STARS: A Statistical Traffic Pattern Discovery System for MANETs

Yang Qin, Dijiang Huang, *Senior Member*, *IEEE*, and Bing Li, *Student Member*, *IEEE*

Abstract—Many anonymity enhancing techniques have been proposed based on packet encryption to protect the communication anonymity of mobile ad hoc networks (MANETs). However, in this paper, we show that MANETs are still vulnerable under passive statistical traffic analysis attacks. To demonstrate how to discover the communication patterns without decrypting the captured packets, we present a novel statistical traffic pattern discovery system (STARS). STARS works passively to perform traffic analysis based on statistical characteristics of captured raw traffic. STARS is capable of discovering the sources, the destinations, and the end-to-end communication relations. Empirical studies demonstrate that STARS achieves good accuracy in disclosing the hidden traffic patterns.

Index Terms—Anonymous communication, mobile ad hoc networks, statistical traffic analysis

1 Introduction

Mobile ad hoc networks (MANETs) are originally designed for military tactic environments. Communication anonymity is a critical issue in MANETs, which generally consists of the following aspects: 1) Source/destination anonymity—it is difficult to identify the sources or the destinations of the network flows. 2) End-to-end relationship anonymity—it is difficult to identify the end-to-end communication relations. To achieve anonymous MANET communications, many anonymous routing protocols such as ANODR [1], MASK [2], and OLAR [3] (see more in [4], [5], [6], [7], and [8]) have been proposed. Though a variety of anonymity enhancing techniques like onion routing [9] and mix-net [10] are utilized, these protocols mostly rely on packet encryption to hide sensitive information (e.g., nodes' identities and routing information) from the adversaries. However, passive signal detectors can still eavesdrop on the wireless channels, intercept the transmissions, and then perform traffic analysis attacks.

Over the past few decades, traffic analysis models have been widely investigated for static wired networks (e.g., [9], [10], [11], [12], [13]). For example, the simplest approach to track a message is to enumerate all possible links a message could traverse, namely, the brute force approach [11]. Recently, statistical traffic analysis attacks have attracted broad interests due to their passive nature, i.e., attackers only need to collect information and perform analysis quietly without changing the network behavior (such as injecting or modifying packets). The predecessor attacks [14], [15], [16] and disclosure attacks [17], [18], [19], [20] are two representatives. However, all these previous approaches do not work well to analyze MANET traffic because of the following three natures of MANETs: 1) *The broadcasting nature:* In wired networks, a point-to-point message transmission usually has only one possible receiver. While in wireless networks, a message is broadcasted, which can have multiple possible receivers and so incurs additional uncertainty. 2) *The ad hoc nature:* MANETs lack network infrastructure, and each mobile node can serve as both a host and a router. Thus, it is difficult to determine the role of a mobile node to be a source, a destination, or just a relay. 3) *The mobile nature:* Most of existing traffic analysis models do not take into consideration the mobility of communication peers, which make the communication relations among mobile nodes more complex.

In [21], Huang devised an evidence-based statistical traffic analysis model specially for MANETs. In this model, every captured packet is treated as an evidence supporting a point-to-point (one-hop) transmission between the sender and the receiver. A sequence of point-to-point traffic matrices is created, and then they are used to derive end-to-end (multihop) relations. This approach provides a practical attacking framework against MANETs but still leaves substantial information about the communication patterns undiscovered. First, the scheme fails to address several important constrains (e.g., maximum hop-count of a packet) when deriving the end-to-end traffic from the one-hop evidences. Second, it does not provide a method to identify the actual source and destination nodes (or to calculate the source/destination probability distribution). Moreover, it only uses a naïve accumulative traffic ratio to infer the end-to-end communication relations (e.g., the probability for node j to be the intended destination of node i is computed as the ratio of the traffic from i to j to all traffic coming out from node i), which incurs a lot of inaccuracy in the derived probability distributions.

Reusing the evidence-based model, in this paper, we propose a novel statistical traffic pattern discovery system (STARS). STARS aims to derive the *source/destination probability distribution*, i.e., the probability for each node to be a message source/destination, and the *end-to-end link probability distribution*, i.e., the probability for each pair of

• Y. Qin is with Facebook, 1601 Willow Road, Menlo Park, CA 94025. E-mail: yangqin@fb.com.
• D. Huang and B. Li are with the School of Computing, Informatics, and Decision Systems Engineering, Arizona State University, Tempe, AZ 85287. E-mail: {Dijiang.Huang, Bing.Li.4}@asu.edu.

Manuscript received 14 May 2012; revised 20 Jan. 2013; accepted 4 Aug. 2013; published online 15 Aug. 2013.
For information on obtaining reprints of this article, please send e-mail to: tdsc@computer.org, and reference IEEECS Log Number TDSC-2012-05-0089.
Digital Object Identifier no. 10.1109/TDSC.2013.33.

nodes to be an end-to-end communication pair. To achieve its goals, STARS includes two major steps: 1) Construct point-to-point traffic matrices using the time-slicing technique, and then derive the end-to-end traffic matrix with a set of traffic filtering rules; and 2) Apply a heuristic approach to identify the actual source and destination nodes, and then correlate the source nodes with their corresponding destinations.

The contribution of STARS is twofold: 1) To the best of our knowledge, STARS is the first statistical traffic analysis approach that considers the salient characteristics of MANETs: the broadcasting, ad hoc, and mobile nature; and 2) most of the previous approaches are partial attacks in the sense that they either only try to identify the source (or destination) nodes or to find out the corresponding destination (source) nodes for given particular source (destination) nodes. STARS is a complete attacking system that first identifies all source and destination nodes and then determines their relationship.

The remainder of the paper is organized as follows: Section 2 describes the related work. Section 3 presents the fundamental system models and assumptions. In Section 4, STARS is described in detail. Section 5 presents the simulations' setup, results, and analysis. In Section 6, we further discuss how to take advantage of STARS with limited attacking ability. Finally, we conclude our work and indicate future research in Section 7.

2 RELATED WORK

Traffic analysis attacks against the static wired networks (e.g., Internet) have been well investigated. The brute force attack proposed in [11] tries to track a message by enumerating all possible links a message could traverse. In node flushing attacks (*a.k.a* blending attacks, $n-1$ attacks) [10], the attacker sends a large quantity of messages to the targeted anonymous system (which is called a *mix-net*). Since most of the messages modified and reordered by the system are generated by the attacker, the attacker can track the rest a few (normal) messages. The timing attacks as proposed in [9] focus on the delay on each communication path. If the attacker can monitor the latency of each path, he can correlate the messages coming in and out of the system by analyzing their transmission latencies. The message tagging attacks (e.g., [12]) require attackers to occupy at least one node that works as a router in the communication path so that they can tag some of the forwarded messages for traffic analysis. By recognizing the tags in latter transmission hops, attackers can track the traffic flow. The watermarking attacks are actually variants of the message tagging attacks. They reveal the end-to-end communication relations by purposely introducing latency to selected packets.

Different from the attacks mentioned above, statistical traffic analysis intends to discover sensitive information from the statistical characteristics of the network traffic, for example, the traffic volume. The adversaries usually do not change the network behavior (such as injecting or modifying packets). The only thing they do is to quietly collect traffic information and perform statistical calculations. The predecessor attacks are first pointed out by Reiter and Rubin [14]. Later works such as [15] and [16] extend them to all kinds of anonymous communication systems including onion-routing [9], mix-net [10], and DC-net [22]. In a typical predecessor attack, the attackers act exactly as legitimate nodes in the network communications. They collectively maintain a single predecessor counter for each legitimate node in the system. When an attacker finds himself to be on an anonymous path to the targeted destination, he increments the shared counter for its predecessor node in this path. The counters are then used for the attackers to infer the possible source nodes of the given destination. Obviously, to launch such an attack, a large number of legitimate nodes must first be compromised and controlled by the attackers. This is usually not achievable in MANETs. Moreover, in a MANET protected by anonymity enhancing techniques, it is a difficult task itself to identify an actual destination node as the target due to the ad hoc nature. That is, destinations are indistinguishable from other nodes (e.g., relays) in a MANET. In fact, they usually act as relay nodes as well, forwarding traffic for others. The adversaries are not able to determine whether a particular node is a destination depending on whether the node sends out traffic. This is totally different from the situation in traditional infrastructural networks where the role of every node is determined. The statistical disclosure attacks as mentioned in [17], [18], [19], and [20] are similar. A statistical disclosure attack often targets a particular given source node and intends to expose its corresponding destinations. It is assumed that the packets initiated by the source are sent to several destinations with certain probability distribution. The background (covering) traffic also has certain probability distribution (usually assumed to be uniformly distributed). After a large number of observations, the attackers are able to figure out the possible destinations of the given source. Nonetheless, the statistical disclosure attacks cannot be applied to MANETs either, because the attackers cannot easily identify the actual source nodes in MANETs. Even if a source node is identified, the attacks can only be performed when the attackers know for sure when the targeted source is originating traffic and can observe the network behavior in the absence of the source. However, the attackers are prevented from being able to do so by the ad hoc nature of MANETs, i.e., they cannot tell if the source is originating traffic or just forwarding traffic as a relay.

Due to the unique characteristics of MANETs, very limited investigation has been conducted on traffic analysis in the context of MANETs. He et al. proposed a timing-based approach in [23] to trace down the potential destinations given a known source. In this approach, assuming the transmission delays are bounded at each relay node, they estimate the flow rates of communication paths using packet matching. Then based on the estimated flow rates, a set of nodes that partition the network into two parts, one part to which the source can communicate in sufficient rate and the other to which it cannot, are identified to estimate the potential destinations. In [24], Liu et al. designed a traffic inference algorithm (TIA) for MANETs based on the assumption that the difference between data frames, routing frames, and MAC control frames is visible to the passive adversaries, so that they can recognize the point-to-point traffic using the MAC control frames, identify the end-to-end flows by tracing the routing frames, and then infer the actual traffic pattern using the data frames. The TIA achieves good accuracy in traffic inference, while the mechanism is tightly tied to particular anonymous routing protocols but not a general approach. Both [23] and [24] are analytical strategies which heavily rely on the deterministic network behaviors.

3 System Models

In this section, we present the fundamental system models adopted (assumed) by STARS.

3.1 Communication Model

We assume the anonymity enhancing techniques (such as [1], [2], [3]) are used to protect the MANETs. However, these techniques are designed to different levels of anonymity. To focus on the statistical traffic analysis, we assume, based on [21], that a combination of these techniques is applied and the targeted MANET communication system is subject to the following model:

1. The PHY/MAC layer is controlled by the commonly used 802.11(a/b/g) protocol. But all MAC frames (packets) are encrypted so that the adversaries cannot decrypt them to look into the contents.
2. Padding is applied so that all MAC frames (packets) have the same size. Nobody can trace a packet according to its unique size.
3. The "virtual carrier sensing" option is disabled. The source/destination addresses in MAC and IP headers are set to a broadcasting address (i.e., all "1") or to use identifier changing techniques. In this case, adversaries are prevented from identifying point-to-point communication relations.
4. No information about the traffic patterns is disclosed from the routing layer and above.
5. Dummy traffic and dummy delay are not used due to the highly restricted resources in MANETs.

3.2 Attack Model

The attackers' goal is to discover the traffic patterns among mobile nodes. Particularly, we have the following four assumptions for attackers:

1. The adversaries are passive signal detectors, i.e., they are not actively involved in the communications. They can monitor every single packet transmitted through the network.
2. The adversary nodes are connected through an additional channel which is different from the one used by the target MANET. Therefore, the communication between adversaries will not influence the MANET communication.
3. The adversaries can locate the signal source according to certain properties (e.g., transmission power and direction) of the detected signal, by using wireless location tracking techniques [25] such as triangulation, nearest sensor, or RF fingerprinting. Note that none of these techniques can identify the source of a signal from several nodes very close to each other. Hence, this assumption actually indicates that the targeted networks are sparse in terms of the node density. In other words, any two nodes in such a network are distant from each other so that the location tracking techniques in use are able to uniquely identify the source of a wireless signal. *In the following of this paper, unless specifically denoted as "signal source" or "source of signal," the word "source" indicates the source of a network flow.*
4. The adversaries can trace the movement of each mobile node, by using cameras or other types of sensors. In this case, the signals (packets) transmitted by a node can always be associated with it even when the node moves from one spot to another.

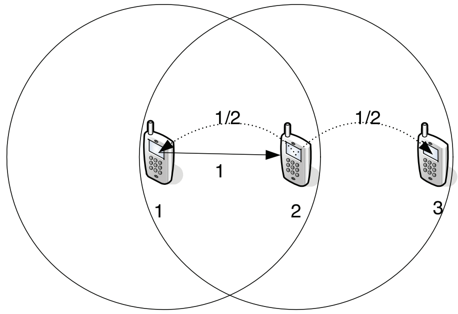


Fig. 1. A simple wireless ad hoc network.

4 Statistical Traffic Pattern Discovery System

To disclose the hidden traffic patterns in a MANET communication system, STARS includes two major steps. First, it uses the captured traffic to construct a sequence of point-to-point traffic matrices and then derives the end-to-end traffic matrix. Second, further analyzing the end-to-end traffic matrix, it calculates the probability for each node to be a source/destination (the *source/destination probability distribution*) and that for each pair of node to be an end-to-end communication link (the *end-to-end link probability distribution*).

To illustrate the basic idea of STARS, we use a simple scenario shown in Fig. 1 as an example. In this network, there are three wireless nodes (1, 2, and 3). Node 2 is located in the transmission range of node 1, and node 3 is located in the transmission range of node 2 (but not the transmission range of node 1). Two consecutive packets are detected: node 1 broadcasts a packet and then node 2 broadcasts a packet.

4.1 Traffic Matrices Construction

4.1.1 Point-to-Point Traffic Matrix

With the captured point-to-point (one-hop) traffic in a certain period T, we first need to build point-to-point traffic matrices such that each traffic matrix only contains "independent" one-hop packets. Note that two packets captured at different time could be the same packet appearing at different locations, such as the two packets sent by node 1 and node 2 consecutively in Fig. 1, so they are "dependent" on each other. To avoid a single point-to-point traffic matrix from containing two dependent packets, we apply a "time slicing" technique as shown in Fig. 2. That is, we take snapshots of the network, and each snapshot is triggered by a captured packet. A sequence of snapshots during a time interval Δt_e constructs a slice represented by

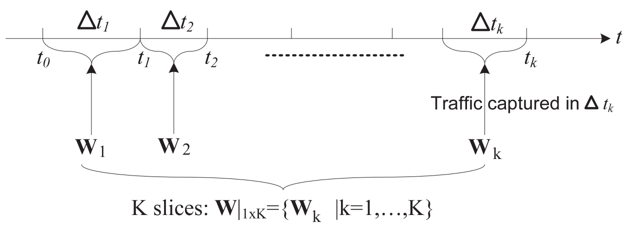


Fig. 2. Slicing the time domain.

a traffic matrix $\mathbf{W}_e$, which is an $N \times N$ one-hop traffic relation matrix. The length of each time interval Δt_e is determined by two criteria: 1) A node can be either a sender or a receiver within this time interval. But it cannot be both. 2) Each traffic matrix must correctly represent the one-hop transmissions during the corresponding time interval. In this way, the construction of matrices $\mathbf{W}|_{\mathbf{1}\times\mathbf{e}} = (\mathbf{W}_1, \mathbf{W}_2, \ldots, \mathbf{W}_K)$ will automatically involve mobility in the traffic matrices constructions. For example, traffic matrix $\mathbf{W}_e = (w_e(i,j))_{N\times N}$ is created for direct transmissions between nodes during time interval Δt_e. Since each snapshot of the network is triggered by capturing a packet, as long as potential receiver j is located within sender i's communication range (i.e., $d_{i,j} \leq r$), a small change of distance $d_{i,j}$ due to mobility will not alter the value assigned to $w_e(i,j)$. If j moves out of the communication range of i due to mobility, the value of $w_e(i,j) = 0$. In this way, we slice the period T into a sequence of time intervals $\Delta t_1, \Delta t_2, \ldots, \Delta t_K$, and record the captured packets into their corresponding traffic matrices $\mathbf{W}|_{\mathbf{1}\times\mathbf{K}} = (\mathbf{W}_1, \mathbf{W}_2, \ldots, \mathbf{W}_K)$. In each traffic matrix $\mathbf{W}_e = (w_e(i,j))_{N\times N}$ (N is the size of the network, $e = 1, 2, \ldots, K$), the entry $w_e(i,j)$ is the point-to-point traffic volume (number of packets) captured from node i to node j during the time interval Δt_e(*we define* $w_e(i,i)$ *to be 0*). In addition, we use $w_e(i,j).pkt$ to denote the set of all packets contributing to $w_e(i,j)$. The "time slicing" has to make sure that all packets captured in any of the time intervals are independent with each other. In other words, two packets residing in different entries of the same matrix must not be the same packet transmitted through multiple hops. Note that, using the "time slicing" techniques, we also effectively handle the nodal mobility by taking snapshots of a sequence of relatively fixed network topologies.

In addition to the "time slicing," we need to follow the three rules listed below: 1) The number of captured packets rather than the actual size of payloads is considered as the "traffic volume," since the size of payloads does not affect the traffic pattern (and we assumed all MAC frames are of the same length due to the application of padding). 2) All nodes within the transmitting range of a packet have the same probability to be the actual receiver. For example, if a node i broadcasts a packet in the time interval Δt_e, and nodes $j_1, j_2, \ldots j_n$ are all within i's transmitting range, then the entries $w_e(i,j_1)$, $w_e(i,j_2) \ldots w_e(i,j_n)$ should be all equally increased by $1/n$. This is equivalent to dividing a packet into n subpackets and each sent to one neighboring node. For simplicity, in the remainder of the paper, we denote the original packet as *"virtual size"* 1 and each of the subpackets as *"virtual size"* $1/n$. 3) Each packet p in $w_e(i,j).pkt$, has three associated features: $p.vsize$, $p.time$, and $p.hop$, denoting the "virtual size," transmitting time, and hop count of this packet, respectively. A packet's hop count is set to 1 when added to the point-to-point traffic matrix.

That said, for the example given in Fig. 1, we could derive:

$$\mathbf{W}_1 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \mathbf{W}_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0.5 & 0 & 0.5 \\ 0 & 0 & 0 \end{bmatrix}.$$

Note that in $\mathbf{W}_2$, a real packet sent by node 2 is divided into two subpackets of virtual size 0.5, which means nodes 1 and 3 are equally likely to be the actual receiver.

4.1.2 End-to-End Traffic Matrix

Given a sequence of point-to-point traffic matrices $\mathbf{W}|_{\mathbf{1}\times\mathbf{K}}$, our goal is to derive the end-to-end traffic matrix $\mathbf{R} = (r(i,j))_{N\times N}$, where $r(i,j)$ is the accumulative traffic volume from node i to node j, including both the point-to-point traffic captured directly and multihop traffic deduced from the point-to-point traffic. In this paper, we use the term accumulative traffic matrix and end-to-end traffic matrix interchangeably. The following Algorithm 1 (function f) takes $\mathbf{W}|_{\mathbf{1}\times\mathbf{K}}$ as the inputs to derive the accumulative traffic matrix $\mathbf{R}$.

Algorithm 1. —$f(\mathbf{W}|_{1\times K})$.
1: $\mathbf{R} = \mathbf{W}_1$
2: **for** $e = 1$ to $K-1$ **do**
3: $\quad\mathbf{R} = g(\mathbf{R}, \mathbf{W}_{e+1}) + \mathbf{W}_{e+1}$
4: **end for**
5: **return R**

In this algorithm, each update to $\mathbf{R}$ (line 3) includes the multihop traffic derivation function g shown as in Algorithm 2, and the addition of the point-to-point traffic matrix which is the evidence of possible direct (single-hop) communication.

Algorithm 2. —$g(\mathbf{R}, \mathbf{W}_{e+1})$.
1: $\mathbf{R}' = \mathbf{R}$
2: **for** $i = 1$ to N **do**
3: **for** $k = 1$ to N **and** $k \neq i$ **do**
4: **for** $j = 1$ to N **do**
5: **for each** $x \in w_{e+1}(j,k).pkt$ **do**
6: **if** $\exists\, y \in r(i,j).pkt$ s.t. $x.time - y.time < \mathcal{T}$ **and** $y.hop < \mathcal{H}$ **then**
7: create z with $z.time = x.time$
$z.hop = y.hop + 1$
$z.vsize = \min\{x.vsize, y.vsize\}$
8: $r'(i,k).pkt = r'(i,k).pkt \cup \{z\}$
9: $r'(i,k) = r'(i,k) + z.vsize$
10: **end if**
11: **end for**
12: **end for**
13: **end for**
14: **end for**
15: **return** $\mathbf{R}'$

Function g takes two inputs: 1) $\mathbf{R}$ is an end-to-end traffic matrix derived from point-to-point matrices $\mathbf{W}_1$ to $\mathbf{W}_e$, and 2) $\mathbf{W}_{e+1}$ is the next point-to-point traffic matrix. The output is the end-to-end traffic matrix derived from $\mathbf{W}_1$ to $\mathbf{W}_{e+1}$.

For each packet x recorded in $\mathbf{W}_{e+1}$, the function tries to find a packet y in $\mathbf{R}$ that is potentially the same packet transmitted at x's previous hop. If such a packet y exists, then a multihop flow (packet) from the source of y to the destination of x should be derived. For instance, in our example scenario, we first let $\mathbf{R} = \mathbf{W}_1$. Then $g(\mathbf{R}, \mathbf{W}_2)$ should derive all possible end-to-end flows. $\mathbf{W}_2$ contains two packets, sent from node 2 to nodes 1 and 3, respectively. Let $p_{2,1}$ and $p_{2,3}$ denote these two packets. The current $\mathbf{R}$ contains only one packet $p_{1,2}$ sent from node 1 to node 2. Thus, it is possible that $p_{1,2}$ and $p_{2,3}$ are the same packet appearing at different hops. In this case, a new packet $p_{1,3}$ is derived to represent a multihop flow from node 1 to node 3. Since the volume of a multihop flow consisting of a sequence of one-hop transmissions cannot exceed the volume of any of the transmissions, we have $p_{1,3}.vsize = \min\{p_{1,2}.vsize, p_{2,3}.vsize\} = 0.5$. Two constraints are considered for reasonable traffic inference: The difference between the transmitting time of a packet at two consecutive hops cannot be too large and the hop-count of a packet cannot exceed a maximum value . We use $\mathcal{T}$ and $\mathcal{H}$ to represent the timing threshold and maximal hop-count threshold, respectively. If the network diameter is d, the average transmission distance of a mobile node is r, we can derive the approximated maximal hop-count threshold as: $\mathcal{H} = \lceil d/r \rceil$. The timing threshold $\mathcal{T}$ must be at least the value of the maximum retransmission time. It depends on the specification of the MAC protocol. For instance, if the 802.11 protocol is being used, $\mathcal{T}$ is determined by the maximum number of retransmissions, the contention window size, and the exponential back-off algorithm.

After executing function $f(\mathbf{W}\mid_{1\times K})$, we can derive the accumulative traffic matrix $\mathbf{R}$ for the time period $\sum_{k=1}^{K} \Delta t_k$, in which the ith row is the vector of the outgoing traffic from node i and the jth column is the vector of the traffic destining to node j.

Applying Algorithm 1, we derive the following matrix $\mathbf{R}$ for our presented example. For simplicity, we assume the timing and hop-count thresholds do not filter any packet out.

$$\mathbf{R} = \begin{bmatrix} 0 & 1 & 0.5 \\ 0.5 & 0 & 0.5 \\ 0 & 0 & 0 \end{bmatrix}.$$

It can be seen that, $\mathbf{R}$ contains not only all the one-hop packets captured by $\mathbf{W}_1$ and $\mathbf{W}_2$, but also a derived two-hop flow of size 0.5 from node 1 to node 3.

4.2 Traffic Pattern Discovery

The traffic matrix $\mathbf{R}$ tells us the deduced end-to-end traffic volume between each pair of nodes. However, we still need to perform further investigation to discover the actual *source/destination probability distribution* and *end-to-end link probability distribution*, that is, to figure out who are the actual sources and destinations and who are communicating with whom.

4.2.1 Source/Destination Probability Distribution

We denote the actual source and destination probability distribution, respectively, as two vectors $\mathbf{S} = (s(1), s(2), \ldots, s(N))$ and $\mathbf{D} = (d(1), d(2), \ldots, d(N))$, where $s(i)$ and $d(i)$ ($i = 1$ to N) represent the probability for node i to be an actual source and destination, respectively. Note that if the total number of source nodes is m, then we should have $\sum_{i=1}^{N} s(i) = m$ for $\mathbf{S}$. However, since we only care about the relative order among all possibilities (to know which nodes are more possible to be the actual sources) but not the total number m, we can always assume $m = 1$. It is the same case for $\mathbf{D}$ and all the probability vectors we will calculate later in this paper. That is, *all probability distribution vectors in this paper are normalized*[1] *and only the relative orders among the elements of each vector actually make sense.*

To derive $\mathbf{S}$ and $\mathbf{D}$, we compute two series of vectors which converge to $\mathbf{S}$ and $\mathbf{D}$, respectively: the source probability distribution vector series $\overline{S} = (\mathbf{S_0}, \mathbf{S_1}, \ldots, \mathbf{S_n}, \ldots)$, and the destination probability distribution vector series $\overline{D} = (\mathbf{D_0}, \mathbf{D_1}, \ldots, \mathbf{D_n}, \ldots)$.

First, both $\mathbf{S_0}$ and $\mathbf{D_0}$ should be uniform probability distribution vectors: $\mathbf{S_0} = \mathbf{D_0} = (1/N, 1/N, \ldots, 1/N)$, since without any traffic information, all nodes are equally likely to be sources and destinations.[2]

Second, we note that the ith row $(r(i,1) \ldots r(i,N))$ in the matrix $\mathbf{R}$ is a vector of the traffic from node i to every node in the MANET. If we multiply this vector by $\mathbf{D_0}$ (inner product), we get

$$s'(i) = \sum_{j=1}^{N} r(i,j) \times d_0(j), \tag{1}$$

which is the probability for node i to be a source based on the destination probability distribution $\mathbf{D_0}$. This is intuitive, since if a node sends a lot of packets to another node with high probability of being a destination, the node itself has a high probability of being a source. According to this, the normalized inner product of $\mathbf{R}$ and $\mathbf{D_0}$ is a vector of probabilities for nodes to be source nodes. Similarly, using $\mathbf{S}'$ to denote the vector $(s'(1), s'(2), \ldots, s'(N))$ resulted from (1) and multiplying the ith row in the transpose of $\mathbf{R}$ (i.e., $\mathbf{R^T}$) by $\mathbf{S}'$, we will get

$$d_1(i) = \sum_{j=1}^{N} r(j,i) \times s'(j), \tag{2}$$

which is the probability for node i to be a destination derived from $\mathbf{S}'$ and in turn based on $\mathbf{D_0}$. This claim is based on the fact that if a node receives a lot of packets from a node with high probability of being a source, the node itself has a high probability of being a destination. Consequently, the normalized inner product of $\mathbf{R^T}$ and $\mathbf{S}'$ generates $\mathbf{D_1}$ as a new probability vector for nodes to be destinations. Through this procedure, $\mathbf{D_1}$ is closer to the actual destination probability distribution than $\mathbf{D_0}$.

For the example scenario given in Fig. 1, we initialize $\mathbf{D}_0$ to be $(1/3, 1/3, 1/3)^T$, without any prior knowledge about the actual destinations. Then we compute $\mathbf{S}' = \mathbf{R} \cdot \mathbf{D}_0 = (1/2, 1/3, 0)^T$, which can be normalized to $\mathbf{S}' = (3/5, 2/5, 0)^T$. $\mathbf{S}'$ indicates that node 1 is most likely to

1. In this paper, "normalizing" a vector means dividing each element of the vector by the summation of all the elements, which makes the summation of all the elements of the "normalized" vector equal to 1.

2. If there is prior knowledge available about the sources and destinations, $\mathbf{S_0}$ and $\mathbf{D_0}$ should be set to reflect the prior knowledge.

be an actual source, while node 3 is definitely not a source. Next, we multiply $\mathbf{R^T}$ by $\mathbf{S}'$ and get the normalized $\mathbf{D}_1$ as: $(0.15, 0.46, 0.39)^T$, which should be closer to the actual destination probability distribution than $\mathbf{D}_0$.

According to the analysis above, we derive the following iterative algorithm for $\overline{D}$:

$$\mathbf{D_{n+1}} = (\mathbf{R^T} \cdot \mathbf{R}) \cdot \mathbf{D_n}, \tag{3}$$

and similarly that for $\overline{S}$:

$$\mathbf{S_{n+1}} = (\mathbf{R} \cdot \mathbf{R^T}) \cdot \mathbf{S_n}. \tag{4}$$

We also notice that geographically adjacent nodes may have negative impacts on the accuracy of the algorithms above. For example, if node j is one of the neighbors of node i, j may frequently forward the packets originated from node i to other nodes in the network and also frequently forward the packets from other nodes to node i. In this case, the high probability for node j to be a source does not indicate the high probability for node i to be a destination, though the traffic volume from j to i is large. On the other hand, the high probability for node j to be a destination and the large traffic volume from i to j do not indicate the high probability for node i to be a source. We call this kind of negative impacts as the "neighborhood noise." Especially, when the mobility is low, the negative impacts will be substantial since the neighborhood of a node rarely changes.

To reduce the neighborhood noise, we utilize the vector space similarity assessment. The vector space similarity (or cosine similarity) of two vectors $\mathbf{V}$ and $\mathbf{U}$ is defined as follows:

$$Sim(\mathbf{V}, \mathbf{U}) = \mathbf{V} \cdot \mathbf{U} / (|\mathbf{V}||\mathbf{U}|),$$

where $\mathbf{V} \cdot \mathbf{U}$ denotes the dot product of $\mathbf{V}$, and $\mathbf{U}$, $|\mathbf{V}|$, and $|\mathbf{U}|$ denote the norm of $\mathbf{V}$ and $\mathbf{U}$. We realize that, if two nodes have similar outgoing and incoming traffic vectors (in the end-to-end traffic matrix $\mathbf{R}$), they are likely to be neighboring nodes (relays of each other), and so they should have less impact on the source/destination probability distribution of each other. Thus, we rewrite (1) and (2) by the following two formulas:

$$s'(i) = \sum_{j=1}^{N} r(i,j) \times d_0(j) \times c(i,j), \tag{5}$$

$$d_1(i) = \sum_{j=1}^{N} r(j,i) \times s'(j) \times c(j,i), \tag{6}$$

where

$$c(i,j) = c(j,i) = 1 - \frac{Sim(\mathbf{O(i)}, \mathbf{O(j)}) + Sim(\mathbf{I(i)}, \mathbf{I(j)})}{2},$$

where $\mathbf{O(i)}$ and $\mathbf{O(j)}$ denote the ith row and jth row in $\mathbf{R}$ (i.e., the outgoing traffic from i and j), while $\mathbf{I(i)}$ and $\mathbf{I(j)}$ denote the ith and jth column in $\mathbf{R}$ (i.e., the incoming traffic to i and j).

Define a function $\mathbf{\Phi}$ such that $\mathbf{\Phi}(\mathbf{R}) = (\phi(i,j))_{N\times N}$, where $\phi(i,j) = r(i,j) \times c(i,j)$. Obviously, we have $\mathbf{\Phi}(\mathbf{R^T}) = \mathbf{\Phi^T}(\mathbf{R})$, in which $\mathbf{\Phi^T}(\mathbf{R})$ denotes the transpose of $\mathbf{\Phi}(\mathbf{R})$. According to (5) and (6), we improve (3) and (4) with the following two iterations, respectively:

$$\mathbf{D_{n+1}} = (\mathbf{\Phi^T}(\mathbf{R}) \cdot \mathbf{\Phi}(\mathbf{R})) \cdot \mathbf{D_n}, \tag{7}$$

$$\mathbf{S_{n+1}} = (\mathbf{\Phi}(\mathbf{R}) \cdot \mathbf{\Phi^T}(\mathbf{R})) \cdot \mathbf{S_n}. \tag{8}$$

By introducing the vector space similarity assessment, we ensure that, two nodes with higher probability to be neighbors (relays of each other) have less impact on each other's source/destination probability distribution, which reasonably reduces the neighborhood noise. Finally, we propose the following Algorithms 3 and 4 to compute $\overline{S}$ and $\overline{D}$.

Algorithm 3. —$Src(\mathbf{R})$.

```
S_0 = (1/N, 1/N, ..., 1/N)
n = 0
do
  S_{n+1} = (Φ(R) · Φ^T(R)) · S_n
  normalize S_{n+1}
  n = n + 1
while S_n ≠ S_{n-1}
S = S_n
return S
```

Algorithm 4. —$Dest(\mathbf{R})$.

```
D_0 = (1/N, 1/N, ..., 1/N)
n = 0
do
  D_{n+1} = (Φ^T(R) · Φ(R)) · D_n
  normalize D_{n+1}
  n = n + 1
while D_n ≠ D_{n-1}
D = D_n
return D
```

The iterations will converge to $\mathbf{S}$ and $\mathbf{D}$, which are the actual source and destination probability distribution vectors.

Based on the proposed algorithms, we can derive the two final vectors for the example scenario as given below. Here, we ignore the neighborhood noise reduction for simplicity:

$$\mathbf{S} = (0.77, 0.23, 0)^T,$$

$$\mathbf{D} = (0.08, 0.55, 0.37)^T.$$

4.2.2 *End-to-End Link Probability Distribution*

Our goal in this section is to derive a probability distribution matrix $P = (p(i,j))_{N\times N}$, in which each entry $p(i,j)$ represents the probability of the $i \rightarrow j$ linkability (i.e., node i and node j are a pair of actual source and destination). Again, note that only the relative order among these entries is of interest, since we aim at discovering the most possible communication links.

As described above, the probability for node i to be a destination depends on two factors: the traffic from each node j to node i and node j's probability to be a source. Suppose $j - i$ is an actual source-destination pair. If we set the total traffic coming out from j to zero, the probability for i to be a destination will decrease. Similarly, if we set the incoming traffic to node i to zero, the probability for node j to be a source will also decrease. Thus, we can identify a source-destination (S-D) pair by evaluating the significance of the probability reduction due to the elimination of the traffic sent by the source or received by the destination. For instance, in the example scenario shown in Fig. 1, to identify the most possible destination of node 1, we can erase all traffic sent by node 1 from the point-to-point traffic matrices, base on which we compute the destination probability distribution $\mathbf{D}^-$. By comparing $\mathbf{D}^-$ with $\mathbf{D}$ (obtained using the original point-to-point matrices), we can find out the node whose destination probability drops most significantly due to elimination of the traffic sent by node 1. This node is most possible to be the destination of node 1.

That said, we propose Algorithms 5 and 6 to discover the S-D linkability. The two algorithms are quite similar, so we only explain Algorithm 5 here. First, we apply Algorithm 1 (function f) to the original point-to-point traffic matrices and derive the original end-to-end traffic matrix $\mathbf{R}$ (line 1). Then we apply Algorithm 4 (function $Dest$) to $\mathbf{R}$ and obtain the original destination probability distribution vector $\mathbf{D}$ (line 2). Then, the point-to-point matrices are modified by eliminating the traffic sent by node i (line 3), and the destination probability distribution vector $\mathbf{D}^-$ (lines 4 and 5) is recomputed. Subtracting $\mathbf{D}^-$ from $\mathbf{D}$ results in a vector $\mathbf{L'_{s-d}(i)}$, which indicates the level of each node to be affected by the traffic elimination (line 6). Then the normalized vector $\mathbf{L_{s-d}(i)}$ is a vector of probability for each node to be the intended destination of i (line 7). The function Suppress-Sender(i) in Algorithm 5 is used to remove the traffic sent by node i. Accordingly, Suppress-Receiver(j) used in Algorithm 6 is used to remove the traffic received by node j. They work as follows:

Suppress-Sender(i)	Suppress-Receiver(j)
$\mathbf{W'}\vert_{\mathbf{1\times K}} = \mathbf{W}\vert_{\mathbf{1\times K}}$	$\mathbf{W'}\vert_{\mathbf{1\times K}} = \mathbf{W}\vert_{\mathbf{1\times K}}$
for $p = 1$ to K	for $p = 1$ to K
for $q = 1$ to N	for $q = 1$ to N
$w'_p(i,q) = 0$	$w'_p(q,j) = 0$
return $\mathbf{W'}\vert_{\mathbf{1\times K}}$	return $\mathbf{W'}\vert_{\mathbf{1\times K}}$

Algorithm 5. Given a source node i, compute the probability distribution vector $\mathbf{L_{s-d}(i)}$ for each node to be the intended destination of i.

1: $\mathbf{R} = f(\mathbf{W}\ |_{1\times K})$;
2: $\mathbf{D} = Dest(\mathbf{R})$;
3: $\mathbf{W'}\ |_{\mathbf{1\times K}} =$ Suppress-Sender(i);
4: $\mathbf{R'} = f(\mathbf{W'}\ |_{1\times K})$;
5: $\mathbf{D}^- = Dest(\mathbf{R'})$;
6: Calculate the probability reduction vector as: $\mathbf{L'_{s-d}(i)} = \mathbf{D} - \mathbf{D}^-$. If negative elements exist in $\mathbf{L'_{s-d}(i)}$, increase each element by the absolute value of the smallest negative element;
7: Normalize $\mathbf{L'_{s-d}(i)}$ to generate the probability vector $\mathbf{L_{s-d}(i)}$ for each node to be the intended destination of i;
8: Return $\mathbf{L_{s-d}(i)}$.

Algorithm 6. Given a destination node j, compute the probability distribution vector $\mathbf{L_{d-s}(j)}$ for each node to be the corresponding source of j.

1: $\mathbf{R} = f(\mathbf{W}\ |_{1\times K})$;
2: $\mathbf{S} = Src(\mathbf{R})$;
3: $\mathbf{W'}|_{\mathbf{1\times K}} = \mathrm{Suppress} - \mathrm{Receiver}(j)$;
4: $\mathbf{R'} = f(\mathbf{W'}\ |_{1\times K})$;
5: $\mathbf{S}^- = Src(\mathbf{R'})$;
6: Calculate the probability reduction vector as: $\mathbf{L'_{d-s}(j)} = \mathbf{S} - \mathbf{S}^-$. If negative elements exist in $\mathbf{L'_{d-s}(j)}$, increase each element by the absolute value of the smallest negative element;
7: Normalize $\mathbf{L'_{d-s}(j)}$ to generate the probability vector $\mathbf{L_{d-s}(j)}$ for each node to be the corresponding source of j;
8: Return $\mathbf{L_{d-s}(j)}$.

Going back to our example in Fig. 1, let us illustrate how Algorithm 5 computes the probability of each node being the intended destination of node 1. We first erase all traffic sent by node 1 from the point-to-point traffic matrices and get

$$\mathbf{W'}_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \mathbf{W'}_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0.5 & 0 & 0.5 \\ 0 & 0 & 0 \end{bmatrix}.$$

Then the new end-to-end matrix can be derived as

$$\mathbf{R'} = \begin{bmatrix} 0 & 0 & 0 \\ 0.5 & 0 & 0.5 \\ 0 & 0 & 0 \end{bmatrix}.$$

Given $\mathbf{R'}$, the new destination probability vector $\mathbf{D}^-$ can be calculated using Algorithm 4 as

$$\mathbf{D}^- = (0.5, 0, 0.5)^T.$$

Previously, we already calculated $\mathbf{D}$ as $(0.08, 0.55, 0.37)^T$, which means $\mathbf{L'_{s-d}(1)} = \mathbf{D} - \mathbf{D}^- = (-0.42, 0.55, -0.13)^T$ and $\mathbf{L_{s-d}(1)} = (0, 0.77, 0.23)^T$. Obviously, the destination probability of node 2 drops most significantly by eliminating the traffic sent by node 1, and so we know that node 2 is the most possible destination of node 1.

Now, for an arbitrary source node i, we can derive a vector $\mathbf{L_{s-d}(i)} = (l_{s-d}(i,1), l_{s-d}(i,2), \ldots, l_{s-d}(i,N))$, which specifies the probability for each node to be the intended destination of i. For an arbitrary destination node j, we can derive a vector $\mathbf{L_{d-s}(j)} = (l_{d-s}(j,1), l_{d-s}(j,2), \ldots, l_{d-s}(j,N))$, which specifies the probability for each node to be the corresponding source of j. Since $l_{s-d}(i,j)$ represents the probability for j to be the destination of i (given that node i is a source) and $l_{d-s}(j,i)$ represents the probability for i to be the source of j (given that node j is a destination), we can derive the probability of a source-destination link $(i \rightarrow j)$ as follows:

$$p(i,j) = p'(i,j) / \sum_{m=1}^{N} \sum_{n=1}^{N} p'(m,n), \tag{9}$$

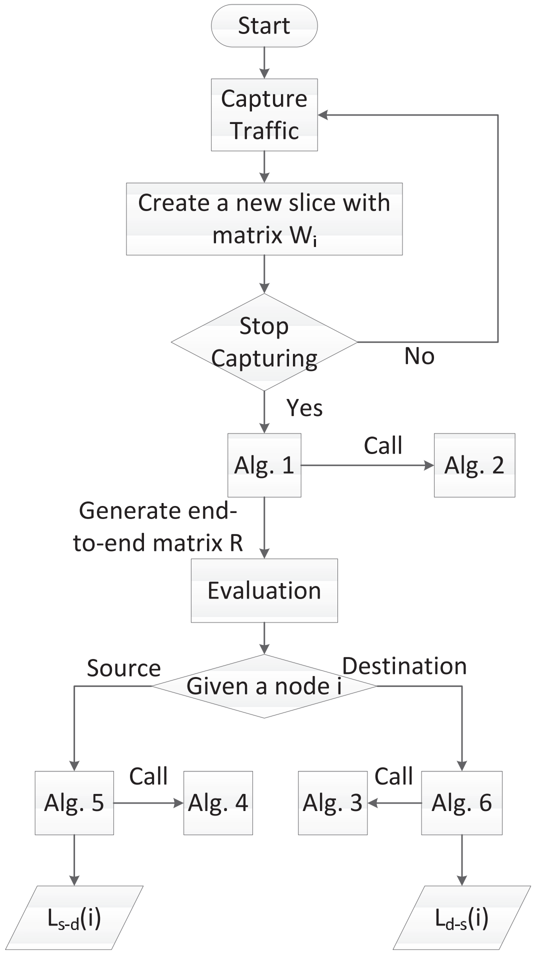


Fig. 3. Work flow of STARS.

where

$$p'(i,j) = s(i) * l_{s-d}(i,j) + d(j) * l_{d-s}(j,i), \tag{10}$$

and $s(i)$ is the ith element of the $\mathbf{S}$ and $d(j)$ is the jth element of the $\mathbf{D}$.

The work flow of STARS is shown in Fig. 3.

5 Experiments

In this section, we present the empirical study, consisting of two components: demonstration and evaluation. First, we use three simple scenarios to demonstrate (partially) the direct outputs of STARS, i.e., the probability distributions. Then, we use the probability distributions produced by STARS to identify the actual sources, destinations and end-to-end links for a large set of simulations, and evaluate the performance in terms of average false-positive rate (fpr) and false-negative rate (fnr). The network environment is simulated using Qualnet [26]. The network protocol stack is modified so that the communication model presented in Section 3.1 is simulated.

5.1 Demonstration

The MANET for demonstration is comprised of 30 mobile nodes randomly deployed in an $800 \times 800\ \mathrm{m}^2$ area. There are three different scenarios: **(S1)** Only one source (node 2) generates constant bit-rate (CBR) traffic to four destinations (nodes 3, 6, 18, and 29). **(S2)** Four source nodes (nodes 2, 5, 11, and 20) generate CBR traffic to the only destination (node 3). **(S3)** CBR traffic is generated between 15 end-to-end communication pairs: 2-3, 5-6, 5-7, 7-8, 8-9, 8-10, 9-10, 10-11, 10-12, 12-13, 12-14, 14-15, 15-16, 16-17, and 16-18. The simulation lasts for 200 seconds for each scenario. Other system parameters are shown in Table 1.

TABLE 1
System Parameters Configuration

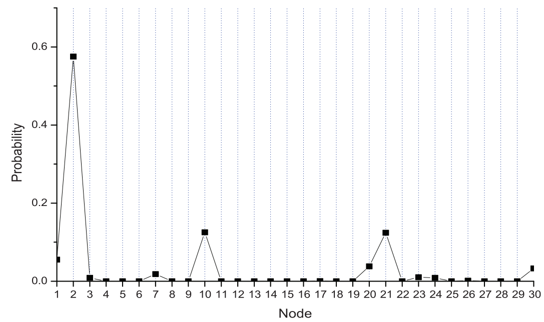
Node Speed	Transmission Rate	Mobility Model	$\mathcal{T}(s)$	$\mathcal{H}$(hops)
5$\sim$10m/s	11$Mbps$	random waypoint	1.0	5

5.1.1 Source/Destination Probability Distribution

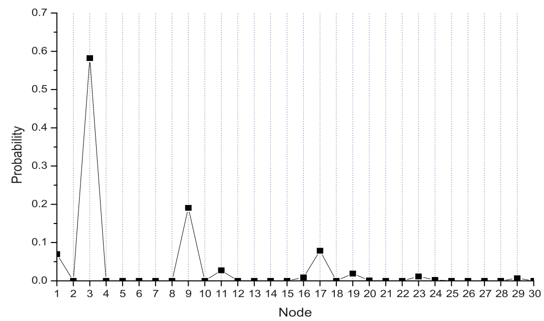
The first two scenarios demonstrate the ability of STARS to identify the source and destination by calculating the source/destination probability distribution. Figs. 4a and 4b are the source probability distribution of (S1) and the destination probability distribution of (S2), derived by Algorithms 3 and 4, respectively. In Fig. 4a, node 2 has much higher probability than other nodes to be the source, and in Fig. 4b, node 3 also has the highest probability to be the destination, which match the simulation setup.

5.1.2 End-to-End Link Probability Distribution

Fig. 5 shows the results of applying Algorithm 5 to (S3). The results of applying Algorithm 6 are symmetric to those shown here, so they are not illustrated. In Fig. 5, the

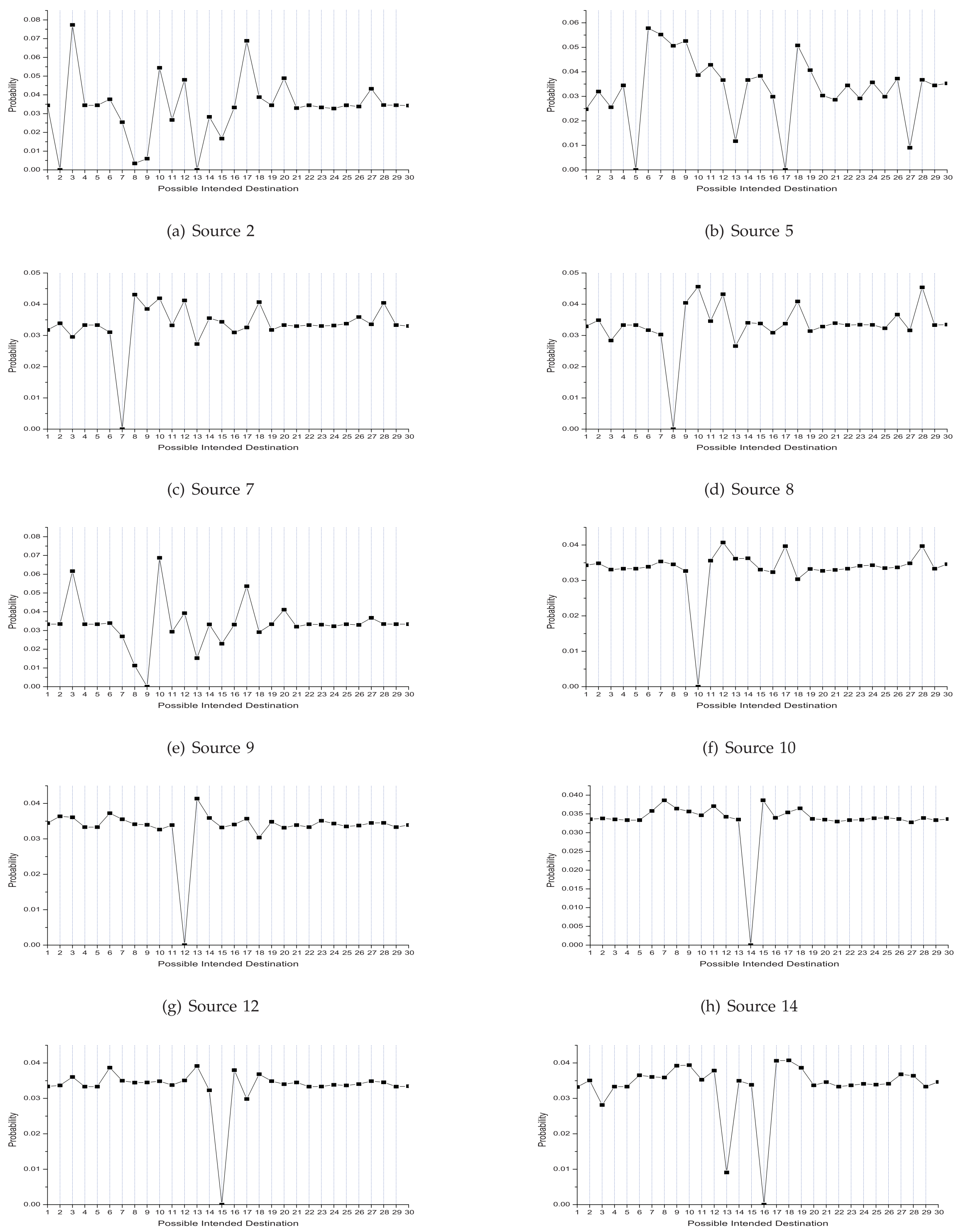


(a) Source Probability Distribution in (S1).



(b) Destination Probability Distribution in (S2).

Fig. 4. Results of (S1) and (S2).



(a) Source 2

(b) Source 5

(c) Source 7

(d) Source 8

(e) Source 9

(f) Source 10

(g) Source 12

(h) Source 14

(i) Source 15

(j) Source 16

Fig. 5. Probability distribution of being the intended destination of each source node in (S3).

probability distribution for every node to be the intended destination is depicted for each source node. Most of these curves tell the truth of the actual traffic pattern. For example, in Fig. 5a, the highest peak is at node 3 (which means node 3 is most likely to be the intended destination of source node 2); in Fig. 5b, the highest peak is at node 6; in Fig. 5e, the highest peak is at node 10. All these results match the actual CBR traffic pattern perfectly. However, some of the derived probability distributions have incorrect indications, such as in Fig. 5d, node 28 has the highest probability to be the destination of node 8. This is because some of the forwarders cannot be distinguished from the actual destination of a source or the actual source of a destination by using STARS, which means the MANET still has a certain level of communication relation anonymity under STARS.

TABLE 2
Confusion Matrix

	Actual Source/E2E Link	Not Actual Source/E2E Link
Identified	True Positive (tp)	False Positive (fp)
Not Identified	False Negative (fn)	True Negative (tn)

5.2 Evaluation

From the previous section, we see that the probability distributions produced by STARS are good indicators of the actual traffic patterns, i.e., actual sources, destinations, and end-to-end links. Different strategies can be used to speculate the actual traffic patterns from the probability distributions. In this section, we evaluate the performance of STARS based on the following two basic strategies, $\mathbf{T_1}$ and $\mathbf{T_2}$.

[$\mathbf{T_1}$] Suppose the number of actual sources, destinations, or end-to-end links is known to be k. We simply select the top k items (nodes or links) with the highest probabilities.

[$\mathbf{T_2}$] Suppose the number k is unknown. We keep selecting the top items with the highest probabilities until both of the two criteria are satisfied: 1) the sum of the probabilities of the selected items has reached u; and 2) the probability of the last selected item is v times larger than the current one. u and v are two adjustable thresholds, which are set to 0.8 and 4 in our experiments, respectively.

The simulated MANET is comprised of 80 mobile nodes deployed in a $1{,}000 \times 1{,}000\ \mathrm{m}^2$ area. Other system parameters remain the same as in the previous section. The number of sources varies from 6 to 15 (totally 10 cases), and each source node has a single destination. Both the sources and destinations are randomly selected. For each of the 10 cases, we run the simulation 10 rounds. In each round, we use $\mathbf{T_1}$ and $\mathbf{T_2}$ to identify the sources[3] and end-to-end links, and compare them with the actual traffic patterns to calculate the false-positive rate and false-negative rate based on the typical confusion matrix as shown in Table 2.

Assuming we identify k' sources (end-to-end links) in which only c of them are actual sources (end-to-end links), the false-positive rate is defined as follows:

$$fpr = \frac{fp}{fp + tn} = \frac{k' - c}{N - k}, \tag{11}$$

and the false-negative rate is defined as follows:

$$fnr = \frac{fn}{fn + tp} = \frac{k - c}{k}, \tag{12}$$

where k is the total number of actual sources (end-to-end links) and N is the total number of all nodes (node pairs).

The average values (over the 10 rounds for each case) of the false-positive rate and false-negative rate are shown in Fig. 6. From Figs. 6a and 6b, we can see that both $\mathbf{T_1}$ and $\mathbf{T_2}$ achieve reasonably good accuracy for source identification. Using $\mathbf{T_1}$, the false-positive rate is almost always less than 0.05, although it increases slightly as the number of sources goes up; and the peak of the false-negative rate is lower than 0.3. $\mathbf{T_2}$ tends to select more nodes as source nodes, which inevitably increases the false-positive rate. However, the false-negative rate is decreased at the same time. A valuable observation is that, $\mathbf{T_1}$ is not necessarily better than $\mathbf{T_2}$ by knowing the number of items to be selected. Figs. 6c and 6d confirm this observation. When the number of actual end-to-end links is known ($\mathbf{T_1}$ is applied), we have almost perfect false-positive rate (close to zero), but unfortunately the extremely large false-negative rate is not acceptable. This simply means that too few links are selected and most of the actual end-to-end communication relations are not discovered. On the other hand, $\mathbf{T_2}$ selects more links, which reveals most of the actual end-to-end links by slightly sacrificing the false-positive rate. Specifically, in most cases, more than 80 percent of the actual end-to-end links are revealed (i.e., the false-negative rate is less than 0.2), while the false-positive rate is not more than 0.16.

To conclude the evaluation, the hidden traffic patterns can be discovered in good accuracy using STARS, even without the number of actual sources, destinations, and end-to-end communication relations known to the traffic analyzers.

3. We skip finding the destinations since it is symmetric with finding the sources.

6 Discussion and Future Work

The adversarial model presented in Section 3.2 assumes that the adversaries can globally monitor the traffic across the entire network region. This assumption is conservative from the network users' point of view. Usually, it is difficult for the attackers to perform such a global traffic detection. However, even though the adversaries are not able to monitor the entire network, they can monitor several parts of the network simultaneously. For example, an attacker can deploy sensors (signal detectors) around some particular mobile nodes to track their movements and eavesdrop all of their traffic. These sensors may even move accordingly. With the restricted capabilities, the attacker can take advantage of STARS to perform traffic analysis as follows:

1. divide the entire network into multiple regions geographically;
2. deploy sensors along the boundaries of each region to monitor the cross-component traffic;
3. treat each region as a supernode and use STARS to figure out the sources, destinations, and end-to-end communication relations; and
4. analyze the traffic even when nodes are close to each other by treating the close nodes as a supernode.

We call this variant of STARS as the *Generalized STARS* (GSTARS). To perform GSTARS, the adversaries only need to monitor the nodes beside the boundaries of the supernodes. The traffic inside each supernode can be ignored, since it will not affect the inter-region traffic patterns. In addition, GSTARS does not need the signal detectors to be able to precisely locate the signal source. They are only required to determine which supernode (region) the signals are sent from. Moreover, in STARS, the actual receiver of a point-to-point transmission is not identifiable among all the potential receivers within the sender's transmitting range. This inaccuracy can be mitigated in GSTARS because most potential receivers of a packet will be contained within one or a few supernodes. GSTARS will be the direction of our future research.

(a) Average False Positive Rate of Source Identification.

(b) Average False Negative Rate of Source Identification.

(c) Average False Positive Rate of E2E Link Identification.

(d) Average False Negative Rate of E2E Link Identification.

Fig. 6. Evaluation results.

7 Conclusion

In this paper, we propose a novel STARS for MANETs. STARS is basically an attacking system, which only needs to capture the raw traffic from the PHY/MAC layer without looking into the contents of the intercepted packets. From the captured packets, STARS constructs a sequence of point-to-point traffic matrices to derive the end-to-end traffic matrix, and then uses a heuristic data processing model to reveal the hidden traffic patterns from the end-to-end matrix. Our empirical study demonstrates that the existing MANET systems can achieve very restricted communication anonymity under the attack of STARS.

Acknowledgments

This work was supported by the US Army Research Office Grant W911NF-11-1-0191.

References

[1] J. Kong, X. Hong, and M. Gerla, "An Identity-Free and On-Demand Routing Scheme against Anonymity Threats in Mobile Ad Hoc Networks," *IEEE Trans. Mobile Computing*, vol. 6, no. 8, pp. 888-902, Aug. 2007.

[2] Y. Zhang, W. Liu, W. Lou, and Y. Fang, "MASK: Anonymous On-Demand Routing in Mobile Ad Hoc Networks," *IEEE Trans. Wireless Comm.*, vol. 5, no. 9, pp. 2376-2385, Sept. 2006.

[3] Y. Qin and D. Huang, "OLAR: On-Demand Lightweight Anonymous Routing in MANETs," *Proc. Fourth Int'l Conf. Mobile Computing and Ubiquitous Networking (ICMU '08)*, pp. 72-79, 2008.

[4] M. Blaze, J. Ioannidis, A. Keromytis, T. Malkin, and A. Rubin, "WAR: Wireless Anonymous Routing," *Proc. Int'l Conf. Security Protocols*, pp. 218-232, 2005.

[5] A. Boukerche, K. El-Khatib, L. Xu, and L. Korba, "SDAR: A Secure Distributed Anonymous Routing Protocol for Wireless and Mobile Ad Hoc Networks," *Proc. IEEE 29th Ann. Int'l Conf. Local Computer Networks (LCN '04)*, pp. 618-624, 2004.

[6] S. Seys and B. Preneel, "ARM: Anonymous Routing Protocol for Mobile Ad Hoc Networks," *Proc. IEEE 20th Int'l Conf. Advanced Information Networking and Applications Workshops (AINA Workshops '06)*, pp. 133-137, 2006.

[7] R. Shokri, M. Yabandeh, and N. Yazdani, "Anonymous Routing in MANET Using Random Identifiers," *Proc. Sixth Int'l Conf. Networking (ICN '07)*, p. 2, 2007.

[8] R. Song, L. Korba, and G. Yee, "AnonDSR: Efficient Anonymous Dynamic Source Routing for Mobile Ad-Hoc Networks," *Proc. Third ACM Workshop Security of Ad Hoc and Sensor Networks (SASN '05)*, pp. 33-42, 2005.

[9] M. Reed, P. Syverson, and D. Goldschlag, "Anonymous Connections and Onion Routing," *IEEE J. Selected Areas in Comm.*, vol. 16, no. 4, pp. 482-494, May 2002.

[10] D. Chaum, "Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms," *Comm. ACM*, vol. 24, no. 2, pp. 84-88, 1981.

[11] J. Raymond, "Traffic Analysis: Protocols, Attacks, Design Issues, and Open Problems," *Proc. Int'l Workshop Designing Privacy Enhancing Technologies: Design Issues in Anonymity and Unobservability*, pp. 10-29, 2001.

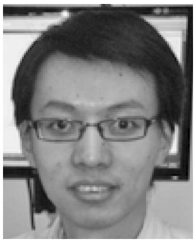
[12] W. Dai, "Two Attacks against a PipeNet-Like Protocol Once Used by the Freedom Service," http://weidai.com/freedom-attacks.txt, 2013.

[13] X. Wang, S. Chen, and S. Jajodia, "Network Flow Watermarking Attack on Low-Latency Anonymous Communication Systems," *Proc. IEEE Symp. Security and Privacy*, pp. 116-130, 2007.

[14] M. Reiter and A. Rubin, "Crowds: Anonymity for Web Transactions," *ACM Trans. Information and System Security*, vol. 1, no. 1, pp. 66-92, 1998.

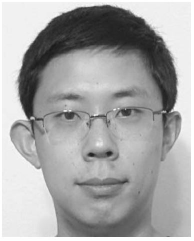
[15] M. Wright, M. Adler, B. Levine, and C. Shields, "The Predecessor Attack: An Analysis of a Threat to Anonymous Communications Systems," *ACM Trans. Information and System Security*, vol. 7, no. 4, pp. 489-522, 2004.

[16] D. Figueiredo, P. Nain, and D. Towsley, "On the Analysis of the Predecessor Attack on Anonymity Systems," technical report, Computer Science, pp. 04-65, 2004.
[17] G. Danezis, "Statistical Disclosure Attacks: Traffic Confirmation in Open Environments," *Proc. Security and Privacy in the Age of Uncertainty (SEC '03),* vol. 122, pp. 421-426, 2003.
[18] G. Danezis and A. Serjantov, "Statistical Disclosure or Intersection Attacks on Anonymity Systems," *Proc. Sixth Information Hiding Workshop (IH '04),* pp. 293-308, 2004.
[19] G. Danezis, C. Diaz, and C. Troncoso, "Two-Sided Statistical Disclosure Attack," *Proc. Seventh Int'l Conf. Privacy Enhancing Technologies,* pp. 30-44, 2007.
[20] C. Troncoso, B. Gierlichs, B. Preneel, and I. Verbauwhede, "Perfect Matching Disclosure Attacks," *Proc. Eighth Int'l Symp. Privacy Enhancing Technologies,* pp. 2-23, 2008.
[21] D. Huang, "Unlinkability Measure for IEEE 802.11 Based MANETs," *IEEE Trans. Wireless Comm.,* vol. 7, no. 3, pp. 1025-1034, Mar. 2008.
[22] D. Chaum, "The Dining Cryptographers Problem: Unconditional Sender and Recipient Untraceability," *J. Cryptology,* vol. 1, no. 1, pp. 65-75, 1988.
[23] T. He, H. Wong, and K. Lee, "Traffic Analysis in Anonymous MANETs," *Proc. Military Comm. Conf. (MILCOM '08),* pp. 1-7, 2008.
[24] Y. Liu, R. Zhang, J. Shi, and Y. Zhang, "Traffic Inference in Anonymous MANETs," *Proc. IEEE Seventh Ann. Comm. Soc. Conf. Sensor Mesh and Ad Hoc Comm. and Networks (SECON '10),* pp. 1-9, 2010.
[25] J. Wexler, "All About Wi-Fi Location Tracking," *Network World,* http://features.techworld.com/mobile-wireless/2374/all-about-wi-fi-location-tracking/, 2004.
[26] Scalable Network Technologies, "QualNet Simulator," http://www.qualnetcomm.com/, 2008.



Yang Qin received the BS degree from the University of Science and Technology of China in 2006 and the PhD degree from Arizona State University. He is currently working for Facebook. His research primarily focuses on wireless ad hoc networks, including anonymous communications focusing on anonymous routing protocol design and traffic analysis attacks, ad hoc network security, and cloud computing.

Dijiang Huang received the BS degree from the Beijing University of Posts and Telecommunications, China, in 1995 and the MS and PhD degrees from the University of Missouri-Kansas City in 2001 and 2004, respectively. He is an associate professor at the School of Computing Informatics and Decision System Engineering, Arizona State University. His current research interests include computer networking, security, and privacy. He is an associate editor of the *Journal of Network and System Management* and an editor of the *IEEE Communications Surveys and Tutorials*. He has served as an organizer for many international conferences and workshops. His research is supported by the US National Science Foundation, the US Office of Naval Research (ONR), the US Army Research Office, NATO, and the Consortium of Embedded Systems. He is a recipient of the ONR Young Investigator Program Award. He is a senior member of the IEEE.



Bing Li received the BS degree from Shandong University in 2008 and the MS degree from the Beijing University of Posts and Telecommunications in 2011. He is currently working toward the PhD degree at Arizona State University. His research interests include ad hoc network security and cloud computing. He is a student member of the IEEE.