This paper addresses the problem of non-intrusive traffic analysis of anonymous Mobile Ad Hoc Networks (MANETs). The goal is to trace down the destination of a certain source node with the minimum dependency on the underlying network infrastructure. Specifically, it is assumed that nodes in the network do not participate in tracing, the eavesdropper is not allowed to tamper traffic, and both the payload and the packet headers are encrypted at every hop. To deal with these constraints, a timing-based approach is taken, which identifies potential destinations by estimating end-to-end flow rates based on node transmission activities. Furthermore, it is shown that as routes change due to node movement, the tracing method quickly converges to the true destination by exploring topology diversity. The proposed method is proved to be consistent under mild conditions and shown to perform well even in the presence of intersecting flows and chaff traffic.

**Keywords:** Traffic analysis, Privacy and anonymity, MANETs.

1. INTRODUCTION

Mobile Ad Hoc Networks often support sensitive applications such as military missions. Since these applications are usually deployed in hostile environments, it is desirable to protect the communicating nodes by hiding their identities through anonymous networking (e.g., [1]). For example, consider an anonymous MANET illustrated in Fig. 1, where in addition to data encryption, the Medium Access Control (MAC) and the network layer packet headers are also encrypted at every hop by a distinct encryption function shared by neighbors.

While the above techniques protect traffic content, it does not prevent eavesdropping on traffic activity. In particular, the eavesdropper can monitor transmission activities in timing by deploying traffic sensors (e.g., energy detectors) in the vicinity of the nodes of interest. Due to the open wireless media, such eavesdropping is difficult to prevent. Although the eavesdropper’s observations are naturally perturbed by delays and multiplexed traffic, and relay nodes may introduce extra dummy packets (both multiplexed packets and dummy packets are referred to as chaff noise to the flow of interest), it has been shown that it is still possible to detect the information flow reliably (see [2,3]). Moreover, given the transmission activities, a global eavesdropper (assume centralized eavesdropping) can easily construct a graph showing all the one-hop communications, referred to as a communication graph, by matching the transmission timestamps of data packets and their acknowledgements. Although a single communication graph does not reveal end-to-end communications, the graph is constantly changing in MANETs due to mobility, reducing the uncertainty over time. Such observations may add up to reveal valuable information about the underlying networks.

1.1. Related Work

The problem of anonymous networking aims at protecting user privacy against eavesdroppers. Specifically, anonymous routing protocols such as [4] have been proposed to protect node identities, source-destination relationships, and even routes from malicious eavesdroppers by mechanisms such as dynamic pseudonyms, per-hop encryption, mixing, timing perturbation, and inserting dummies. In addition, methods derived from Chaum’s Mix networks [5] have been proposed to prevent traffic analysis by mixing packets at intermediate nodes. The focus, however, has been on network layer anonymity, e.g., see [1,6] and references therein, while very little attention has been given to physical layer behavior. In [7], Venkitasubramaniam et al. gave a first rigorous trade-off between achievable rates and anonymity from an information-theoretic perspective, which forms the dual of the current work.

A dual problem of anonymous networking is the detection of stepping-stone attacks ([8]), which aims at tracing back an intrusion path formed by application-layer connections through compromised hosts to locate the original attacker. Existing work...
on stepping-stone detection only deals with one-hop traceback by detecting pairs of relayed connections, and most existing detectors are not robust against artificially inserted dummy packets; see [9] and references therein. He and Tong in [10] proposed the first timing-based detector that is consistent even if chaff noise grows proportionally to the total traffic size. In [3], they generalized the detector to handle various types of flows and derived fundamental limits on flow detectability. These techniques, however, do not solve our problem of end-to-end tracing because in ad hoc networks every relay node may as well be the sources or destinations of other flows, and thus a connection may be correlated with multiple adjacent connections, making hop-by-hop tracing inapplicable. Moreover, the flow detector in [3] is based on the assumption that connections either form an information flow or are jointly independent, and cannot deal with partial flows.

Tracing has also been studied as a feature in network design, where multiple mechanisms have been proposed to provide internal support of tracing, e.g., IP traceback [11] and fault tracing [12]. These techniques are intrusive in that they need insider participation and therefore fall outside the scope of our work.

1.2. Summary of Results and Organization

We consider the problem of tracing the destination of a certain source node in anonymous MANETs. By developing a method which traces destinations without using traffic content or insider support, we demonstrate the limitation of existing anonymous networking protocols in MANETs.

Our tracing method has two components: intersection and traffic analysis. The intersection component identifies candidate destinations as the nodes that are always reachable from the source throughout the observation. The traffic analysis component further reduces uncertainty by analyzing node transmission activities. Specifically, we develop an algorithm that can find out the set of nodes which partition the network into two parts—one part to which the source can communicate at sufficient rate and the other to which it cannot—and use these nodes to estimate the destination. The partition is done by estimating flow rates through packet matching, under the assumption that delays are bounded at each relay node. The estimated destination is proved to converge to the actual one if either the source rate or the mobility is sufficiently high. Moreover, simulations show that the proposed method is reasonably accurate even if all the relay nodes generate their own flows as well as injecting chaff noise. We point out that since our model of eavesdropper is rather conservative, our results provide lower bounds for more powerful eavesdroppers.

The rest of the paper is organized as follows. Section 2 defines the problem. Section 3 presents the tracing algorithm, which is then analyzed theoretically in Section 4 and by simulations in Section 5. Section 6 discusses the issues of generalization, and then Section 7 concludes the paper. Proofs of the analysis are contained in Section 8.

2. PROBLEM FORMULATION

2.1. Notation

We use the convention that uppercase letters denote random variables, lowercase letters realizations, boldface letters vectors, and plain letters scalars. Given a realization of a point process \( s \), we use \( \mathcal{S} \) to denote the set of elements in this realization. Given two sequences \((a_1, a_2, \ldots)\) and \((b_1, b_2, \ldots)\), their superposition \((a_k)_{k=1}^{\infty} \odot (b_k)_{k=1}^{\infty}\) is defined as \((c_k)_{k=1}^{\infty}\) such that \(c_1 \leq c_2 \leq \ldots\), and \((a_k)_{k=1}^{\infty} \cup (b_k)_{k=1}^{\infty} = (c_k)_{k=1}^{\infty}\). Given a directed graph \( G = (V, E) \), where \( V \) is the set of nodes and \( E \) the set of links, \( N^- (j) \) and \( N^+ (j) \) denote the sets of immediate predecessors and successors of node \( j \).

2.2. Definitions

As illustrated in Fig. 2, in a communication graph, link \((u, v)\) means that node \( u \) is talking to node \( v \). After filtering out acknowledgement packets, we can partition the transmissions of each node into separate links; let a point process \( S_{u,v} \) denote the transmission of data packets on link \((u, v)\), i.e.,

\[
S_{u,v} = (S_{u,v}(1), S_{u,v}(2), S_{u,v}(3), \ldots),
\]

where \( S_{u,v}(k) \) is the \( k \)th transmission of data packets on the corresponding links.

If a path is carrying an information flow, then the transmission activities of links on this path will demonstrate certain correlation, defined as follows.

**Definition** There exists a \((0 \rightarrow n)\) flow on a path \((0, 1, 2, \ldots, n)\) if the sequence of processes \((S_{i-1,i})_{i=1}^{n}\) can be decomposed into subprocesses \((F_{i-1,i})_{i=1}^{n}\) and \((W_{i-1,i})_{i=1}^{n}\):

\[
S_{i-1,i} = F_{i-1,i} \odot W_{i-1,i}, \quad i = 1, \ldots, n,
\]

where \((F_{i-1,i})_{i=1}^{n}\), called an information flow, satisfies that for every realization, there exist bijections \(g_i : F_{i-1,i} \rightarrow F_{i,i+1} \quad (i = 1, \ldots, n-1)\) such that \(g_i(s) - s \in [0, \Delta] \) for all \( s \in F_{i-1,i}\), and \((W_{i-1,i})_{i=1}^{n}\) is called chaff noise.

As illustrated in Fig. 3, the information flow \((F_{0,1}, \ldots, F_{n-1,n})\) models the transmissions of information packets along the path. Chaff noise \((W_{i-1,i})_{i=1}^{n}\) models transmissions not belonging to the flow (e.g., dummy packets, multiplexed flows). The bijection \(g_i\) is a mapping between the timestamps of information packets at adjacent nodes. The condition that \(g_i\) is a bijection ensures packet-conservation. The condition \(g_i(s) - s \in [0, \Delta]\) implies causality as well as a maximum delay\(^3\) \(\Delta\). Assume that \(\Delta\) is known.

\(^2\)Assume no simultaneous transmissions almost surely (a.s.).

\(^3\)This condition was proposed by Donoho et al. in [2].
2.3. Problem Statement

Suppose we want to trace down the destination of a source node 0. Given M reports, each report contains a connected, directed graph \( G_i \) (i = 1, ..., M) denoting the connected component containing 0 in the i-th communication graph, and a set of transmission activities \( S'_i = \{ S_{u,v} : (u, v) \in E_i \} \), where \( S_{u,v} = (S_{u,v}^1(1), S_{u,v}^2(2), ...) \) denotes the transmission timestamps on link \((u, v)\) in \( G_i \) (as defined in (1)). Assume point-to-point communication and single-path routing. Our problem is the following: find a node \( \theta \) such that:

1. there exists a \((0 \rightarrow \theta)\) flow in each \( G_i \) (i = 1, ..., M), i.e., \( \exists \) a path \((0, j_1, \ldots, j_k, \theta)\) in \( G_i \) s.t. there is a \((0 \rightarrow \theta)\) flow on this path;
2. if \( j \) is the successor of \( \theta \) (i.e., \( \exists \) a path from \( \theta \) to \( j \)) in every \( G_i \), then \( \exists i_0 \in \{1, \ldots, M\} \) s.t. there is no \((0 \rightarrow j)\) flow in \( G_{i_0} \);
3. \( \exists i_1 \in \{1, \ldots, M\} \) s.t. there is no \((j \rightarrow \theta)\) flow through 0 in \( G_{i_1} \) for all \( j \) being an immediate predecessor of 0 (i.e., \( j \in N^{-}(0) \)).

Condition (1) says that 0 should always talk to \( \theta \), condition (2) says that \( \theta \) must be the final destination (i.e., it cannot be the relay for some other node), and condition (3) says that \( \theta \) must be the true destination of 0 (i.e., it cannot be the destination of flows relayed through 0). Assume that \( \theta \) is well-defined, i.e., node 0 does not change destination throughout the reports. Discussions on generalizing these assumptions are presented later in Section 6.

3. TRACING ALGORITHMS

In this section, we present tracing algorithms based on given graphs and transmission activities. There are two features that can help detection: topology change and link correlation. Accordingly, our tracing algorithms have two components: intersection and traffic analysis.

3.1. Tracing on a Single Graph

If the topology does not change during our observation, then the problem is reduced to pure traffic analysis. The basic idea here is that transmission activities on links of the same flow are likely to be more similar than those from different flows. Thus, how well these activities match can reveal the relationship between links.

Specifically, it is known that under the flow model in Definition 2.2, First-In-First-Out (FIFO) scheme with maximum delay \( \Delta \) matches the most packets for any given pair of transmission sequences\(^4\). For a multi-hop path \((0, 1, \ldots, n)\), the maximum flow rate can be estimated by repeatedly applying FIFO to the measured transmission activities\(^5\) \((s_{i-1,i})_{i=1}^{n-1}\). Specifically, let \( s'_2 = \text{FIFO}(s_1, s_2) \) denote the subsequence of matched packets in \( s_2 \), the maximum (empirical) flow rate is the rate of \( s_{n-1,n} \), computed by the following iterations \((s'_{0,1} = s_{0,1})\):

\[
s'_{i,i+1} = \text{FIFO}(s'_{i-1,i}, s_{i,i+1}), \quad i = 1, \ldots, n-1. \tag{3}
\]

Our tracing algorithm is built on FIFO. For the ease of presentation, we introduce the following definitions.

**Definition** Given a graph \( G = (V, E) \) and a set of transmission activities \( S \) on \( G \), define the following:

1. Supported rate: A path \((0, 1, \ldots, n)\) is said to support rate \( R \) if the \((0 \rightarrow n)\) flow generated by repeatedly applying FIFO to \((s_{i-1,i})_{i=1}^{n-1}\) (as in (3)) has a rate at least \( R \).
2. Rate-\( R \) sourced graph (R-SG): The R-SG of \( j \) is the directed graph formed by all the paths starting from node \( j \) that support rate \( R \).

In the R-SG of node \( j \), only node \( j \) has no incoming links (i.e., \( j \) is the source) and hence the name “sourced graph”. The R-SG of \( j \) partitions the nodes into two sets: one set of nodes to which \( j \) can communicate at rate \( R \) and the other to which it cannot. For example, as illustrated in Fig. 4, the R-SG of 0 contains paths \((0, 1, 2)\), \((0, 4, 1)\), and \((0, 3)\), for which nodes 1, 2, and 3 are destinations. Destinations of the paths forming an R-SG are referred to as the destinations of the R-SG. Note that an R-SG is not necessarily acyclic, and its destinations are not always sinks.

**Fig. 4.** R-SG example: the R-SG of \(-1\) contains paths \((-1, 0, 4, 1)\) and \((-1, 0, 3)\); the R-SG of 0 contains paths \((0, 1, 2)\), \((0, 4, 1)\), and \((0, 3)\). Node \(-1\): a super node representing the immediate predecessors of node 0.

Intuitively, R-SG shows the dominant flow patterns. For example, in Fig. 4, node 0 sends a flow to 2 and also relays flows to 1 and 3. If the flow rates are sufficiently large, or the delay bound \( \Delta \) is sufficiently small, then as we estimate the rates from node

\(^4\)This scheme is the same as “Bounded-Greedy-Match” in [13], which also proved its optimality.

\(^5\)He and Tong [10] gave an algorithm which may find more matchings than repeatedly applying FIFO. That algorithm, however, requires all the transmission activities to be known at the source a priori, whereas in this paper, each node only observes its own incoming and outgoing traffic.
0 (by FIFO), rates to nodes on the actual flow paths (i.e., nodes 1, . . ., 4) will be significantly larger than those to the other nodes because paths to the other nodes will contain links on different flows which usually have very different transmission activities. Thus, for a proper R, the R-SG of node 0 will separate the flows coming out of 0 from the rest of the graph, and its destinations will provide candidate destinations of 0 (e.g., {1, 2, 3} in Fig. 4). Moreover, since node 0 may relay other flows, we also need to filter out destinations of the relayed flows (t = 1, 3) by repeating the above for predecessors of 0. For simplicity, represent the predecessors of 0 by a super node −1, and the total incoming traffic \( \sum_{j \in N^- (0)} s_{j, 0} \) by \( s_{-1, 0} \).

Based on the above idea, we develop an algorithm called “Trace-Destination” (TD) as shown in Algorithm 1. Given a graph \( G \) and a set of measured transmission activities \( s \), TD estimates the destination of node 0 by computing the destinations of the \( \tau \)-SG of −1, denoted by \( \Theta_{-1} \) (line 2), and those of 0, denoted by \( \Theta_0 \) (lines 3–6). Here \( \tau \geq 0 \) is a design parameter. The main computation is done by an auxiliary algorithm called “Recursive-Tracing” (RT) shown in Algorithm 2, which recursively traces down the destinations of incoming traffic \( s_j \) of node \( j \). For each outgoing link of node \( j \) (line 2), RT estimates the rate through this link by FIFO (line 3) and traces down the link if the rate is above \( \tau \) (lines 4–5). If node \( j \) has no outgoing link, or none of them support rate \( \tau \), then \( j \) itself is a destination (line 7). The result of RT is the destinations of the \( \tau \)-SG starting from \( s_j \). Finally, TD returns nodes that are destinations in the \( \tau \)-SG of 0 but not −1 as potential destinations (line 10), or simply that of 0 if it is unique (line 8). The estimated destination \( \hat{\Theta} \) will be randomly picked out of the potential destinations found by TD.

**Algorithm 1 TRACE DESTINATION (TD)**

**Ensure:** Return set of nodes which are destinations in the \( \tau \)-SG of node 0 but not its predecessors.

1. compute \( s_{-1, 0} \) by merging \( s_{j, 0} \) for all \( j \in N^- (0) \)
2. \( \Theta_{-1} = RT(s_{-1, 0}, 0, G, s, \Delta) \)
3. initialize \( \Theta_0 \) to the empty set
4. for each \( j \in N^+ (0) \) do
5. \( \Theta_0 = \Theta_0 \cup RT(s_{0, j}, j, G, s, \Delta) \)
6. if \( |\Theta_0| = 1 \) then
7. return \( \Theta_0 \)
8. else
9. return \( \Theta_0 \setminus \Theta_{-1} \)
10. \n
3.2. Tracing across Multiple Graphs

As nodes move, we may observe more than one communication graph. Given \( G_i = (V_i, E_i) \) (i = 1, . . ., M), let \( V'_i \) denote the set of nodes reachable from 0. A straightforward tracing method is to take the intersection \( \bigcap_{i=1}^M V'_i \) and randomly pick one node in this intersection as the estimated destination. Refer to this method as the simple intersection method (SIM) and the estimator \( \hat{\theta}_{SIM} \). We aim at achieving faster convergence by combining the intersection method with traffic analysis.

**Algorithm 2 RECURSIVE TRACING (RT)**

**Require:** Sequence \( s_j \) is the incoming traffic of node \( j \).

**Ensure:** Return set of nodes which are destinations in the \( \tau \)-SG of node \( j \) starting from \( s_j \).

1. initialize \( \Theta \) to the empty set
2. for each \( k \in N^+ (j) \) do
3. \( s'_{j, k} = \text{FIFO}(s_j, s_{j, k}, \Delta) \)
4. if the rate of \( s'_{j, k} \geq \tau \) then
5. \( \Theta = \Theta \cup RT(s'_{j, k}, j, G, s, \Delta) \)
6. if \( N^+ (j) \) is empty or all the \( s'_{j, k} \)'s have rates < \( \tau \) then
7. \( \Theta = \{j\} \)
8. return \( \Theta \)

Specifically, we use a majority rule to combine the detection results of individual graphs. Given the results \( \Theta^M \equiv (\Theta_1, \ldots, \Theta_M) \) of TD on \( M \) communication graphs, the final estimate \( \hat{\theta} \) is the node in the intersection \( \bigcap_{i=1}^M V'_i \) with the highest frequency, i.e.,

\[ \hat{\theta}_{TD} = \arg \max_{j \in \bigcap_{i=1}^M V'_i} \pi(j; \Theta_i^M), \]

where \( \pi(j; \Theta_i^M) \) counts the number of times that \( j \) appears in \( \Theta_i^M \) (\( \hat{\theta} \) is picked randomly in case of tie).

**4. PERFORMANCE ANALYSIS**

The tracing algorithms proposed in Section 3 have certain limitations because transmission activities typically fluctuate over time, and not all the information flows are detectable. If, however, the flow of interest lasts sufficiently long, then accurate tracing can be achieved. Specifically, let \( \lambda \) denote link capacity and \( R \) the rate of the flow of interest. We have the following results.

**Theorem 4.1** Suppose the transmission activities on each link form a Poisson process of rate bounded by \( \lambda \). Then as time increases (for fixed \( M \)), a necessary condition for consistent estimation* of \( \theta \) is that \( R > \lambda^2 \Delta / (1 + \lambda \Delta) \). Moreover, if \( R > \lambda / 2 + \lambda^2 \Delta / [2(1 + \lambda \Delta)] \) and \( \lambda / 2 + \lambda^2 \Delta / [2(1 + \lambda \Delta)] < \tau < R \), then \( \theta_{TD} \) is consistent.

**Proof:** See Appendix.

Theorem 4.1 gives two bounds on the threshold rate of trackable flows: a lower bound below which no estimator guarantees consistency and an upper bound above which the proposed estimator is consistent. The gap between these bounds decreases as we relax the delay constraint, and both eventually go to \( \lambda \) when the constraint is removed. The idea behind is that the estimated rate will drop considerably as we deviate from the actual flow path, and thus if the actual flow rate is higher, we can correctly detect it.

*For example, He et al. in [14] derived conditions for undetectable flows.

**That is, the error probability goes to zero asymptotically.**

It is implicitly assumed that different flows have independent transmission activities.
identify it by choosing a threshold $\tau$ in between. Actually, under the sufficient condition in Theorem 4.1, the error probability of the proposed method decays exponentially, as stated below.

**Theorem 4.2** Suppose that $M = 1$, and the flow of interest lasts for time $T$. Then under the sufficient conditions in Theorem 4.1, the error probability of $\theta_{\triangle}$, denoted by $P_e(M, T)$, decays exponentially with $T$, and the error exponent satisfies

$$\lim_{T \to \infty} \frac{1}{T} P_e(M, T) \geq \min \left( (\lambda - \gamma)^2 / (2R), (\lambda - \gamma)^2 / \lambda \right) \Delta \sigma(\tau),$$

where $\gamma (\tau - \lambda + R < \gamma < \lambda)$ is the solution to the equation

$$\frac{(\lambda - \gamma)^2}{\lambda} = 2(\gamma - \tau + \lambda - R) \eta \left( \frac{\gamma}{\gamma - \tau + \lambda - R} + 1 + \lambda \Delta \right)$$

for $\eta(a, b) \triangleq a \log(a/b) - (1 + a) \log((1 + a)/(1 + b)) (a, b > 0)$.

**Proof:** See Appendix.

Theorem 4.2 helps to decide the threshold $\tau$ because choosing $\tau$ to maximize $\sigma(\tau)$ in (4) may enable faster convergence of the algorithm. We point out that even though the specific results hinge on Poisson assumption, our algorithm and analysis also apply to other types of traffic.

Besides traffic analysis, intersection method also plays an important role. As $M$ increases, it is easy to see that consistent tracing can be achieved if other nodes are not always reachable from the monitored source in the communication graphs. For example, a sufficient condition is that nodes other than the destination become disconnected from the source at some time almost surely, which is easily satisfied in MANETs over large fields. Compared with SIM (see Section 3.2), the proposed method has the advantage that it enriches topology-level information with detailed information on link-level activities and therefore may achieve accurate tracing even if the intersection method fails due to low mobility.

## 5. SIMULATIONS

The performance analysis in Section 4 focuses on the asymptotic performance of pure traffic analysis, whereas in this section, we will verify the overall performance of the proposed algorithms in specific MANETs.

### 5.1. Simulation Setup

We simulateootnote{The simulation is conducted using MATLAB 2007b.} a MANET of 25 nodes in a field of 750 m $\times$ 750 m. Each node has a communication range of 250 m. Nodes move according to a slot-based random walk model, where at the beginning of each slot (of length $T$ seconds), each node randomly selects a direction in $[0, 2\pi]$ and a speed in $[0, 10/T]$ m/second and moves accordingly for the rest of the slot.

Node 0 is always communicating with $\theta$, and the other nodes randomly select destinations in each slot. We use shortest path routing, where links can be shared by different flows, and routes stay unchanged within each slot. For each flow, the source generates traffic according to a Poisson process of rate $\lambda$, which is then relayed by each relay node after adding i.i.d. uniform delays drawn from $[0, \Delta]$ (the packet order may be permuted). After generating all the flows, we pad each link with independent Poisson chaff noise to rate$^{10}$ $\lambda$.

We simulate the network for 5000 slots and independently generate 100 sets of transmission activities per slot (i.e., totally $5 \times 10^5$ Monte Carlo runs). In the simulation, $\lambda = 1$ packet/second, $\Delta = 1/18$ second, and $R = 1/9$ packet/second.

### 5.2. Simulation Results

We simulate TD and SIM (defined in Section 3.2) under various threshold, observation time, and mobility. We first simulate TD versus different threshold $\tau$ to find the best threshold. As shown in Fig. 5, for $M = 1$, if we plot the error probability of TD as a function of $\tau$, we see that the error probability decreases with $\tau$ until it stabilizes around $\tau \in [0.12, 0.14]$ and then starts to increase slowly. By this observation, we set $\tau = 0.13$ in the subsequent tests to minimize error probability.

![Fig. 5. Error probability $P_e$ vs. threshold $\tau$ (number of graphs $M = 1$, observation time per graph $T = 50, 100, 500$; $\tau^* \approx 0.13$ minimizes $P_e$.)](image)

We then simulate TD and SIM for various total observation time $MT$ to see how their performance improves as the observation time increases; see Fig. 6. In contrast to the exponential decay predicted by Theorem 4.2, we see that when flows can overlap, the error probability of TD decays subexponentially. Actually, the decay rates are approximately $O((MT)^{-0.13})$, $O((MT)^{-0.27})$, and $O((MT)^{-0.38})$ for $M = 1, 3,$ and 5, respectively. The error probability of SIM stays constant because SIM is not affected by the observation time. For small $M$ (here $M \leq 5$), both TD and SIM have lower error probabilities as $M$ increases. Moreover, compared with SIM, the error probability of TD is 50–70% lower.

Finally, we simulate TD and SIM as we vary $M$ for a fixed total observation time $MT$ to examine the role of mobility, as illustrated in Fig. 7. For fixed $MT$, larger $M$ represents higher mobility. The error probability of SIM monotonically decreases with

$^{10}$That is, if a link is shared by $k$ flows, then chaff noise will be an independent Poisson process of rate $(\lambda - kR)$ ($R$ is chosen such that $\lambda - kR \geq 0$).
the increase of $M$ as expected. It can be shown that the decay rate is approximately $O(e^{-0.043M})$. For TD, the error probability is not a monotone function of $M$. For small $M$ (here $M \leq 10$), the error probability decreases with $M$ because in this range, larger $M$ can bring more diversity without significantly affecting traffic analysis. As $M$ increases beyond a certain point (here $M \geq 20$), however, the error probability starts to increase with $M$. This is because the observation time per graph $T$ decreases as $M$ increases, resulting in less accurate traffic analysis on individual graphs. We point out that for very large $M$, TD is reduced to the intersection method, and therefore its error probability again decreases with $M$. In the middle range, the contradictory effects of the increase of $M$ (i.e., less accurate traffic analysis and smaller intersection) result in a local minimum of the error probability at some finite $M^*$ (in the simulation, $M^* \approx 20$).

6. DISCUSSIONS

We have considered timing-based tracing of communications in MANETs. Motivated as a security threat in anonymous networks, the proposed method can also be used for external network monitoring, where in contrast with existing monitoring systems, our method does not interfere with network operations or require internal support, and thus enables universal network monitoring.

8. APPENDIX

8.1. Proof of Theorem 4.1

The necessary condition directly follows from [14], which says that flows of rate below $\lambda^2\Delta/(1 + \lambda\Delta)$ are statistically indistinguishable from independent traffic. For the sufficient condition, note that if a node is not on the flow paths from the source, then its estimated rate must be no larger than $\lambda - R + \lambda^2\Delta/(1 + \lambda\Delta)$, where the extra $\lambda - R$ corresponds to the case when the node is on the flow path of some relay node. Hence, rates for nodes on the flow paths will be higher if $R > \lambda - R + \lambda^2\Delta/(1 + \lambda\Delta)$. Since the empirical rates converge almost surely (see [14]), as time increases, the empirical $\tau$-SG of 0 and $-1$ will converge a.s. to subgraphs of the communication graph formed by flow paths.
stemming from 0 and −1 if \(\lambda/2 + \lambda^2 \Delta/[2(1 + \lambda \Delta)] < \tau < R\). Moreover, because the flow paths are disjoint (since \(R > \lambda/2\)), \(\theta\) must be a destination of 0, but cannot be a destination of −1. Similarly, if another node \(j (j \neq \theta)\) is a destination of 0, it must be a destination of −1. Thus, the difference between the destinations of 0 and −1, as is computed by TD, will converge to \(\theta\).

8.2. Proof of Theorem 4.2

We start by partitioning the error event into two types: \(\theta\) is not detected by TD (i.e., \(\theta \notin \Theta_0 \setminus \Theta_{-1}\)) and \(3j \neq \theta\) that is detected. Let the empirical rate from \(j\) to \(k\) be \(\hat{R}_{j,k}\). For the actual flow, \(\hat{R}_{j,k} \sim \mathcal{N}(R, R/T)\) for large \(T\) by Central Limit Theorem (CLT). We will use the following result derived from [10]: given independent Poisson processes \(S_1, S_2\) of rate \(\lambda\), the empirical rate \(\hat{R}\) of a flow embedded into these processes by FIFO satisfies\(^{11}\)

\[
\Pr\{\hat{R} \geq x\} \leq Q\left(\frac{2(\lambda - \gamma)T}{\sqrt{2\lambda T}}\right) + \exp\left(-2(\gamma - x)T\eta\left(\frac{\gamma}{\gamma - x}, 1 + \lambda \Delta\right)\right)
\]

for large \(T\), where \(\gamma \in (x, \lambda)\). The first term in (6) is a bound obtained by CLT on the probability that the total number of packets \(K\) is less than \(2\gamma T\). The second term is a bound on \(\Pr\{\hat{R} \geq x|K \geq 2\gamma T\}\). By Theorem 4.1 in [10], it can be shown that

\[
\Pr\{\hat{R} \geq x|K = k\} \leq \exp\left(-2(\gamma - x)T\eta\left(\frac{k}{k - 2xT}, 1 + \lambda \Delta\right)\right).
\]

Replacing \(k\) by \(2\gamma T\) yields the second term.

If \(\theta\) is not detected, the reason can be: (i) TD does not reach \(\theta\) because the empirical rate \(\hat{R}_{0,\theta}\) is below \(\tau\), (ii) \(\theta\) is mistaken as a relay node because \(\hat{R}_{0,\theta} \geq \tau\) for some \(j \in N^{+}(\theta)\), or (iii) \(\theta\) is mistaken as a destination of −1. The first term has an error exponent of \((R - \tau)^2/(2R)\), and the second and the third terms both have the same error exponent as (6) for \(x = \tau - \lambda + R\), which is maximized when the two terms have equal exponents, implying the optimal value of \(\gamma\) as the solution to (5). Similar analysis holds if \(\theta\) is not the only detected node. Since the overall error probability is upper bounded by the sum of the probabilities of the above events, the dominant error exponent is \(\min((R - \tau)^2/(2R), (\lambda - \gamma)^2/\lambda)\).

9. REFERENCES


\(^{11}\)Here \(Q(\cdot)\) is the tail probability of the standard Gaussian distribution.