{j-1}^{t-|k-(j-1)|} \\ \leq &128\mu \sum_{j=\lfloor m^- \rfloor+1}^{\lceil m^+ \rceil - 1} [(1+\delta)\mu]^{t-\frac{j-1+|k-(j-1)|}{2}} \\ &+\frac{64}{n} \sum_{j=\lceil m^+ \rceil}^{t} [(1+\delta)\mu]^{2j+1+t-\frac{j-1+|k-(j-1)|}{2}} \end{aligned}$$

Since $[(1+\delta)\mu]^{(1+\alpha')t} \leq n$,

$$\begin{aligned}
&\sum_{j=1}^{t} 2R_j U_{j-1}^{t-|k-(j-1)|} \\
\leq & 128\mu \sum_{j=\lfloor m^- \rfloor+1}^{\lceil m^+ \rceil -1} [(1+\delta)\mu]^{t-\frac{j-1+|k-(j-1)|}{2}} \\
&+ 64 \sum_{j=\lceil m^+ \rceil}^{t} [(1+\delta)\mu]^{2j+1-\alpha' t-\frac{j-1+|k-(j-1)|}{2}}
\end{aligned}$$

Next, we discuss the upper bounds for different k values. We first show several necessary inequalities. We have

$$m^+ - m^- = \frac{2\log\mu + \log 8}{2\log[(1+\delta)\mu]} \leq 1 + \frac{\log 8}{2\log\mu} < 2. \tag{A.76}$$

Recall, we consider the case where $t > \lfloor m^- \rfloor$. As shown in A.75, we have

$$\lfloor m^- \rfloor \in \left[\frac{t}{2}, t\right) \tag{A.77}$$

Hence, we have

$$\lceil m^+ \rceil \in \left[\frac{t}{2}, t+3\right) \tag{A.78}$$

Then, we consider the following cases for different k values. Recall that k is the level of node s'.

$*$ $k \leq \lfloor m^- \rfloor$. We have

$$\begin{aligned}
&\sum_{j=1}^{t} 2R_j U_{j-1}^{t-|k-(j-1)|} \\
\leq& 128\mu \sum_{j=\lfloor m^- \rfloor+1}^{\lceil m^+ \rceil -1} [(1+\delta)\mu]^{t-\frac{j-1+|k-(j-1)|}{2}} \\
&+ 64 \sum_{j=\lceil m^+ \rceil}^{t} [(1+\delta)\mu]^{2j+1-\alpha' t-\frac{j-1+|k-(j-1)|}{2}} \\
=& 128\mu \sum_{j=\lfloor m^- \rfloor+1}^{\lceil m^+ \rceil -1} [(1+\delta)\mu]^{t-\frac{j-1-(k-(j-1))}{2}} \\
&+ 64 \sum_{j=\lceil m^+ \rceil}^{t} [(1+\delta)\mu]^{2j+1-\alpha' t-\frac{j-1-(k-(j-1))}{2}} \\
=& 128\mu \sum_{j=\lfloor m^- \rfloor+1}^{\lceil m^+ \rceil -1} [(1+\delta)\mu]^{t-j+1+\frac{k}{2}} \\
&+ 64 \sum_{j=\lceil m^+ \rceil}^{t} [(1+\delta)\mu]^{-\alpha' t+j+2+\frac{k}{2}} \\
\leq& 256\mu[(1+\delta)\mu]^{t-\lfloor m^- \rfloor+\frac{k}{2}} \\
&+ 128[(1+\delta)\mu]^{(1-\alpha')t+2+\frac{k}{2}} \\
\leq_{(a)}& 256[(1+\delta)\mu]^{\frac{3}{4}t} + 128[(1+\delta)\mu]^{(\frac{3}{2}-\alpha')t+2}
\end{aligned}$$

where (a) is due to $k \leq \lfloor m^- \rfloor$ and $\frac{t}{2} \leq \lfloor m^- \rfloor < t$.

$*$ $\lfloor m^- \rfloor + 1 \leq k \leq \lceil m^+ \rceil - 2$. In this case, we have

$$k \in [\frac{t}{2}+1, t+1),$$

according to Inequalties (A.78) and (A.77). Hence, we have

$$\begin{aligned}
&\sum_{j=1}^{t} 2R_j U_{j-1}^{t-|k-(j-1)|}\\
\le&128\mu \sum_{j=\lfloor m^-\rfloor+1}^{k} [(1+\delta)\mu]^{t-\frac{j-1+|k-(j-1)|}{2}}\\
&+128\mu \sum_{j=k+1}^{\lceil m^+\rceil-1} [(1+\delta)\mu]^{t-\frac{j-1+|k-(j-1)|}{2}}\\
&+64 \sum_{j=\lceil m^+\rceil}^{t} [(1+\delta)\mu]^{2j+1-\alpha' t-\frac{j-1+|k-(j-1)|}{2}}\\
=&128\mu \sum_{j=\lfloor m^-\rfloor+1}^{k} [(1+\delta)\mu]^{t-\frac{j-1+(k-(j-1))}{2}}\\
&+128\mu \sum_{j=k+1}^{\lceil m^+\rceil-1} [(1+\delta)\mu]^{t-\frac{j-1-(k-(j-1))}{2}}\\
&+64 \sum_{j=\lceil m^+\rceil}^{t} [(1+\delta)\mu]^{2j+1-\alpha' t-\frac{j-1-(k-(j-1))}{2}}\\
=&128\mu \sum_{j=\lfloor m^-\rfloor+1}^{k} [(1+\delta)\mu]^{t-\frac{k}{2}}\\
&+128\mu \sum_{j=k+1}^{\lceil m^+\rceil-1} [(1+\delta)\mu]^{t-j+1+\frac{k}{2}}\\
&+64 \sum_{j=\lceil m^+\rceil}^{t} [(1+\delta)\mu]^{-\alpha' t+j+2+\frac{k}{2}}\\
\le_{(a)}&256\mu[(1+\delta)\mu]^{t-\frac{k}{2}}\\
&+128[(1+\delta)\mu]^{(1-\alpha')t+2+\frac{k}{2}}\\
\le&256[(1+\delta)\mu]^{\frac{3}{4}t+\frac{1}{2}}+128[(1+\delta)\mu]^{(\frac{3}{2}-\alpha')t+2},
\end{aligned}$$

where (a) is due to $m^+ - m^- < 2$ which we proved in Inequality (A.76) and $k \in [\frac{t}{2}+1, t+1)$.

- $k \ge \lceil m^+\rceil - 1$. In this case, we have

$$k \in [\frac{t}{2}-1, t],$$

according to Inequalties (A.78) and (A.77). Hence, we have

$$\begin{aligned}
&\sum_{j=1}^{t} 2R_j U_{j-1}^{t-|k-(j-1)|}\\
\leq&128\mu \sum_{j=\lfloor m^- \rfloor+1}^{\lceil m^+ \rceil-1} [(1+\delta)\mu]^{t-\frac{j-1+|k-(j-1)|}{2}}\\
&+64 \sum_{j=\lceil m^+ \rceil}^{k} [(1+\delta)\mu]^{2j+1-\alpha' t-\frac{j-1+|k-(j-1)|}{2}}\\
&+64 \sum_{j=k+1}^{t} [(1+\delta)\mu]^{2j+1-\alpha' t-\frac{j-1+|k-(j-1)|}{2}}\\
=&128\mu \sum_{j=\lfloor m^- \rfloor+1}^{\lceil m^+ \rceil-1} [(1+\delta)\mu]^{t-\frac{j-1+(k-(j-1))}{2}}\\
&+64 \sum_{j=\lceil m^+ \rceil}^{k} [(1+\delta)\mu]^{2j+1-\alpha' t-\frac{j-1+(k-(j-1))}{2}}\\
&+64 \sum_{j=k+1}^{t} [(1+\delta)\mu]^{2j+1-\alpha' t-\frac{j-1-(k-(j-1))}{2}}\\
=&128\mu \sum_{j=\lfloor m^- \rfloor+1}^{\lceil m^+ \rceil-1} [(1+\delta)\mu]^{t-\frac{k}{2}}\\
&+64 \sum_{j=\lceil m^+ \rceil}^{k} [(1+\delta)\mu]^{2j+1-\alpha' t-\frac{k}{2}}\\
&+64 \sum_{j=k+1}^{t} [(1+\delta)\mu]^{-\alpha' t+j+2+\frac{k}{2}}\\
\leq_{(a)}&256\mu[(1+\delta)\mu]^{t-\frac{k}{2}}+128[(1+\delta)\mu]^{\frac{3k}{2}-\alpha' t+1}\\
&+128[(1+\delta)\mu]^{(1-\alpha')t+2+\frac{k}{2}}\\
\leq&256[(1+\delta)\mu]^{\frac{3}{4}t+\frac{1}{2}}+128[(1+\delta)\mu]^{(\frac{3}{2}-\alpha')t+1}\\
&+128[(1+\delta)\mu]^{(\frac{3}{2}-\alpha')t+2}\\
\leq&256[(1+\delta)\mu]^{\frac{3}{4}t+\frac{1}{2}}+256[(1+\delta)\mu]^{(\frac{3}{2}-\alpha')t+2}
\end{aligned}$$

(a) is due to $m^+ - m^- < 2$ which we proved in Inequality A.76 and $k \in [\frac{t}{2}-1, t]$.

Therefore, we obtain a universal bound for different k.

$$\sum_{j=1}^{t} 2R_j U_{j-1}^{t-|k-(j-1)|} \leq c'[(1+\delta)\mu]^{\frac{3}{4}t+\frac{1}{2}} \tag{A.79}$$

$$+ c'[(1+\delta)\mu]^{(\frac{3}{2}-\alpha')t+2}, \tag{A.80}$$

for all $k \in [1, t]$, where $c' \geq 256$.

As a summary, based on Equations (A.69),(A.73) and (A.80)

$$H \leq c[(1+\delta)\mu]^{\frac{3}{4}t+\frac{1}{2}} + c[(1+\delta)\mu]^{(\frac{3}{2}-\alpha')t+2} \tag{A.81}$$

$$= c[(1+\delta)\mu]^{\frac{3}{4}t+\frac{1}{2}} + c[(1+\delta)\mu]^{(\frac{5}{4}-\frac{\alpha}{2})t+2} \tag{A.82}$$

for all $k \in [1, t]$, where $c \geq 257$. □

A.4 Proof of Theorem 4

Proof. Similar to the proof of Theorem 3, we assume E_1, E_2 and E_3 occur. The BND one node can have is bounded by the number of all infected nodes. Therefore, the upper bound on BND is the sum of degree of all infected nodes. The edges of one infected node compose three parts: (1) the edge which infects the node; (2) the edges between the node and its offsprings; (3) the collision edges attaching to the node. Therefore, the total degree of all the infected nodes is upper bounded by

$$\sum_{i=0}^{t} Z_i^{\leq t} + \sum_{i=0}^{t} Z_i^{\leq t}[(1+\delta)\mu] + 2R_{t+1}$$

To use $\sum_{i=0}^{t} Z_i^{\leq t}[(1+\delta)\mu]$ above as the upper bound on offsprings, we need to extend E_1 to the range of $\mathcal{L}_t$. It is easy to check that

$$E_1' = \{\forall v \in \mathcal{L}_t, \phi'(v) \in ((1-\delta)\mu, (1+\delta)\mu)\}.$$

happens with a high probability with the same proof of Lemma 12.

A lower bound on BND of the actual source is

$$Z_t^t[(1-\delta)\mu]$$

according to E_1'. Therefore, we have

$$\frac{Z_t^t[(1-\delta)\mu]}{\sum_{i=0}^t Z_i^{\leq t} + \sum_{i=0}^t Z_i^{\leq t}[(1+\delta)\mu] + 2R_{t+1}} \tag{A.83}$$

$$= \frac{(1-\delta)\mu}{\frac{\sum_{i=0}^t Z_i^{\leq t}}{Z_t^t}(1+(1+\delta)\mu) + \frac{2R_{t+1}}{Z_t^t}} \tag{A.84}$$

$$\geq_{(a)} \frac{(1-\delta)\mu}{\frac{1+(1+\delta)\mu}{1-\epsilon'} + \frac{2R_{t+1}}{Z_t^t}}. \tag{A.85}$$

$$\geq \frac{(1-\delta)}{\frac{\frac{1}{\mu}+(1+\delta)}{1-\epsilon'} + \frac{2R_{t+1}}{Z_t^t\mu}} \tag{A.86}$$

$$\geq_{(b)} \frac{(1-\delta)}{\frac{\delta''+(1+\delta)}{1-\epsilon'} + \delta'} \tag{A.87}$$

$$\geq \frac{1-\delta}{1+1.1\delta}, \tag{A.88}$$

where inequality (a) holds due to Lemma 16, inequality (b) is based on Lemma 17 and $\delta'', \delta', \epsilon'$ can be arbitrarily small when $n \to \infty$.

In the proof of Lemma 12, we need $\mu > \frac{2+\delta}{\delta^2}\log n$. Hence, we have

$$\frac{Z_t^t[(1-\delta)\mu]}{\sum_{i=0}^t Z_i^{\leq t} + \sum_{i=0}^t Z_i^{\leq t}[(1+\delta)\mu] + 2R_{t+1}} \geq \frac{1-\delta}{1+1.1\delta}$$

when $\mu > \frac{2+\delta}{\delta^2}\log n$.

Assume we want the ratio to be $\geq 1-x$, we have

$$\frac{1-\delta}{1+1.1\delta} = 1-x.$$

Therefore

$$\delta = \frac{x}{2.1-1.1x}$$

Hence,

$$\frac{2+\delta}{\delta^2} = \frac{1.32x^2 - 7.14x + 8.82}{x^2}$$

Therefore, when

$$\mu > \frac{9}{x^2}\log n > \frac{1.32x^2 - 7.14x + 8.82}{x^2}\log n$$

we have the ratio $\geq 1-x$.

Note, the condition that $\alpha > \frac{1}{2}$ is not used in the high probability result of E_1, E_2, E_3. Therefore, we only need $\alpha \in (0,1)$ for this theorem. Therefore, the theorem is proved.

□

Lemma 16. *If the conditions in Theorem 4 hold and events E_1, E_2 and E_3 occur, we have given any $\epsilon > 0$, for sufficiently large n, the following inequality holds*

$$\frac{Z_t^t}{\sum_{i=0}^t Z_i^{\leq t}} \geq 1 - \epsilon.$$

Proof. For any $\epsilon > 0$, we use induction and assume that

$$Z_{m-1}^{m-1} \geq (1-\epsilon) \sum_{i=0}^{m-1} Z_i^{\leq m-1}$$

Consider time slot m, we have

$$\begin{aligned}
\frac{Z_m^m}{\sum_{i=0}^m Z_i^{\leq m}} &= \frac{Z_m^m}{\sum_{i=0}^m Z_i^{\leq m-1} + \sum_{i=0}^m Z_i^m} \\
&=_{(a)} \frac{Z_m^m}{\sum_{i=0}^{m-1} Z_i^{\leq m-1} + \sum_{i=0}^{m-1} Z_i^m + Z_m^m} \\
&= \frac{1}{\underbrace{\frac{\sum_{i=0}^{m-1} Z_i^{\leq m-1}}{Z_m^m}}_{\text{A}} + \underbrace{\frac{\sum_{i=0}^{m-1} Z_i^m}{Z_m^m}}_{\text{B}} + 1}
\end{aligned}$$

In (a) we use that $Z_m^{\leq m-1} = 0$ and $Z_i^{\leq m} = Z_i^{\leq m-1} + Z_i^m$.

Use induction assumption for part A, we have

$$\frac{Z_m^m}{\sum_{i=0}^m Z_i^{\leq m}} \geq \frac{1}{\frac{Z_{m-1}^{m-1}}{(1-\epsilon)Z_m^m} + \underbrace{\frac{\sum_{i=0}^{m-1} Z_i^m}{Z_m^m}}_{\text{B}} + 1} \tag{A.89}$$

Based on event E_3, we have

$$\frac{\frac{Z_{m-1}^{m-1}}{Z_m^m}}{} \leq \frac{1}{(1-\delta)^2 \mu q}$$

and

$$Z_m^m \geq [(1-\delta)^2 \mu q]^m.$$

Next, we establish an upper bound on $\sum_{i=0}^{m-1} Z_i^m$. Note $\sum_{i=0}^{m-1} Z_i^m$ represents the number of nodes which are from level 0 to level $m-1$ and are infected at time m. Denote by $C(\mathcal{V})$ the set of offsprings of node set $\mathcal{V}$ who are not collision nodes and were infected by node $\mathcal{V}$. Define $C^2(\mathcal{V}) = C(C(\mathcal{V}))$. Recall the number of collsion nodes from level 0 to level $m-1$ is upper bounded by $2R_m$. We establish an upper bound as following

$$\sum_{i=0}^{m-1} Z_i^m \leq 2R_m + \left| C\left(\cup_{i=0}^{m-2} \mathcal{Z}_i^{m-1}\right) \right|.$$

Similarly, we have

$$\left|\cup_{i=0}^{m-2}\mathcal{Z}_i^{m-1}\right| \leq 2R_{m-1} + \left|C\left(\cup_{i=0}^{m-3}\mathcal{Z}_i^{m-2}\right)\right|.$$

Based on E_1, we have

$$\sum_{i=0}^{m-1} Z_i^m \leq 2R_m + 2R_{m-1}[(1+\delta)\mu] + \left|C^2\left(\cup_{i=0}^{m-3}\mathcal{Z}_i^{m-2}\right)\right|$$

Repeating the step above, we have

$$\sum_{i=0}^{m-1} Z_i^m \leq \sum_{j=0}^{m} 2R_j[(1+\delta)\mu]^{m-j}$$

Based on E_2, we evaluate the upper bound for different values of m.

- $0 < m \leq \lfloor m^- \rfloor$. We have

  $$\sum_{i=0}^{m-1} Z_i^m = 0.$$

  Hence,

  $$\begin{aligned}\frac{Z_m^m}{\sum_{i=0}^m Z_i^{\leq m}} \geq & \frac{1}{\frac{Z_{m-1}^{m-1}}{(1-\epsilon)Z_m^m}+1} \\ \geq & \frac{1}{\frac{1}{(1-\epsilon)(1-\delta)^2\mu q}+1}\end{aligned}$$

  For any $\epsilon > 0$, we have

  $$\frac{1}{(1-\epsilon)(1-\delta)^2\mu q} \leq \epsilon$$

  for sufficiently large n.

  Therefore, we have

  $$\frac{Z_m^m}{\sum_{i=0}^m Z_i^{\leq m}} \geq (1-\epsilon).$$

- $\lfloor m^- \rfloor < m < \lceil m^+ \rceil$. We have

  $$\begin{aligned}\sum_{i=0}^{m-1} Z_i^m \leq & \sum_{j=\lfloor m^- \rfloor+1}^{\lceil m^+ \rceil-1} 2R_j[(1+\delta)\mu]^{m-j} \\ \leq_{(a)} & 32\mu[(1+\delta)\mu]^{m-\lfloor m^- \rfloor-1}\end{aligned}$$

  (a) is based on the fact that $R_j \leq 8\mu$ and $\lceil m^+ \rceil - \lfloor m^- \rfloor \leq 2$.

Hence, we have

$$\begin{aligned}
&\frac{\sum_{i=0}^{m-1} Z_i^m}{Z_m^m} \\
\le&\frac{32\mu[(1+\delta)\mu]^{m-\lfloor m^-\rfloor-1}}{[(1-\delta)^2\mu q]^m} \\
\le&\frac{32}{\mu}\frac{(1+\delta)^{m-\lfloor m^-\rfloor-1}}{(1-\delta)^{2m}q^m\mu^{\lfloor m^-\rfloor-1}} \\
=&\frac{32}{\mu}\left(\frac{(1+\delta)^{1-\frac{\lfloor m^-\rfloor-1}{m}}}{(1-\delta)^2 q\mu^{\frac{\lfloor m^-\rfloor-1}{m}}}\right)^m \\
\le&\frac{32}{\mu}\left(\frac{(1+\delta)^{1-\frac{\lfloor m^-\rfloor-1}{\lceil m^+\rceil}}}{(1-\delta)^2 q\mu^{\frac{\lfloor m^-\rfloor-1}{\lceil m^+\rceil}}}\right)^m
\end{aligned}$$

Note $\lceil m^+\rceil < \lfloor m^-\rfloor + 2$. For sufficiently large n, we have

$$\frac{\lfloor m^-\rfloor-1}{\lceil m^+\rceil} \ge 1-\frac{3}{\lfloor m^-\rfloor} \ge \frac{1}{2}.$$

Hence we have

$$\frac{\sum_{i=0}^{m-1} Z_i^m}{Z_m^m} \le \frac{32}{\mu}\left(\frac{(1+\delta)^{\frac{1}{2}}}{(1-\delta)^2 q\mu^{\frac{1}{2}}}\right)^m \le \frac{\epsilon}{2}$$

for sufficiently large n.

In addition, we have

$$\frac{1}{(1-\delta)^2 q\mu} \le \frac{\epsilon}{2}$$

for sufficiently large n.

$$\begin{aligned}
\frac{Z_m^m}{\sum_{i=0}^m Z_i^{\le m}} \ge& \frac{1}{\frac{Z_{m-1}^{m-1}}{(1-\epsilon)Z_m^m}+\frac{\sum_{i=0}^{m-1}Z_i^m}{Z_m^m}+1} && \text{(A.90)} \\
\ge&\frac{1}{\frac{\epsilon}{2(1-\epsilon)}+\frac{\epsilon}{2}+1} \ge (1-\epsilon). && \text{(A.91)}
\end{aligned}$$

- $\lceil m^+\rceil \le m \le t$. Define

  $$\alpha' = \frac{\alpha}{2}+\frac{1}{4}.$$

  and we have $\alpha' \in (1/4, 3/4)$. Follow the same argument in Equation (A.72), we obtain that

  $$[(1+\delta)\mu]^{(1+\alpha')t} \le n. \qquad \text{(A.92)}$$

We have

$$
\begin{aligned}
\sum_{i=0}^{m-1} Z_i^m \leq & \sum_{j=\lfloor m^- \rfloor+1}^{\lceil m^+ \rceil -1} 2R_j[(1+\delta)\mu]^{m-j} \\
& + \sum_{j=\lceil m^+ \rceil}^{m-1} 2R_j[(1+\delta)\mu]^{m-j} \\
\leq & 32\mu[(1+\delta)\mu]^{m-\lfloor m^- \rfloor -1} \\
& + \frac{8}{n} \sum_{j=\lceil m^+ \rceil}^{m-1} [(1+\delta)\mu]^{m+j+1} \\
\leq & 32\mu[(1+\delta)\mu]^{m-\lfloor m^- \rfloor -1} + \frac{16}{n}[(1+\delta)\mu]^{2m} \\
\leq & 32\mu[(1+\delta)\mu]^{m-\lfloor m^- \rfloor -1} \\
& + 16[(1+\delta)\mu]^{2m-(1+\alpha')t}
\end{aligned}
$$

The last inequality holds based on Inequality (A.92).

Hence, we have

$$
\begin{aligned}
& \frac{\sum_{i=0}^{m-1} Z_i^m}{Z_m^m} \\
\leq & \underbrace{\frac{32\mu[(1+\delta)\mu]^{m-\lfloor m^- \rfloor -1}}{[(1-\delta)^2 q\mu]^m}}_{(A)} + \underbrace{\frac{16[(1+\delta)\mu]^{2m-(1+\alpha')t}}{[(1-\delta)^2 q\mu]^m}}_{(B)}
\end{aligned}
$$

(A) has been handled in the previous case. For sufficiently large n we have $(A) \leq \frac{\epsilon}{4}$.

Next, we focus on (B). Since $m \leq t$, for sufficiently large n, we have

$$
\frac{m+1}{t} \leq 1 + \frac{\alpha'}{2}.
$$

Hence, for sufficiently large n

$$
\begin{aligned}
& \frac{16[(1+\delta)\mu]^{2m-(1+\alpha')t}}{[(1-\delta)^2 q\mu]^m} \\
= & \frac{16}{\mu} \left(\frac{(1+\delta)^{\frac{2m}{t}-1-\alpha'}}{(1-\delta)^{2\frac{m}{t}} q^{\frac{m}{t}}} \mu^{\frac{m+1}{t}-1-\alpha'} \right)^t \\
\leq & \frac{16}{\mu} \left(\frac{(1+\delta)^{2-\alpha'}}{(1-\delta)^2 q} \mu^{-\alpha'/2} \right)^t \\
\leq & \frac{\epsilon}{4}.
\end{aligned}
$$

Hence, we have

$$\frac{\sum_{i=0}^{m-1} Z_i^m}{Z_m^m} \leq \frac{\epsilon}{2}$$

Following the analysis in the previous case, we have

$$\frac{Z_m^m}{\sum_{i=0}^{m} Z_i^{\leq m}} \geq 1 - \epsilon.$$

As a summary, we proved that

$$\frac{Z_t^t}{\sum_{i=0}^{t} Z_i^{\leq t}} \geq 1 - \epsilon.$$

□

Lemma 17. *If the conditions in Theorem 4 hold and events E_1, E_2 and E_3 occur, we have given any $\epsilon > 0$, for sufficiently large n, the following inequality holds*

$$\frac{2R_{t+1}}{Z_t^t \mu} \leq \epsilon \tag{A.93}$$

Proof. Note the upper bound of R_{t+1} can be obtained by a same proof of Lemma 13 and the conclusions are the same when we extend the range from t to $t+1$. Based on Lemma 14, we have

$$Z_t^t \geq [(1-\delta)^2\mu]^t.$$

When $t < \lceil m^+ \rceil$, Inequality A.93 trivially holds.

For $t \geq \lceil m^+ \rceil$, we have

$$\begin{aligned}
&\frac{2R_{t+1}}{Z_t^t \mu} && \text{(A.94)}\\
\leq&\frac{8[(1+\delta)\mu]^{2t+3}}{n\mu[(1-\delta)^2\mu]^t} && \text{(A.95)}\\
=&\frac{8(1+\delta)^{2t+3}}{(1-\delta)^{2t}}\frac{1}{\mu^{\alpha t-2}} && \text{(A.96)}\\
=&\frac{8}{\mu}\left(\frac{1+\delta}{(1-\delta)^2} \times \frac{1}{\mu^{\frac{\alpha t-3}{2t+3}}}\right)^{2t+3} && \text{(A.97)}
\end{aligned}$$

Note $\frac{\alpha t-3}{2t+3} > 0$ for sufficiently large t. Therefore, we have

$$\frac{2R_{t+1}}{Z_t^t \mu} \leq \epsilon.$$

For sufficiently large n. □

A.5 Proof of Lemma 5

We present the proof of Theorem 4.2 from (Draief and Massouli, 2010) with some minor changes to provide a more specific lower bound on μq. This proof is included for the sake of completeness and is not a contribution of this dissertation.
Proof. Given some $\epsilon > 0$, define

$$d_j^{\pm} = \begin{cases} (1 \pm \epsilon)^j \mu^j & \text{if } j = 1, 2, \\ (1 \pm \epsilon)^2 (1 \pm \frac{\epsilon}{\mu})^{j-2} \mu^j & \text{if } j = 3, \cdots, D'. \end{cases}$$

where $D' = \left\lceil \frac{\log n}{2 \log \mu} \right\rceil$. Define

$$\Gamma_i(u) = \{v : d_{uv}^g = i\},$$

and

$$d_i(u) = |\Gamma_i(u)|.$$

We first prove the following lemma.

Lemma 18. *Let $\epsilon > 0$ be fixed. Define for all $u \in \{1, \cdots, n\}$ and all $i = 1, \cdots, D'$, the event $E_i(u)$ by*

$$E_i(u) = \{d_i^- \leq d_i(u) \leq d_i^+\}.$$

Assumes that $\gamma_l \log n \leq \mu << \sqrt{n}$, for large enough n,we have

$$\Pr(E_i(u)) \geq 1 - D' n^{-\frac{\gamma_l \epsilon^2}{2+\epsilon}}, \ u \in \{1, \cdots, n\}, \ i = 1, \cdots, D'.$$

Proof. Let $u \in \{1, \cdots, n\}$ and $i \in \{1, \cdots, D'\}$ be fixed. Note that, conditional on $d_1(u), \cdots, d_{i-1}(u), d_i(u)$ admits a binomial distribution with parameters

$$\begin{aligned} &\mathcal{L}(d_i(u) | d_1(u), \cdots, d_{i-1}(u)) \\ =&\mathrm{Bi}(n - 1 - d_1(u) - \cdots - d_{i-1}(u), 1 - (1-p)^{d_{i-1}(u)}) \end{aligned}$$

where $\mathcal{L}(X|\mathcal{F})$ is the distribution of the random variable X conditional on the event $\mathcal{F}$. Denote by $\bar{E}_i(u)$ the complement of $E_i(u)$. It readily follows that

$$\begin{aligned} &\Pr(\bar{E}_i(u) | E_1(u), \cdots, E_{i-1}(u)) \\ \leq &\Pr(\mathrm{Bi}(n, 1 - (1-p)^{d_{i-1}^+}) \geq d_i^+) \\ + &\Pr(\mathrm{Bi}(n - 1 - d_1^+ - \cdots - d_{i-1}^+, 1 - (1-p)^{d_{i-1}^-}) \leq d_i^-). \end{aligned}$$

Note that, for all $j < D'$, one has

$$\begin{aligned}
d_j^- &\le d_j^+ \le d_{D'-1}^+ \\
&= (1+\epsilon)^2 \left(1+\frac{\epsilon}{\mu}\right)^{D'-3} \mu^{D'-1} \\
&= \left(\frac{\mu(1+\epsilon)}{\mu+\epsilon}\right)^2 (\mu+\epsilon)^{D'-1} \\
&\le \left(\frac{\mu(1+\epsilon)}{\mu+\epsilon}\right)^2 (\mu+\epsilon)^{\frac{\log n}{2\log\mu}} \\
&= \left(\frac{\mu(1+\epsilon)}{\mu+\epsilon}\right)^2 \exp\left(\frac{\log n}{2}\frac{\log(\mu+\epsilon)}{\log\mu}\right) \\
&= \left(\frac{\mu(1+\epsilon)}{\mu+\epsilon}\right)^2 \exp\left(\frac{\log n}{2}\right) \\
&\times \exp\left(\frac{\log n}{2}\left(\frac{\log(\mu+\epsilon)}{\log\mu} - 1\right)\right) \\
&= \left(\frac{\mu(1+\epsilon)}{\mu+\epsilon}\right)^2 \exp\left(\frac{\log n}{2}\right) \\
&\times \exp\left(\frac{\log n}{2}\frac{\log(\mu+\epsilon)-\log\mu}{\log\mu}\right) \\
&= \left(\frac{\mu(1+\epsilon)}{\mu+\epsilon}\right)^2 \\
\exp\left(\frac{\log n}{2}\right) &\times \exp\left(\frac{\log n}{2}\frac{\log(1+\epsilon/\mu)}{\log\mu}\right)
\end{aligned}$$

Note $\log(1+x) \le x$ for $x \ge 0$. We have

$$\begin{aligned}
&\le \left(\frac{\mu(1+\epsilon)}{\mu+\epsilon}\right)^2 \sqrt{n}\exp\left(\frac{\log n}{2}\frac{\epsilon}{\mu\log\mu}\right) \\
&\le (1+\epsilon)^3\sqrt{n}
\end{aligned}$$

Since $\mu \ge \gamma_1 \log n$, we have

$$\begin{aligned}
&\le \left(\frac{\mu(1+\epsilon)}{\mu+\epsilon}\right)^2 \sqrt{n}\exp\left(\frac{1}{2}\frac{\epsilon}{\gamma_l\log\mu}\right) \\
&\le (1+\epsilon)^2\sqrt{n}\exp\left(\frac{1}{2}\frac{\epsilon}{\gamma_l\log\mu}\right)
\end{aligned}$$

For sufficiently large n, we have μ is sufficiently large and $\exp\left(\frac{1}{2}\frac{\epsilon}{\gamma_l\log\mu}\right) \to 1$ as $n \to \infty$. Hence, we have

$$= (1+\epsilon)^2(1+o(1))\sqrt{n}$$

Next, we compute the mean of $\mathrm{Bi}(n, 1-(1-p)^{d_{i-1}^+})$ and $\mathrm{Bi}(n-1-d_1^+ - \cdots - d_{i-1}^+, 1-(1-p)^{d_{i-1}^-})$.

Since $i-1 \leq D'-1$, we have $d_{i-1}^+ p = d_{i-1}^+ \frac{\mu}{n} \to 0$ as $n \to \infty$. Based on Taylor expansion, we have we have

$$(1-p)^{d_{i-1}^+} = 1 - d_{i-1}^+ p + o(d_{i-1}^+ p)$$

Hence,

$$\begin{aligned} & n(1-(1-p)^{d_{i-1}^+}) \\ =& d_{i-1}^+ pn - o(d_{i-1}^+ pn) = (1-o(1)) d_{i-1}^+ \mu \end{aligned}$$

Note

$$\begin{aligned} & n - 1 - d_1^+ - \cdots - d_{i-1}^+ \geq n - D'(1+\epsilon)^2(1+o(1))\sqrt{n} \\ \geq & n - (1+\epsilon)^2(1+o(1)) \log n \sqrt{n} \end{aligned}$$

Therefore

$$\begin{aligned} & (n-1-d_1^+ - \cdots - d_{i-1}^+)(1-(1-p)^{d_{i-1}^-}) \\ & \geq (n-(1+\epsilon)^2(1+o(1)\log n\sqrt{n})(d_{i-1}^- p - o(d_{i-1}^- p)) \\ & = d_{i-1}^- pn - o(d_{i-1}^- pn) - d_{i-1}^- p(1+\epsilon)^2(1+o(1)\log n \sqrt{n} \\ & + o(d_{i-1}^- p(1+\epsilon)^2(1+o(1)\log n\sqrt{n}) \\ & \geq (1-o(1)) d_{i-1}^- \mu \end{aligned}$$

Using the Chernoff bound, we have

$$\begin{aligned} & \Pr\left(\mathrm{Bi}(n, 1-(1-p)^{d_{i-1}^+}) \geq d_i^+\right) \\ & \leq \exp\left(-\frac{\xi^2}{2+\xi} n \left(1-(1-p)^{d_{i-1}^+}\right)\right) \end{aligned}$$

where

$$(1+\xi) n \left(1-(1-p)^{d_{i-1}^+}\right) = d_i^+$$

Therefore,

$$\xi = \frac{d_i^+}{(1-o(1)) d_{i-1}^+ \mu} - 1 \geq \begin{cases} \epsilon & \text{if } j = 1, 2, \\ \frac{\epsilon}{\mu} & \text{if } j = 3, \cdots, D'. \end{cases}$$

Therefore, when $i = 1, 2$ we have

$$\begin{aligned}
&\Pr\left(\mathrm{Bi}(n, 1-(1-p)^{d_{i-1}^+}) \geq d_i^+\right)\\
&\leq \exp\left(-\frac{\xi^2}{2+\xi}(1-o(1))d_{i-1}^+\mu\right)\\
&\leq \exp\left(-\frac{\xi^2}{2+\xi}(1-o(1))\mu\right)\\
&\leq \exp\left(-\frac{\xi^2}{2+\xi}(1-o(1))\gamma_l \log n\right)\\
&\leq n^{-\gamma_l(1-o(1))\frac{\epsilon^2}{2+\epsilon}}
\end{aligned}$$

when $i > 2$, we have

$$\begin{aligned}
&\Pr\left(\mathrm{Bi}(n, 1-(1-p)^{d_{i-1}^+}) \geq d_i^+\right)\\
&\leq \exp\left(-\frac{\xi^2}{2+\xi}(1-o(1))d_{i-1}^+\mu\right)
\end{aligned}$$

Note, since $i > 2$, we have $d_{i-1}^+ \geq (1+\epsilon)^2\mu^2$. Hence, we have

$$\begin{aligned}
&\leq \exp\left(-\frac{\xi^2}{2+\xi}(1-o(1)(1+\epsilon)^2\mu^3\right)\\
&\leq \exp\left(-\epsilon^2(1-o(1))(1+\epsilon)\mu\right)\\
&\leq n^{-\epsilon^2(1+\epsilon)(1-o(1)\gamma_l}
\end{aligned}$$

Therefore, we have for all $i \leq D'$,

$$\Pr\left(\mathrm{Bi}(n, 1-(1-p)^{d_{i-1}^+}) \geq d_i^+\right) \leq n^{-\gamma_l(1-o(1))\frac{\epsilon^2}{2+\epsilon}}$$

Similarly, using the Chernoff bound, we have

$$\begin{aligned}
&\Pr(\mathrm{Bi}(n-1-d_1^+ - \cdots - d_{i-1}^+, 1-(1-p)^{d_{i-1}^-}) \leq d_i^-)\\
&\leq \exp\left(-\frac{\xi'^2}{2}(n-1-d_1^+ - \cdots - d_{i-1}^+)(1-(1-p)^{d_{i-1}^-})\right)
\end{aligned}$$

where

$$(1-\xi')(n-1-d_1^+ - \cdots - d_{i-1}^+)(1-(1-p)^{d_{i-1}^-}) = d_i^-$$

Therefore,

$$\xi' = 1 - \frac{d_i^-}{(1-o(1))d_{i-1}^-\delta} \geq \begin{cases} (1-\delta)\epsilon & \text{if } j = 1, 2,\\ (1-\delta)\frac{\epsilon}{\mu} & \text{if } j = 3, \cdots, D'. \end{cases}$$

for any fixed $\delta \in (0, 1)$ when n is sufficiently large. Therefore, when $i = 1, 2$ we have

$$\begin{aligned}
&\Pr(\mathrm{Bi}(n - 1 - d_1^+ - \cdots - d_{i-1}^+, 1 - (1 - p)^{d_{i-1}^-}) \\
&\leq \exp\left(-\frac{\xi'^2}{2}(1 - o(1)d_{i-1}^-\mu\right) \\
&\leq \exp\left(-\frac{\xi'^2}{2}(1 - o(1)\mu\right) \\
&\leq \exp\left(-\frac{\xi'^2}{2}(1 - o(1)\gamma_l \log n\right) \\
&\leq n^{-\gamma_l(1-o(1))\frac{(1-\delta)^2\epsilon^2}{2}}
\end{aligned}$$

when $i > 2$, we have

$$\begin{aligned}
&\Pr(\mathrm{Bi}(n - 1 - d_1^+ - \cdots - d_{i-1}^+, 1 - (1 - p)^{d_{i-1}^-}) \\
&\leq \exp\left(-\frac{\xi'^2}{2}(1 - o(1))d_{i-1}^-\mu\right)
\end{aligned}$$

Note, since $i > 2$, we have $d_{i-1}^- \geq (1 - \epsilon)^2\mu^2$. Hence, we have

$$\begin{aligned}
&\leq \exp\left(-\frac{\xi'^2}{2}(1 - o(1)(1 - \epsilon)^2\mu^3\right) \\
&\leq \exp\left(-\frac{1}{2}(1 - \delta)^2\epsilon^2(1 - o(1))(1 + \epsilon)\mu\right) \\
&\leq n^{-\frac{1}{2}(1-\delta)^2\epsilon^2(1-o(1))(1+\epsilon)\gamma_l}
\end{aligned}$$

Therefore, we have for all $i \leq D'$,

$$\begin{aligned}
&\Pr(\mathrm{Bi}(n - 1 - d_1^+ - \cdots - d_{i-1}^+, 1 - (1 - p)^{d_{i-1}^-}) \\
&\leq n^{-\frac{\gamma_l(1-o(1))(1-\delta)^2\epsilon^2}{2}}
\end{aligned}$$

Next, using union bounds, we have

$$\begin{aligned}
&\Pr(E_i(u)) \\
&\geq \Pr(E_1(u), \cdots, E_i(u)) \\
&\geq \Pr(E_1(u), \cdots, E_{i-1}(u)) \\
&- \Pr(\bar{E}_i(u)|E_1(u), \cdots, E_{i-1}(u)) \\
&\geq 1 - \sum_{j=1}^{i} \Pr(\bar{E}_j(u)|E_1(u), \cdots, E_{j-1}(u)) \\
&\geq 1 - D'n^{-\frac{\gamma_l(1-o(1))(1-\delta)^2\epsilon^2}{2}} - D'n^{-\gamma_l(1-o(1))\frac{\epsilon^2}{2+\epsilon}} \\
&\geq 1 - D'n^{-\frac{\gamma_l\epsilon^2}{2+\epsilon}}
\end{aligned}$$

for sufficiently large n. □

Next, we consider the upper bound of the diameter.

For any arbitrary nodes u, v, note that

$$\begin{aligned}&\Pr(d^g_{uv} > 2D' + 1 | \Gamma_1(u), \cdots, \Gamma_{D'}(u), \Gamma_1(v), \cdots, \Gamma_{D'}(v))\\ &\leq (1-p)^{d_{D'}(u) d_{D'}(v)}\end{aligned}$$

Note if their D' neighborhood has non-empty intersection, we have $d^g_{uv} \leq 2D'$. Therefore, we obtain that

$$\Pr(d^g_{uv} > 2D' + 1) \leq \Pr(\bar{E}_{D'}(u)) + \Pr(\bar{E}_{D'}(v)) + (1-p)^{(d^-_{D'})^2}$$

The last term is evaluated as follows:

$$\begin{aligned}&(1-p)^{(d^-_{D'})^2}\\ &\leq \exp(-p(d^-_{D'})^2)\\ &= \exp\left(-p\left((1-\epsilon)^2(1-\frac{\epsilon}{\mu})^{D'-2}\mu^{D'}\right)^2\right)\\ &= \exp\left(-p\left(\frac{1-\epsilon}{1-\frac{\epsilon}{\mu}}\right)^4(\mu-\epsilon)^{2D'}\right)\\ &\leq \exp\left(-p\left(\frac{1-\epsilon}{1-\frac{\epsilon}{\mu}}\right)^4(\mu-\epsilon)^{\log n/\log\mu}\right)\\ &= \exp\left(-pn^{\frac{\log(\mu-\epsilon)}{\log\mu}}\left(\frac{1-\epsilon}{1-\frac{\epsilon}{\mu}}\right)^4\right)\\ &= \exp\left(-pn^{1+\frac{\log(1-\epsilon/\mu)}{\log\mu}}\left(\frac{1-\epsilon}{1-\frac{\epsilon}{\mu}}\right)^4\right)\\ &= \exp\left(-\mu e^{\log n\frac{\log(1-\epsilon/\mu)}{\log\mu}}\left(\frac{1-\epsilon}{1-\frac{\epsilon}{\mu}}\right)^4\right)\\ &\leq \exp\left(-\mu(1-\frac{\epsilon}{\mu})^{-2}(1-\epsilon)^4\right)\\ &\leq \exp\left(-\mu(1-\epsilon)^4\right)\\ &\leq n^{-\gamma_l(1-\epsilon)^4}\end{aligned}$$

Therefore, we have

$$\Pr(d^g_{uv} > 2D' + 1) \leq n^{-\gamma_l(1-\epsilon)^4} + 2D' n^{-\frac{\gamma_l\epsilon^2}{2+\epsilon}}$$

Finally, we have

$$\begin{aligned}\Pr(\text{Diamter} > 2D'+1) &\leq \sum_{u \neq v} \Pr(d_{uv}^g > 2D'+1) \\ &\leq n^2 \times \left(n^{-\gamma_l(1-\epsilon)^4} + 2D' n^{-\frac{\gamma_l \epsilon^2}{2+\epsilon}} \right)\end{aligned}$$

Therefore, we have

$$\gamma_l > \max \left(\frac{2}{(1-\epsilon)^4}, \frac{2(2+\epsilon)}{\epsilon^2} \right)$$

Note $\frac{2}{(1-\epsilon)^4} - \frac{2(2+\epsilon)}{\epsilon^2}$ is a increasing function for $\epsilon(0,1)$ and $\max\left(\frac{2}{(1-\epsilon)^4}, \frac{2(2+\epsilon)}{\epsilon^2}\right) \geq 23.35$. The optimal value is when $\epsilon = 0.459$. Therefore, when

$$\gamma_l > 24.$$

we have

$$\begin{aligned}\Pr(\text{Diamter} > 2D'+1) &\leq \sum_{u \neq v} \Pr(d_{uv}^g > 2D'+1) \\ &\leq n^{-\delta_2} + 2D' n^{-\delta_1}\end{aligned}$$

where δ_1 and δ_2 is fixed positive constant. Note $D' \leq \log n$. We have

$$\lim_n \Pr(\text{Diamter} > 2D'+1) = 0.$$

Note $2\lceil x/2 \rceil \leq \lceil x \rceil$. Hence

$$2D' + 1 \leq D + 2$$

Hence, we have

$$\lim_n \Pr(\text{Diamter} \leq D+2) = 1.$$

The theorem is proved. □

A.6 Necessary Inequalities

We use the following Chernoff bounds.

Lemma 19. *Let $X_1, X_2, \cdots, X_n$ be i.i.d Poisson trials such that $\Pr(X_i) = p_i$. Let $X = \sum_{i=1}^n X_i$ and $E(X) = \mu$. For any $\delta > 0$, we have*

$$\Pr(X \geq (1+\delta)\mu) \leq \left(\frac{e^\delta}{(1+\delta)^{1+\delta}} \right)^\mu \leq \exp\left(-\frac{\delta^2 \mu}{2+\delta} \right)$$

and for $\delta \in (0,1)$

$$\Pr(X \leq (1-\delta)\mu) \leq \exp\left(-\frac{\delta^2 \mu}{2} \right)$$

Proof. We only need to prove $\left(\frac{e^\delta}{(1+\delta)^{1+\delta}}\right)^\mu \leq \exp\left(-\frac{\delta^2\mu}{2+\delta}\right)$. All other bounds are proved in (Mitzenmacher and Upfal, 2005). We need to show

$$\begin{aligned}
\left(\frac{e^\delta}{(1+\delta)^{1+\delta}}\right)^\mu &\leq \exp\left(-\frac{\delta^2\mu}{2+\delta}\right) \\
\mu\left(\delta-(1+\delta)\log(1+\delta)\right) &\leq -\frac{\delta^2\mu}{2+\delta} \\
(2+\delta)\delta-(1+\delta)(2+\delta)\log(1+\delta)+\delta^2 &\leq 0 \\
(2\delta-(2+\delta)\log(1+\delta))(1+\delta) &\leq 0 \\
2\delta-(2+\delta)\log(1+\delta) &\leq 0
\end{aligned}$$

Denote by $f(\delta)=2\delta-(2+\delta)\log(1+\delta)$. We have

$$f'(\delta)=2-\log(1+\delta)-\frac{2+\delta}{1+\delta}=1-\log(1+\delta)-\frac{1}{1+\delta}$$

$$f''(\delta)=-\frac{1}{1+\delta}+\frac{1}{(1+\delta)^2}=\frac{1}{1+\delta}(\frac{1}{1+\delta}-1)\leq 0$$

Hence, $f'(\delta)\leq f'(0)=0$. Therefore, we have

$$f(\delta)\leq f(0)=0$$

Hence we prove the lemma. □

We need the following bounds

Lemma 20. *When $x>0$, we have*

$$1-x\leq e^{-x}$$

and when $x\in(0,\frac{\log 2}{2})$,

$$1-x\geq e^{-2x}.$$

Proof. Let $f_1(x)=1-x-e^{-x}$. We have

$$f_1'(x)=-1+e^{-x}<0$$

when $x>0$. Hence, $f_1(x)\leq f_1(0)=0$. Therefore, we have $1-x\leq e^{-x}$.

Let $f_2(x)=1-x-e^{-2x}$. We have

$$f_2'(x)=-1+2e^{-2x}.$$

When $x<\frac{\log 2}{2}$, we have $f_2'(x)>0$. Therefore $f_2(x)\geq f_2(0)=0$. We have $1-x\geq e^{-2x}$. □

We obtain the following bound using the similar proof procedures. $\forall x>0, 1-\frac{1}{x}\leq \log(x)\leq x-1$.

Lemma 21. *For $x \geq 2$ and integer $n \geq 0$ we have*

$$x^n \leq \sum_{i=0}^{n} x^i \leq 2x^n$$

Proof.

$$\begin{aligned}
\sum_{i=0}^{n} x^n - 2x^n &= \frac{x^{n+1}-1}{x-1} - 2x^n \\
&= \frac{x^{n+1}-1-2x^{n+1}+2x^n}{x-1} \\
&= \frac{2x^n - 1 - x^{n+1}}{x-1} \\
&= \frac{x^n(2-\frac{1}{x^n}-x)}{x-1} \\
&\leq \frac{x^n(2-\frac{1}{x^n}-2)}{x-1} \\
&\leq 0
\end{aligned}$$

Hence, we obtain the inequality in the lemma. □

APPENDIX B

PROOFS OF CHAPTER 3

B.1 Proof of Lemma 8

Define $x_{k,k-1} = t_k - t_{k-1}$, so the cost C can be written as

$$C(\mathbf{x}) = \sum_{k=2}^{n}(t_k - t_{k-1} - \eta)^2 = \sum_{k=2}^{n}(x_{k,k-1} - \eta)^2.$$

The cost minimization problem can be written as

$$\begin{aligned} & \min C(\mathbf{x}) = \textstyle\sum_{k=2}^{n}(x_{k,k-1} - \eta)^2 && \text{(B.1)} \\ \text{subject to:} \quad & \textstyle\sum_{k=2}^{n} x_{k,k-1} = t_n - t_1 && \text{(B.2)} \\ & x_{k,k-1} \geq 0. && \text{(B.3)} \end{aligned}$$

Note that $C(\mathbf{x})$ is a convex function in $\mathbf{x}$. By verifying the KKT condition (Boyd and Vandenberghe, 2004), it can be shown that the optimal solution to the problem above is $x_{k,k-1} = \frac{\tau_n - \tau_1}{n-1}$, which implies $t_k = \tau_1 + (k-1)\frac{\tau_n - \tau_1}{n-1}$.

B.2 Proof of Theorem 7

Assume all nodes in the network are infected nodes and the infection time of two nodes (say Node v and Node w) are observed. Without loss of generality, assume $\tau_v < \tau_w$. Furthermore, assume the graph is undirected (i.e., all edges are bidirectional) and

$$|\tau_v - \tau_w| \geq \eta(|\mathcal{I}| - 1).$$

We will prove the theorem by showing that computing the cost of Node v is related to the longest path problem between Nodes v and w.

To compute $C(v)$, we consider those spreading trees rooted at Node v. Given a spreading tree $\mathcal{P} = (T, \mathbf{t})$ rooted at Node v, denote by $\mathcal{Q}(v, w)$ the set of edges on the path from Node v to Node w. The cost of the spreading tree can be written as

$$\begin{aligned} C(\mathcal{P}) = & \sum_{(h,u)\in\mathcal{E}(T)\backslash\mathcal{Q}(v,w)} (t_u - t_h - \eta)^2 && \text{(B.4)} \\ & + \sum_{(h,u)\in\mathcal{Q}(v,w)} (t_u - t_h - \eta)^2 && \text{(B.5)} \end{aligned}$$

Recall that only the infection time of Nodes v and w are known. Furthermore, Nodes v and w will not both appear on a path in $T\backslash\mathcal{Q}(v, w)$. Therefore, by choosing $\tau_u - \tau_h = \eta$ for each $(h, u) \in \mathcal{E}(T)\backslash\mathcal{Q}(v, w)$, we have

$$(B.4) = 0.$$

Next applying Lemma 8, we obtain that

$$(B.5) \quad \geq \quad |\mathcal{Q}(v, w)| \left(\frac{\tau_w - \tau_v}{|\mathcal{Q}(v, w)|} - \eta\right)^2, \tag{B.6}$$

where the equality is achieved by assigning the timestamps according to Lemma 8.

For fixed $|\tau_w - \tau_v|$ and η, we have

$$\begin{aligned}\frac{\partial(B.6)}{\partial|\mathcal{Q}(v,w)|} &= \eta^2 - \left(\frac{\tau_w - \tau_v}{|\mathcal{Q}(v,w)|}\right)^2 \\ &<_{(a)} \eta^2 - \left(\frac{\eta(|\mathcal{I}|-1)}{|\mathcal{Q}(v,w)|}\right)^2 \\ &<_{(b)} \eta^2 - \left(\frac{\eta(|\mathcal{I}|-1)}{(|\mathcal{I}|-1)}\right)^2 = 0,\end{aligned}$$

where inequality (a) holds because of the assumption $\tau_w - \tau_v > \eta(|\mathcal{I}|-1)$ and inequality (b) is due to $|\mathcal{Q}(v,w)| \leq |\mathcal{I}|-1$. So $(B.6)$ is a decreasing function of $|\mathcal{Q}(v,w)|$ (the length of the path).

Let ϕ denote the length of the longest path between v and w. Given the longest path between v and w, we can construct a spreading tree $\mathcal{P}^*$ by generating T^* using the breadth-first search starting from the longest path and assigning timestamps $\mathbf{t}^*$ as mentioned above. Then,

$$C(v) = C(\mathcal{P}^*) = \min_{\mathcal{P}_v \in \mathcal{L}(\mathcal{I},\boldsymbol{\tau})} C(\mathcal{P}_v) = \phi\left(\frac{\tau_w - \tau_v}{\phi} - \eta\right)^2. \tag{B.7}$$

Therefore, the algorithm that computes $C(v)$ can be used to find the longest path between Nodes v and w. Since the longest path problem is NP-hard (Garey and Johnson, 1979), the calculation of $C(v)$ must also be NP-hard.

B.3 Proof of Theorem 9

Note that the complexity of the modified breadth first search is $O(|\mathcal{E}(g_i)|)$ since each edge in the subgraph formed by the infected nodes only needs to be considered once. We next analyze the complexity of EIF:

- Step 1: The complexity of computing the paths from an infected node to all other infected nodes is $O(|\mathcal{E}(g_i)|)$. Given $|\iota|$ infected nodes with timestamps, the computational complexity of Step 1 is $O(|\iota||\mathcal{E}(g_i)|)$.
- Step 2: The complexity of sorting a list of size $|\iota|$ is $O(|\iota|\log(|\iota|))$.
- Steps 3 and 4: To construct the spreading tree for a given node, $|\iota|$ infected nodes need to be attached in Steps 3 and 4. Each attachment requires the construction of a modified breadth-first tree, which has complexity $O(|\mathcal{E}(g_i)|)$. So the overall computational complexity of Steps 3 and 4 is $O(|\iota||\mathcal{E}(g_i)|)$.
- Step 5: The breadth-first search algorithm is needed to complete the spreading tree, which has complexity $O(|\mathcal{E}(g_i)|)$.

From the discussion above, we can conclude that the computational complexity of constructing the spreading tree from a given node and calculating the associated cost is $O(|\iota||\mathcal{E}(g_i)|)$. CR (or TR) repeats EIF for each infected node, with complexity $O(|\iota||\mathcal{I}||\mathcal{E}(g_i)|)$, and then sort the infected nodes, with complexity $O(|\mathcal{I}|\log|\mathcal{I}|)$. Therefore, the overall complexity of CR (or TR) is $O(|\iota||\mathcal{I}||\mathcal{E}(g_i)|)$.