Secure Routing based on Social Similarity in Opportunistic Networks

Lin Yao, Yanmao Man, Zhong Huang, Jing Deng, Xin Wang

***Abstract*—The lack of pre-existing infrastructure or dynamic topology makes it impossible to establish end-to-end connections in Opportunistic Networks (OppNets). Instead, a store-and-forward strategy can be employed. However, such loosely-knit routing paths depend heavily on the cooperation among participating nodes. Selfish or malicious behaviors of nodes impact greatly on the network performance. In this paper, we design and validate a dynamic trust management model for secure routing optimization. We propose the concept of incorporating social trust into the routing decision process and design a Trust Routing based on Social Similarity (TRSS) scheme. TRSS is based on the observation that nodes move around and contact each other according to their common interests or social similarities. A node sharing more social features in social history record with the destination is more likely to travel close to the latter in the near future and should be chosen as the next-hop forwarder. Furthermore, social trust can be established based on an observed node's trustworthiness and its encounter history. Based on direct and recommended trust, those untrustworthy nodes will be detected and purged from the trusted list. Since only trusted nodes' packets will be forwarded, the selfish nodes have the incentives to well-behave again. Simulation evaluation demonstrates that TRSS is very effective in detecting selfish or even malicious nodes and achieving better performance.**

***Index Terms*—opportunistic network; social similarity; incentive; trust routing**

I. Introduction

OPPORTUNISTIC Networks (OppNets) are a sub-class of Delay-Tolerant Networks (DTNs) where communication opportunities are intermittent and an end-to-end path between source and destination may never exist [1]. OppNets allow the devices in different regions to interconnect by sending messages in a store-and-forward fashion. This makes traditional routing techniques based on the existence of an end-to-end connection unsuitable for OppNets.

Different approaches have been proposed in recent years to address the routing issues in OppNets. *Flooding* is probably the most straightforward one. In flooding, a message is broadcasted from every node to all its encounters until the message reaches a predefined maximum hop count (i.e., Time-To-Live or TTL value) or the destination. In order to reduce the overhead of storage usage and communication bandwidth, routing can be performed more selectively based on the *context information*. The widespread use of smartphones and tablet PCs enriches the collection of social context information. People often move around and come into contact with each other based on their social attributes such as workplace, interest, and friendship. These social contexts play an important role in improving the routing performance [2].

Despite various efforts in developing routing schemes for OppNets, they do not consider relay nodes' trustworthiness. Although nodes with closer social relationship are more likely to meet thus increasing the chance of message delivery, it does not prevent the selfish nodes from dropping packets to conserve their own resources such as buffer space and energy nor does it prevent the malicious nodes from attacking the networks. The assumption on full cooperation from nodes with social relationship helps improve the routing performance on the one hand, but also makes the transmission more vulnerable to the attacks if no protection is provided in the routing scheme. It is thus critical to detect and isolate the selfish and malicious nodes from the well-behaving ones [3], [4].

Because of the unique properties of OppNets, the incentive-based techniques trying to motivate cooperative packet forwarding in mobile ad hoc networks are not suitable for OppNets. The power and networking constraints in opportunistic networks that are not often found in other networks requires efficient usage of each node's battery and storage. Incentive mechanisms in mobile ad hoc networks relying on connected end-to-end paths for control plane messages will then not work well in OppNets with potentially selfish or malicious nodes [5].

In this work, we propose a dynamic trust management scheme to deal with both malicious and selfish misbehaving nodes. We develop a Trust Routing method based on Social Similarity (TRSS) for OppNets. Our trust model can evaluate each node's trustworthiness based on the direct and indirect relationship between nodes. The nodes with lower trust values can be detected and prevented from working with other normal nodes for packet delivery. This trust management scheme not only helps protect the network from transmission disruptions but also serves as an incentive for nodes to well-behave and collaborate in packet transmissions in OppNets.

Our main contributions of the paper are as follows:

(i) We propose to combine similarity from social features and node behaviors into a composite metric in order to assess the trust of a node in OppNets. We also

Lin Yao, Yanmao Man, Zhong Huang are with School of Software, Dalian University of Technology, and Key Laboratory for Ubiquitous Network and Service Software of Liaoning Province in China. Lin Yao is also with the Department of Electrical and Computer Engineering, Stony Brook University, U.S.A. Yanmao Man is also with the Department of Computer Science, University of North Carolina at Greensboro, U.S.A. (email:yaolin@dlut.edu.cn, y_man@uncg.edu, huang.zhong09@gmail.com).

Jing Deng is with the Department of Computer Science, University of North Carolina at Greensboro, U.S.A. (email:jing.deng@uncg.edu).

Xin Wang is with the Department of Electrical and Computer Engineering, Stony Brook University and, also with the Center of Excellence in Wireless & Information Technology of New York State, U.S.A. (email: xwang@ece.sunysb.edu).

design a new encounter-based acknowledgment scheme to evaluate node behaviors.

(ii) We incorporate trust into the routing decisions in OppNets to protect against selfishness and various trust-related attacks, including Promise-Then-Drop, Trust-Boosting, and Defamation, some of which are collaborative attacks.

(iii) We propose a scheme to effectively evaluate node behaviors based on direct and indirect observations, which are facilitated with a novel scheme for behavior feedbacks in the presence of opportunistic links in OppNets. The behavior evaluation also serves as an incentive mechanism to effectively reward the good behaviors of normal nodes while purging selfish and malicious nodes from normal network operation.

(iv) We evaluate the performance of our trust-based OppNet routing protocol along with the proposed dynamic trust management scheme through extensive simulations. Compared to non-trust based protocol (PROPHET [6]) and social-aware routing algorithm (dLife [7]), TRSS improves the delivery ratio about 25% while reducing the number of duplicated packets by about 85% and improves delivery latency for about 10%.

The rest of this paper is organized as follows. In Section II, we discuss the related work. The system model is given in Section III. We present the details of our TRSS scheme in Section IV. Simulation results and discussions are presented in Section V. Finally, we conclude our work in Section VI.

II. Related Work

Based on different data forwarding behavior, we classify the existing routing algorithms for OppNets into two categories: *Flooding-based*, and *Context-based*.

Flooding-based Routing In flooding-based routing schemes, messages are flooded during each user encounter. The classic Epidemic routing algorithm [8] guarantees that all nodes will eventually receive all messages at the cost of large transmission overhead and required buffer size. In order to reduce such overheads, researchers designed schemes to control the message replication process based on different factors, such as time-to-live (TTL), kill-time, and passive cure [9]. For instance, Spyropoulos et al. proposed Spray and Wait [10] with two phases: In the "spray" phase, the source node sprays L copies of the messages over the network; In the "wait" phase, nodes with copies of messages perform direct transmission until the messages find the destination. Similarly, Spray and Focus [11] sprays forwarding tokens and only nodes with forwarding tokens are allowed to forward messages to different relays.

Flooding-based schemes generally can achieve higher message delivery rate by flooding packets throughout the network. However, the sometimes prohibitive communication cost and required buffer size render them inappropriate in most OppNets.

Context-based Routing In context-based routing schemes, next-hop relays are chosen based on user context information, such as mobility patterns, encounter history, and social relationship. Usually, the set of message carriers are selected based on the estimation/prediction of the message delivery probability, depending on encounter frequency, aging encounters, aging messages, and resource allocation [12]. Context-based routing schemes can be further divided into non-incentive context routing and incentive context routing.

a) Non-incentive context-based routing: Non-incentive context-based routing schemes are usually based on the assumption that nodes (users) are trustworthy. For instance, in CAR [13], routing choices are based on candidates' future encounter probability with the destination. Such probabilities are estimated from past encounter frequency [14]–[16], aging encounters (a more recent encounter is preferred) [17], and resource consumption [18], [19].

There are non-incentive context-based routing schemes that consider different social properties, such as Community, Centrality, Friendship, or Similarity. For example, in Bubble Rap [20], nodes belonging to the same social community as the destination are preferred. If no such nodes are found, preference is given those nodes with higher probabilities of reaching the destination community. Friendship-based routing [21] prefers friends, which are defined as those users with frequent, regular, and long-lasting contacts with the destination. In [22], a reputation approach is used to forward messages to nexthop nodes belonging to the same group or with the same label as the destination. In [23], centrality and similarity are used to predict the probability that potential relay nodes may meet the destination. In [24]–[27], it is observed that people having similar interests tend to meet more often and social-aware routings have been proposed. Similar approaches include HiBOp [28] using history of social relationships among nodes and social feature extraction using entropy [29].

Compared with flooding-based routing, these schemes use context information to reduce the routing overhead. However, they do not consider the selfish or malicious behaviors from users.

b) Incentive context-based routing: Incentive context-based routing schemes try to improve route robustness in the presence of selfish or malicious nodes. Different assumptions are made, e.g., credit-based mechanisms [30], [31] assuming centralized credit banks and barter-based mechanisms [32], [33] assuming nodes exchanging equal amounts of services.

Some incentive schemes are based on reputation, which is calculated based on the collaboration level of a node with other entities. For example, in SORI scheme [4], the reputation of a node is quantified by objective measures, and the propagation of reputation is efficiently authenticated by one-way hash chains. In [34], a recommendation model is proposed to distinguish truth-telling and lying agents so as to detect the selfish nodes, obtain true reputation of an agent, and ensure reliability against attacks of defame and collusion. Some cooperative mechanisms operate on nodes' transit behavior [3], [35]. Nodes are given different priority in transmissions based on their forwarding behaviors. Some incentive schemes consider social-network information to improve performance. IRONMAN [36] uses a record of the social-network data from self-reported social networks to bootstrap an incentive mechanism. Self-reported social

networks can be obtained through interview, or from online social networks (e.g., Facebook friends lists). SSAR [37] uses social ties to cope with selfishness; i.e., nodes are willing to forward packets for those with whom they have social ties such as family members and friends even at the cost of their own resources. In [36], [37], only social relationship is considered during the relay selection, while the selfishness of the nodes is ignored.In contrast, our paper considers the individual selfishness, i.e., a selfish node may refuse to forward packets for anyone else when it has limited resources to use. A node chooses its next-hop relays considering both the social similarity between a candidate relay and the destination as well as the relay behaviors.

Credit-based mechanisms are hard to achieve in intermittently-connected OppNets because they require a secure trust mechanism. Barter-based mechanisms' performance can degrade sharply with networks with heterogeneous traffic from different users. Simulation results show that the reputation-based strategies can work well even if many nodes drop packets [38], [39]. However, existing incentive schemes either only check whether an intermediate node forwards messages to other nodes or fail to use trust to protect against collaborative attacks.

Summary of Related Work In order to have efficient routing in social OppNets, social charactersitics as well as user behaviors have to be addressed simutaneously. On one hand, there have been some prior arts exploiting social characteristics such as Community, Centrality, Similarity, or Friendship to facilitate packet forwarding [20], [21], [23]–[25], [27]–[29]. They fail to address the issues caused by selfish or malicious nodes' behaviors. On the other hand, other priopr arts addressed potential attacks from selfish or malicious nodes based on prior social ties from interviews or especially online social networks [36], [37], [40]. Instead, our approach is to use both social similarity as well as nodes' past forwarding history for routing selection. Selected forwarding nodes should have high social similarity with the packet destination (for a better chance of reaching it soon) as well as high trust value, which is constantly re-evaluated based on their forwarding history.

III. System and Attack Models

We consider an OppNet environment without assuming the existence of a trusted authority. Nodes have the following transmission characteristics [41]: Every node uses omni-directional transceivers to monitor its neighbors in promiscuous mode. All links are bi-directional and all nodes have a similar transmission range. A packet will be received/overheard by the nodes within the transmission range. We further assume that users are willing to share their social features in order to participate in cooperative forwarding in OppNets. Note that this may lead to privacy concerns as some users do not want to share some of their social features. In fact, techniques can be developed to take advantage of social features while maintaining a certain level of privacy [41], [42]. We leave the extension of our scheme to protect user privacy toward our future work.

Nodes communicate with each other through the help of other nodes when necessary. Every node forwards messages based on its trust relationship with others. A node's trust is assessed based on direct trust evaluation and indirect trust information like recommendation. The direct trust evaluation is generated from physical neighbors and the indirect trust evaluation is derived from the recommendation of other intermediate nodes. The trust of one node toward another is updated upon encounter events by combining social features with the forwarding behaviors, which will be discussed in Section IV. Packets are only forwarded to those trusted nodes.

We classify three types of nodes in the network: normal nodes, selfish nodes, and malicious nodes. A normal node possesses higher trust values, while a selfish and malicious node has lower trust value. Normal or well-behaving nodes follow the rules to help other nodes store and forward packets. Selfish nodes are differentiated from malicious nodes. Selfish nodes may drop packets out of the rational consideration, for example, in order to save their own resources. Malicious nodes, however, may intentionally break the basic network functions such as launching various attacks. In this paper, we focus on the following attacks:

(i) Promise-Then-Drop: The attacker first promises to forward packets for other nodes, receiving a higher trust from its neighbors. Then it silently drops the packets that it receives and ought to forward.
(ii) Trust-Boosting: Malicious attackers exaggerate the reputation of other malicious users by submitting biased recommendations for them. With the higher trust value, they attract more packets to drop.
(iii) Defamation: The attacker or a group of attackers try to lower the trust level of a well-behaving node by submitting bad recommendations against the target.

Note that most of the above attacks can be launched collaboratively, except maybe attack (i). For instance, a group of malicious nodes can boost their trust values artificially by providing abnormally high recommendations to each other. Similarly, they can defame a target node by submitting low recommendations.

In order to ensure efficient network operation, it is essential to have cooperation among nodes. Therefore, our major goal in this work is to design a trust routing scheme to resist the misbehaviors and to stimulate the nodes to relay packets for others. Each node executes the trust model independently. By ranking nodes based on their trust values, our TRSS scheme can effectively select the trustworthy relays for message forwarding. Furthermore, in order to provide nodes with incentives to cooperate in packet forwarding and punish misbehaving nodes, packets from nodes with very low trust values will be discarded.

IV. The Trust Routing Scheme Based on Social Similarity (TRSS)

In this section, we will elaborate our TRSS scheme. We will present a trust model to quantify the trust relationship among network entities and facilitate their safe and effective interactions. The frequently used notations in the paper are listed in Table I. In the remaining subsections, we present

TABLE I: Frequently Used Notations

Notation	Description
n_i	Node i
S,D	Source, Destination
$\eta_{i,j}$	Social similarity computed by node i toward node j
$\boldsymbol{F}_{n_i}$	The vector of social features for n_i
f_j	j-th feature
L_j	Number of values of j-th feature
N_{jk}	Number of the value of L_j that is k in *HISTORY-TABLE*
$\boldsymbol{M}$	The matching vector of social features
$T_{i,j}$	The evaluated trust value from n_i to n_j
FEATURE -TABLE	A table recording a node's social features
HISTORY -TABLE	A table recording the social encountering history
TRUST-TABLE	A table recording the trust values
EACK	A table recording the information needed for Encounter-based Acknowledgment
PROMISE -THRESHOLD	The trust threshold above which a promise may be made to help forward a data packet
SELECTION -THRESHOLD	The trust threshold above which a selection may be made to forward a data packet

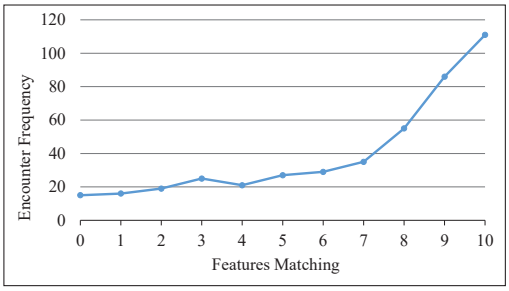


Fig. 1: Encounter Frequency changes along with Social Similarity

TABLE II: *An example for Feature-Table*

	Description	Value
f_1	Country (Living)	China
f_2	City (Working)	Beijing
f_3	Nationality	Chinese
f_4	Languages	Chinese, English
$\cdots$	$\cdots$	$\cdots$
f_i	Affiliation	IBM
$\cdots$	$\cdots$	$\cdots$
f_r	Position	Manager

our algorithms on the calculation of social similarity, establishing trust model, making routing decision, Encounter-based Acknowledgment, and protection against various attacks.

A. Computing Social Similarity

The TRSS scheme is motivated from several social contact networks, such as the Infocom 2006 traces, where people come in contact with each other more frequently if they have more social features in common [29]. We further illustrate this feature in Figure 1 that is derived from the data of cambridge/haggle trace. The X-axis shows the number of features that people share in common, while the Y-axis shows the encounter frequency, defined as the ratio of the total number of encounters and the number of node pairs. Taking point (4, 20) as an example, suppose there are 120 pairs of people who have X = 4 social features in common and their total encounter times is 2400, we have $Y = \frac{2400}{120} = 20$. Notice that, in Figure 1, when the number of common features increases, the encounter frequency increases as well, which shows that people will meet more frequently if they have more social features in common. Therefore, if social features can be exploited to select forwarding nodes, data packets will have a higher chance of reaching the destination. We further exploit this, and introduce our scheme for determining social similarity based on encounter history, there are some basic terminologies.

- ***Social Feature Vector*:**
$$\boldsymbol{F} = \langle f_1, f_2, \ldots, f_i, \ldots, f_r \rangle$$
$\boldsymbol{F}$ is a vector of social features with r elements from f_1 to f_r, where f_i denotes the i-th social feature as shown in *FEATURE-TABLE* (Table II). Each feature has multiple possible values. For example, f_4 can refer to Languages with the values of "Chinese" or "English".
- ***Social History Record*:** Our scheme requires that neighboring nodes exchange their social features to form the social history record. This table is used to record encountered nodes' social features. Take Table II for an example, we have the following social features, $\langle$ Country, City, Nationality, Languages, Affiliation, Position$\rangle$. The social features of a node n_i, $\boldsymbol{F}_i = \langle f_1, f_2, f_3, f_4, f_5, f_6 \rangle$, may be $\langle$"China", "Beijing", "Chinese", "Chinese, English", "IBM", "Manager"$\rangle$. After exchanging social features with its neighbors, n_i's *HISTORY-TABLE* in Table III is updated based on the received social features from all neighboring nodes. For example, if n_i has updated its record f_2 with "Beijing = 2", it means that it has met two users from Beijing.
- ***Social Similarity*:** This parameter is applied to represent the degree of similarity between a node n_i and the destination D. A higher similarity implies that n_i has a better chance of meeting D.

In the TRSS scheme, we use the following method to compute social similarity: a source node broadcasts D's social profile, $\boldsymbol{F}_D = \langle f_{d_1}, f_{d_2}, \cdots, f_{d_r} \rangle$. Each neighbor computes the social similarity according to the received profile and its own *HISTORY-TABLE*. Assuming that feature f_j has L_j different values and each value corresponds to N_{jk} encounters, where $k = 1, 2, \cdots, L_j$. We apply a parameter w_j to denote the ratio between the number of nodes with a certain value of feature f_j and all the encountered nodes. n_i can compute w_i as:

$$w_j = \frac{N_{jk}}{\sum_{k=1}^{L_j} N_{jk}} \tag{1}$$

For example, social feature f_j is "Affiliation". If $\sum_{k=1}^{L_j} N_{jk}$ is equal to 10, this neighbor has encountered 10 users who

TABLE III: *An example for History-Table*

Feature	**Value1**	**Value2**	···
Country	China (2)	United States (3)	···
City	Beijing (2)	New York City (3)	···
Nationality	Chinese (2)	United Kingdom (3)	···
Language	Chinese (2)	English (5)	···
Affiliation	IBM (1)	PKU (1)	···
Position	Manager (3)	Teacher (2)	···
···	···	···	···

work at different affiliations. If N_{jk} is equal to 4 and k corresponds to the value "IBM", it means this neighbor has met 4 users from IBM.

Combining all different features from D's profile, the social similarity between n_i and D is

$$\eta_{i,D} = 1 - \sqrt{\frac{\sum_{j=1}^{r} \alpha_j (1 - w_j)^2}{r}}, \tag{2}$$

where α_j is the weight for the feature j and $\sum_{j=1}^{r} \alpha_j = r$.

In the example of Table III, assume that D has a social feature of $\boldsymbol{F} = \langle$"China","Beijing", "Chinese","Chinese, English","IBM", "Manager"$\rangle$. For the feature "Country", there are two encounters from "China" and three from "United States" in n_i's *HISTORY-TABLE*. Thus, $w_1 = \frac{2}{2+3} = \frac{2}{5}$. Similarly, for feature "Language", we can get $w_4 = \frac{2+5}{2+5} = \frac{7}{7} = 1$. Then, we get

$$\mathbf{M} = \langle w_1, w_2, w_3, w_4, w_5, w_6 \rangle = \langle \frac{2}{5}, \frac{2}{5}, \frac{2}{5}, 1, \frac{1}{2}, \frac{3}{5} \rangle \tag{3}$$

Assuming $\alpha_j = 1$ for all $j = 1, 2, \cdots, r$, the social similarity between n_i and D is 0.502 based on Eq.(2).

B. Trust Model

The concept of ***trust*** is originated from social sciences and defined as the degree of subjective belief about the behaviors of a particular entity [43]. It is a relationship between the evaluating and evaluated node. The likelihood that the evaluating node expects the evaluated one to offer certain services is a trust value within $[0, 1]$, while 0 indicates no trust and 1 means full trust.

In this section, we aim to exploit a trust model in the forwarding-node selection process. Based on the observations of routing behaviors, our trust model to reward or punish the nodes' behavior by increasing or decreasing the trust value. Only nodes with high trust values are chosen as next hops, while those nodes with low trust values, as a punishment, will not be served. Our trust management framework consists of three building blocks, as illustrated in Figure 2. The trust management block serves as the interface between the trust record and applications that request the trust value or the routing behavior feedbacks. The trust record is formulated through the trust maintenance process, which obtains direct trust values from interactions between neighbors and indirect trust values from recommendations. Based on the trust record, misbehaving nodes will be detected and excluded from normal network operations and well-behaving nodes will be rewarded by collaborating with others in packet transmissions.

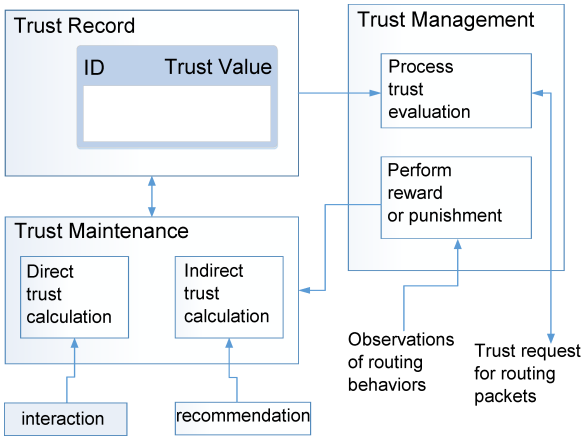


Fig. 2: Trust Model

1) Direct Trust Evaluation: A node can directly form its trust level on another node based on its interaction with it. When two users (nodes) encounter first, each can determine the other's trust according to the social behaviors/characteristics . For example, if both come from the same university, they would trust each other more than the one from a third party which does not have any common social feature. Assume that two nodes n_a and n_b come into contact with each other. Based on the *HISTORY-TABLE*, we initialize the trust value $T_{b,a}$ from n_b to n_a as the social similarity $\eta_{b,a}$ calculated based on Formula (2).

Then, $T_{b,a}$ is updated dynamically based on the packet relaying behaviors of node n_a as below:

$$T_{b,a}^{(new)} = f(f^{-1}(T_{b,a}) \pm T_{a,b}). \tag{4}$$

If n_a helps n_b to forward packets, the trust value from n_b to n_a, which is $T_{b,a}$, will increase; otherwise, $T_{b,a}$ will decrease. Thus, we should adopt an incentive factor which plays the function of rewarding or punishing n_a. To reduce the simulation complexity, we simply adopt an incentive factor $T_{a,b}$. The initial value of $T_{a,b}$ is computed in Eq. (1). Furthermore, to guarantee $T_{a,b}^{(new)}$ within $[0, 1]$, a continuous function $f()$ is applied to normalize $T_{b,a}^{(new)}$ to be within $[0, 1]$.

In our simulation, we choose the following function

$$f(x) = \begin{cases} \frac{1}{2} e^{\beta x} & x < 0 \\ 1 - \frac{1}{2} e^{-\beta x} & x \geq 0, \end{cases} \tag{5}$$

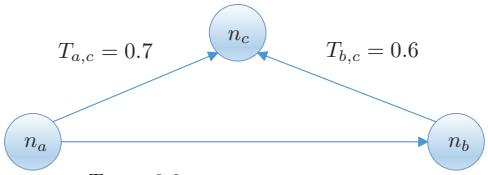
where β is a constant factor that impacts the $T_{b,a}$ adjustment rate. $T_{b,a}$ approaches 0 or 1 if one node keeps the same routing behavior such as selfish or cooperative.

Based on $f(x)$, Eq. (4) then becomes

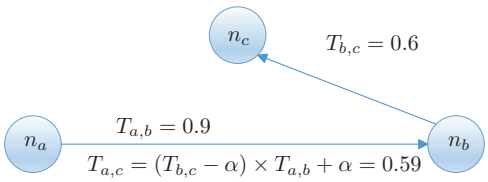
$$T_{b,a}^{(new)} = \begin{cases} T_{b,a} \cdot e^{\beta T_{a,b}} & \tilde{x} < 0 \\ 1 - (1 - T_{b,a}) e^{\pm \beta T_{a,b}} & \tilde{x} \geq 0, \end{cases} \tag{6}$$

where $\tilde{x} = f^{-1}(T_{b,a}) \pm T_{a,b}$.

$f()$ has a steep slope in Equation (5) when x is around 0, which indicates that the trust value is easily changed by some bad or good behaviors at that time. It will benefit the new nodes so that they can be included into data forwarding quickly. So $f()$ helps to evaluate a node's trust more effectively.



(a) Trust re-evaluation on an existing node



(b) Trust valuation on an unknown node when $\alpha = 0.5$

Fig. 3: Illustration of (re)evaluating trust of different nodes

2) Indirect Trust Evaluation: A node may not have enough chance to directly observe the behaviors of an encountered node. In order to build more reliable trust values, our trust model incorporates into its trust calculation the indirect or recommended trust values obtained from an intermediate entity. Specifically, after two encountering nodes n_a and n_b exchange their *TRUST-TABLE*s, they will update the trust value of a third node n_c. Node n_a recalculates its trust on n_c based on the following equation

$$T^{(new)}_{a,c} = (T_{b,c} - T_{a,c}) \cdot T_{a,b} + T_{a,c}, \tag{7}$$

where $T_{a,c}$ is the current trust value that node n_a has on the node n_c. If node n_c is a newly encountered node, $T_{a,c}$ is set to an initial value of $\alpha = 0.5$. $T_{b,c}$ is the recommendations from n_b. The impact on $T_{a,c}$ by the indirect trust value $T_{b,c}$ depends on the trust of n_a towards n_b, and increases with $T_{a,b}$. Figure 3 illustrates two cases that n_a computes $T_{a,c}$ based on the recommendation trust $T_{b,c}$. In Figure 6(a), n_a has already recorded $T_{a,c}$ in its *TRUST-TABLE*. In Figure 6(b), n_a has no record on $T_{a,c}$. In Figure 6(a), n_a updates $T_{a,c}$ based on the recommendation $T_{b,c}$, $T_{a,b}$ and the old $T_{a,c}$. While in Figure 6(b), n_c is a newly encountered node, so $T_{a,c}$ is set to an initial value of $\alpha = 0.5$.

$T_{a,b}$ in Eq. (7) helps to defeat collaborative attacks. Suppose two attackers n_b and n_c try to increase their trust values $T_{a,b}$ and $T_{a,c}$. n_b and n_c recommend fake and high trust value $T_{b,c}$ and $T_{c,b}$ to n_a. However, n_a cannot believe them or generate high trust value in Eq. (7), because the original trust value $T_{a,b}$ or $T_{a,c}$ is not high.

C. Routing Decision

In order to achieve efficient data forwarding in OppNets, it is critical for a node to select the next-hop relays which can continue forwarding data towards the destination. A straightforward method is to simply choose the node with the highest trust value. However, the nodes with high trust values may not be the ones with high social similarity, and the conflicting results may compromise the routing performance.

Our selection algorithm concurrently considers trust value and social profile. Suppose n_a and n_b meet, n_a can determine whether it should choose n_b as the next-hop node to forward its data packet for a destination D based on the following steps:

(1) The two nodes exchange their *FEATURE-TABLE* and update their own *HISTORY-TABLE*.
(2) The two nodes exchange their *EACK*, based on which they execute the Encounter-based Acknowledgment we will discuss in Section IV-D.
(3) The two nodes exchange their *TRUST-TABLE* and then update their individual tables based on the indirect trust evaluation.
(4) Based on the trust value $T_{b,a}$, n_b decides whether to help n_a to forward the packet toward D. If $T_{b,a}$ is higher than the *PROMISE-THRESHOLD*, e.g., 0.5, n_b will compute the similarity value $\eta_{b,D}$ and reply $\eta_{b,D}$ and $T_{b,a}$ to n_a as a "promise".
(5) Based on both $\eta_{b,D}$ and $T_{a,b}$, n_a decides whether to choose n_b as the forwarder. If $\eta_{b,D} > \eta_{a,D}$ and $T_{a,b}$ is higher than the *SELECTION-THRESHOLD*, e.g., 0.5, it will forward the packet to n_b.

D. Encounter-based Acknowledgment

Algorithm 1 Encounter-based Acknowledgment

Data Structure:
Multi-Map $EACK\langle key, value\rangle$ where
key is the node waiting for the acknowledgment
$value$ is the node to be confirmed of behavior and certain msg_{id}

Function: recvMessage(msg)
for each n_i in $routingPath$ of msg **do**
  $wait\_meet \leftarrow n_i$
  $wait\_confirmed \leftarrow \{n_{i+1}, msg_{id}\}$
  store $\langle wait\_meet, wait\_confirmed\rangle$ into $EACK$
end for
Function: encounter(n_B):
for each $mapping\langle key, value\rangle$ in $EACK$ **do**
  if $key = n_B$ **then**
    n_B.updateTrustTable($value$);
  end if
end for

In order to evaluate the behavior of a node and form the trust value, an evaluating node needs to know if packets have actually been forwarded by the evaluated node. Unlike conventional mobile ad hoc networks where a node can overhear the transmission from its next-hop node, the learning of the packet transmission status in OppNets is nontrivial. A next-hop node may not be able to send out the packets immediately but has to wait until it meets a better forwarder or the destination itself. Its transmission is generally not overheard by the sender directly. In addition, a receiver may never come close to the sender in the near future or ever.

There are some works trying to address this issue, by mostly focusing on how to estimate current network topology more

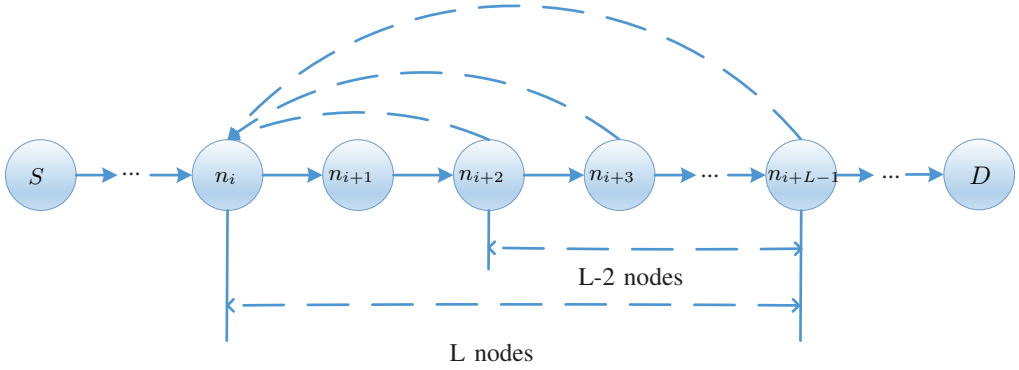


Fig. 4: General Encounter-based Acknowledgment Scenario

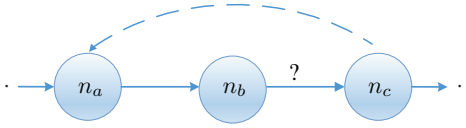


Fig. 5: A Special Encounter-based Acknowledgment Scenario When $L = 3$

accurately [18], [44]. These include relying on friends with similar interests [45], incentive-based ACK [36], [46], virtual checks [46], full encounter history [36]. Instead, what we need here is a light-weight acknowledgment mechanism to provide feedback for delivery promises.

In our TRSS scheme, we develop an acknowledgment technique called Encounter-based Acknowledgment, which does not require active packet transmission until two nodes meet. Each packet carries the IDs of previous $L-1$ hops of the path that it has gone through. A general scenario is shown in Figure 4. The packet forwarding of n_i is known by L nodes, $n_i, n_{i+1}, \cdots n_{i+L-1}$. The last $L-2$ nodes on the list can provide Encounter-based Acknowledgment to n_i to confirm the forwarding behavior of n_{i+1}. Obviously, the chance for the sender to receive the acknowledgment increases with the value of L. By adjusting L, the network load of acknowledgment can be changed. In addition, identity-based security can be added to ensure that EACK counterfeits do not have a opportunity. Or, a simple hash can be computed of the payload and inserted into the packet record as well as EACK, resulting in a small amount of extra overhead.

Algorithm 1 illustrates our scheme in details.

Figure 5 shows a special scenario with $L = 3$, where n_c may eventually meet n_a, which will receive the acknowledgment from n_c about the message delivery status. In order to make the acknowledgment work, n_c needs to record the ID of the message sender as well as ones carried by the message (line $1 \sim 5$ of Algorithm 1). When nodes n_a and n_c encounter, n_c provides n_a with the information that whether it has indeed received a message from n_b (line $6 \sim 10$ of Algorithm 1). If so, the trust for n_b is increased by n_a; Otherwise, the trust value will be decreased. The trust value can be updated based on Equation (4).

E. Protection Against Various Attacks

In this section, we discuss the design of our scheme in defending against some potential attacks.

i) *Protection against Promise-Then-Drop Attack:* With the help from the EACK technique, those nodes launching Promise-Then-Drop attacks will be caught and their new trust recommendations will be lowered by the detecting nodes. Note that our EACK technique cannot detect nodes with bad connections or simply fail. We argue that, as such nodes drop the promised-to-forward packets, their trust should be gradually lowered. Indeed, nodes with such problems should not promise to help other nodes.

ii) *Protection Against Trust-Boosting Attacks:* In our trust evaluation procedure, a node's trust value is assessed based on direct trust evaluation and indirect trust information like recommendations. When two nodes n_a and n_b encounter for the first time, the trust value from n_b to n_a is computed on Eq. (1). n_b must continuously evaluate n_a based on n_a's forwarding behaviors. Furthermore, n_b can directly assess its trust toward n_a and update $T_{b,a}$ after an encounter confirms the behavior of n_a.

iii) *Protection Against Defamation Attacks:* While one or several malicious nodes can artificially lower their recommendation of a target node's behavior, the target's normal behavior of forwarding data packets for other users can still earn its way back to a high trust value. In fact, the intrinsic mobility pattern in OppNets makes it possible for a victim node to "escape" from the collaborative defamation attacks from a group of malicious nodes surrounding itself and regain the trust from well-behaving users.

V. Performance Evaluations

To evaluate the performance of our trust routing model, we use the Opportunistic Network Environment (ONE) simulator [47] specially designed for opportunistic network. We use Infocom 2006 trace [48] in our simulation, including 121,029 encounters among 85 users in a period of 342915 seconds, or roughly 4 days. Instead of using all social features in the dataset, we adopt 6 informative features based on their entropies [29], which are Affiliation, City (of residence), Nationality, Language, Country (of residence), and Job Position (Student/Researcher/Professor). Each user is equipped with

TABLE IV: Simulation Setup

Number of runs	20
Simulation time	342915 seconds
Warmup time	100000 seconds
Transmission Speed	54Mbps
Transmission Range	100 Meters
Buffer size	100MB
TTL time	1433 minutes
Number of nodes	total 85
Ratio of selfish/malicious	0-80%

a wireless device with a transmission rate of 54 Mbps and a transmission range of 100 meters. One distinct packet is generated randomly with the average duration 400 to 600 seconds. The source and destination are randomly chosen from the 85 nodes. The packet size ranges from 0.5 to 1 MBytes. All simulation results are the average of 20 random runs.

We compare our proposed scheme with IRONMAN [36], dLife [7] and Epidemic [8]. IRONMAN uses a record of the social-network data from self reported social networks to bootstrap an incentive mechanism, while dLife is a social-aware routing algorithm based on daily routines and considers time-evolving social ties between members. Epidemic is a flooding-based routing scheme. We compare these schemes using the following eight different metrics :

- *Delivery Ratio*: The ratio of data packets delivered to destination nodes before they are removed from the network due to an expired TTL. This is obvious necessary to gauge how efficient a routing scheme is.
- *Delivery Latency*: The time duration from a message is generated until it is received.
- *Dropped Packets*: The number of packets that are dropped maliciously by misbehaving nodes.
- *Delivery Cost*: The average number of duplicated packets divided by the numbers of packets that successfully reach the destination.
- *Detection Time*: The time consumption to detect a misbehaving/malicious node correctly.
- *Detection Accuracy*: The proportion of misbehaving nodes that are detected correctly.
- *Average Trust Value (ATV)*: The average trust value from all nodes towards one node or a group of nodes under the evaluation.

$$ATV = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m} T_{i,j}}{nm} \ ,$$

where *n* is the number of all nodes, and *m* is the number of evaluated nodes like malicious nodes or selfish nodes.

- *ACK Delivery Ratio (ADR)*: The average number of ACK delivered in the network for each packet that is forwarded successfully to next-hop and covers at least 3 hops.

$$ADR = \frac{t}{p} \ ,$$

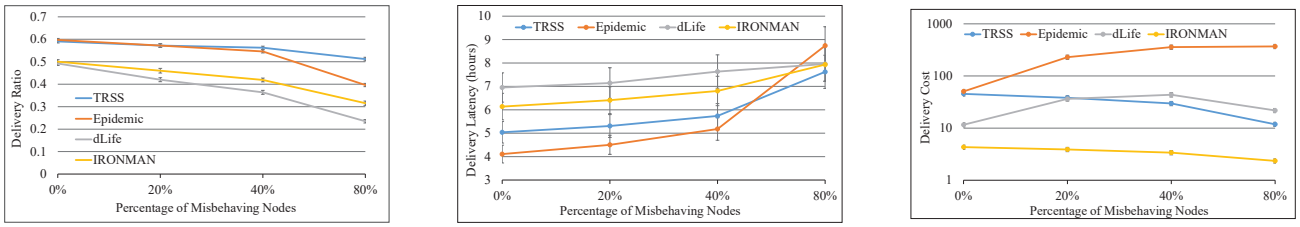
where t is the number of ACK delivered and p is the number of packets that are forwarded successfully to the next-hop and cover at least 3 hops.

A. Impact of Different Percentage of Misbehaving Nodes

In this section, we compare TRSS scheme with other related schemes under different numbers of misbehaving nodes. As discussed in Eq. (6), the initial trust value is very close to 0.5 when α is 0.5. Also, $f(x)$ gets the maximum slope value with $y = 0.5$. Thus, every node can change its trust value obviously by a few behaviors such as simply helping or refusing to forward packets. Consequently, 0.5 can be seen as a boundary to discriminate the good and bad nodes. The trust value of good nodes is higher than 0.5, or vice versa. So the *PROMISE-THRESHOLD* and *SELECTION-THRESHOLD* would better as 0.5. Because of the simulation time, the distribution of trust value would be appropriate with β = 0.06 based on Equation 4 and 5.

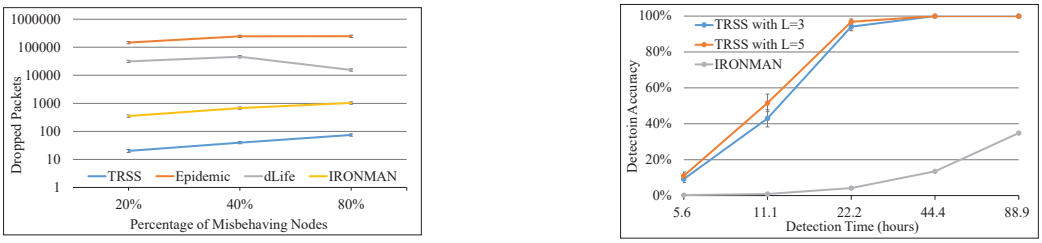
(1) Delivery Ratio. Figure 6a shows that the delivery ratio of TRSS, Epidemic, dLife and IRONMAN decreases with the number of misbehaving nodes, because more packets are dropped. Misbehaving nodes cannot be detected in Epidemic and dLife, so their delivery ratio reduces more quickly than TRSS and IRONMAN. Compared with IRONMAN, TRSS has a stable delivery ratio with TRSS decreasing by 0.07 and IRONMAN decreasing by 0.18. In TRSS, the indirect evaluation and Encountered-based Acknowledgment can detect more misbehaving nodes and choose relays with similar social background and higher encounter probability to the destination. In IRONMAN, only forwarding record is used to detect these nodes without considering trust evaluation and social features. As a result, TRSS works better in defending misbehaving nodes. Moreover, the delivery ratio of TRSS is 29% higher and 118% higher than that of Epidemic and dLife, respectively, at 80% percentage of misbehaving nodes. In ONE simulator, every node unicasts a packet to one of the neighbors in each time period (depending on the transmission rate). To improve the performance, in TRSS, a packet is forwarded to the neighbor with a higher trust value and social similarity. While in Epidemic, a packet may be forwarded to a misbehaving node, and the delivery ratio will decrease if the misbehaving node discards the packet. In other words, this time period is wasted.

(2) Delivery Latency. The average delivery latency is mainly caused by packet queuing and retransmissions. In Figure 6b, as expected, the average delay of four schemes increases with the number of misbehaving nodes. Epidemic is a flooding-based scheme and cannot detect these misbehaving so as not to avoid packets are forwarded to misbehaving nodes, which causes the delivery latency of Epidemic to increase fastest. Although dLife cannot detect misbehaving nodes, its next-hop is selected based on certain daily routines, which helps to avoid going through part of the misbehaving nodes. Thus, it has a stable delay comparatively. As more misbehaving nodes appear, TRSS and IRONMAN can detect them and they have a lower delay compared with Epidemic and dLife. As discussed before, TRSS has a strict policy to detect more



(a) (b) (c)

Fig. 6: (a) Delivery ratio comparison of TRSS, Epidemic, IRONMAN and dLife with different numbers of misbehaving nodes. 95% confidence intervals are also presented, showing the variations of all different runs of simulations. (b) Delivery latency comparison of TRSS, Epidemic, IRONMAN and dLife with different numbers of misbehaving nodes. (c) Delivery cost comparison of TRSS, Epidemic, IRONMAN and dLife with different numbers of misbehaving nodes.



(a) (b)

Fig. 7: (a) Dropped packets comparison for TRSS, Epidemic, IRONMAN and dLife with different numbers of misbehaving nodes. (b) Detection Efficiency comparison of TRSS and IRONMAN.

selfish/malicious nodes than IRONMAN. Thus, TRSS works best.

(3) *Delivery Cost.* In ONE, a packet copy will be kept in the buffer when every node receives a new packet. Though these misbehaving nodes receive packets, they will drop them without forwarding the packets. Once there are no more of these copies detected in the buffer by the upstream nodes, duplications will be generated. Epidemic has the most delivery cost in Figure 6c, because Epidemic cannot detect misbehaving nodes so that most packets are dropped as misbehaving nodes increase, leading to more duplications. Figure 6c shows the delivery cost of dLife increases as more malicious nodes drop packets when the percentage of misbehaving nodes is less than 40%. Although dLife cannot detect misbehaving nodes, dLife chooses the next hops strictly based on the social tie. When the percentage is between 40% and 80%, less and less relays are chosen so as to dropped packets rate decreases gradually, leading to a decreasing delivery cost. Consequently, dLife has a better performance than Epidemic. The delivery cost of TRSS and IRONMAN declines because both are able to detect and filter the misbehaving nodes. IRONMAN has a lower delivery cost because IRONMAN runs with the spray and wait routing scheme that‘ ‘sprays” a limited number of replicas and “waits” for the destinations, while TRSS keeps duplicating and forwarding packets until better relays are encountered. However, the delivery cost of TRSS reduces by 33.5, while IRONMAN reduces only by 1.97, which demonstrates the detection efficiency of misbehaving nodes in TRSS is higher than that in IRONMAN because TRSS adopts the indirect evaluation and the Encounter-based Acknowledgment.

(4) *Dropped Packets.* Figure 7a shows the Selfishness Cost of four schemes. TRSS has the lowest Selfishness Cost, and IRONMAN is lower than dLife and Epidemic, because TRSS can detect malicious nodes more efficiently than IRONMAN as discussed before. Compared with Epidemic, dLife chooses the next hops strictly based on the social tie. Thus, dLife has a better performance than Epidemic. Based on the results in Figure 7a, the extremely low numbers of TRSS call for some explanations: in TRSS, misbehaving nodes will be detected after they drop about two packets (from other nodes) and they will not be chosen to forward packets again. In reality, these nodes should be re-introduced into routing again through some kind of time-out and/or re-evaluation, but we leave that to our future work.

(5) *Detection Efficiency.* Since dLife and Epidemic are not able to support selfish detection, we only compare TRSS with IRONMAN in this scenario. As shown in Figure 7b, TRSS performs much better than IRONMAN. After about 44.4 hours, 100% of misbehaving nodes can be detected in TRSS, while IRONMAN only detects 13.5%. As discussed before, TRSS detects malicious nodes by indirect evaluation and Encounter-based Acknowledgment. Indirect evaluation helps to spread more nodes' trust values widely and rapidly. Encounter-based Acknowledgment mechanism helps to increase the probability of successful ACK delivery. TRSS with $L = 5$ performs slightly better than with $L = 3$, since Figure 4 has shown that the probability for 3 nodes ($L = 5$) to meet n_i is higher than that for 1 node ($L = 3$).

B. Protection Against Different Attacks

We evaluate the protections of our TRSS scheme against different attacks in the following. We set α to 0.5 and β to 0.06.

(1) *Against selfishness.* First, we set the percentage of selfish nodes to 50% and vary the packet dropping probability of selfish nodes from 0% to 100%. We evaluate the performance with our scheme in two cases, utilizing the trust model and without using the trust model. Figure 8a shows the delivery ratio decreases with dropping probability. Also, the delivery ratio with the trust model is higher than that without the trust model.

Then, we set the ratio of selfish nodes to 50% and the dropping probability of selfish nodes to 20% and 100% respectively. From Figure 8b, we can see the Average Trust Value (ATV) can effectively capture the selfish behaviors and decrease with the dropping probability.

(2) *Against Defamation and Boost.* As recommendations are taken into consideration during the indirect trust evaluation, malicious parties can carry out defamation and boost attacks. To resist these attacks, once a node is detected to be malicious because of its low trust values, its recommendations will bring little impact on its neighbors' decision making.

In Figure 8c, the ATV of certain nodes (Defamation victims or Boost victims) varies with hours, and as time goes on, their ATV approaches that of normal nodes. Initially, the ATV of defamation victims is 0.1 while the ATV of boost victims is 0.9. But the ATV from normal nodes to victims is set as 0.5. As simulations proceed, the ATV of boost victims degrades to 0.5 or so, while the ATV of defamation victims increases to 0.5. Meanwhile, the ATV of normal nodes remains 0.5 because there is no packet generated. From Figure 8c, we can clearly see that although the malicious nodes change their own trust tables maliciously, their bad influence on the whole network remains very low. As the simulation proceeds, the incorrect trust values are diluted and go back to 0.5. Thus, TRSS can defend against the defamation attack and boost attack very well.

(3) *Against Self-Promoting.* The simulation results in Figure 8c also show that TRSS can defeat self-promoting attack successfully. Even though a node can promote its trust credit, its ATV will return to the normal value, which shares the same reason with Against Defamation and Boost scenario. So its evaluation will only bring negligible influence on the other nodes in routing decision.

(4) *Against Promise-Then-Drop.* As more normal nodes become selfish, the number of dropped packets should increase sharply, although they have promised to forward packets. TRSS can resist the Promise-Then-Drop attack because TRSS updates the trust values based on EACK instead of the promise. In this simulation, we set $L = 3$. As shown in Figure 9a, the average number of dropped packets is much lower when EACK is applied to detect misbehaving nodes, since packets will not be forwarded to them. As shown in Figure 9a, a starting point lies at the point where number of attacks is 5. It shows that these selfish nodes are not detected before ACK packets return to the source.

C. Impact of the Adjustment factors

(1) *Impact of the value of β.*

In Eq. (5), the adjustment factor β can be modified to adapt the variation ratio of trust value. The impact of β on direct trust evaluation is shown in Figure 9b, where β values are set to 0.1, 0.2, 0.5 and 1, respectively. With $\beta = 0.1$, the ATV increases slowly. With $\beta = 1$, the ATV increases the fastest. Thus, the bigger the β value is, the faster the ATV value changes.

(2) *Impact of the value of L.*

As we discuss in Section IV-D, the Encounter-based Acknowledgment has a factor L, which controls the number of nodes who should reply an ACK for the pre-hop. As we can see in Figure 9c, ADR increases as L increases, as a larger value L results in more nodes along the routing path to return ACKs back to the packet sender (n_i in Figure. 4).

VI. Conclusion

In Opportunistic Networks, store-and-forward routing is adopted to deliver packets. To save the limited resources, some selfish or malicious nodes can drop data packets quietly, degrading the network performance. Thus, designing and applying an appropriate cooperation mechanism to address these attacks is important for practical network.

In this paper, we have designed and validated a trust routing protocol for OppNets. Our trust management combines routing behaviors with social similarity to obtain a composite trust metric. Every node can evaluate other nodes' trustworthiness using direct or indirect trust model. Simulation results show that our protocol can not only accurately detect the selfish or malicious nodes but also improve the delivery performance in the presence of these nodes. Our results also confirm that social information can be exploited to improve the incentive mechanism.

In the future work, we will refine our trust model based on more social network behaviors and investigate the security and privacy in social routing.

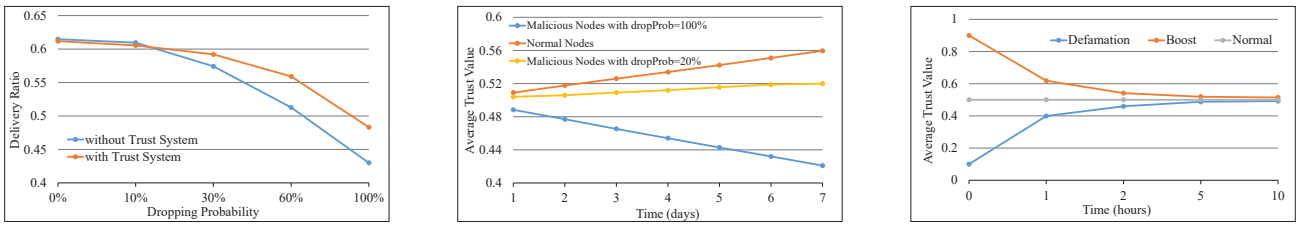
Acknowledgment

This research is sponsored in part by the National Natural Science Foundation of China (contract/grant number: No.61173179, 61103146, 61202441) and Program for New Century Excellent Talents in University (NCET-13-0083). This research is also sponsored in part supported by the Fundamental Research Funds for the Central Universities (No.DUT13JS10 and No.DUT14YQ212).

References

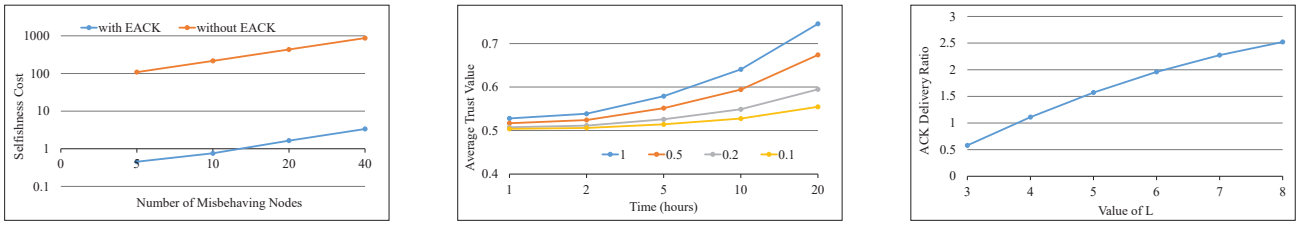
[1] C.-M. Huang, K.-c. Lan, and C.-Z. Tsai, "A survey of opportunistic networks," in *Proc. 22nd International Conference on Advanced Information Networking and Applications-Workshops*. IEEE, 2008, pp. 1672–1677.

[2] P. Hui, J. Crowcroft, and E. Yoneki, "BUBBLE Rap: Social-based forwarding in delay-tolerant networks," *Mobile Computing, IEEE Transactions on*, vol. 10, no. 11, pp. 1576–1589, 2011.



(a) (b) (c)

Fig. 8: (a) Delivery Ratio with or without trust system. (b) Average Trust Value (ATV) at different dropping probabilities. (c) Average Trust Value (ATV) under Defamation or Boost attack.



(a) (b) (c)

Fig. 9: (a) Dropped Packets changes with EACK or without EACK. (b) Average Trust Value, ATV, changes with β. (c) ACK Delivery Ratio (ADR) changes with L

[3] T. Anantvalee and J. Wu, "Reputation-based system for encouraging the cooperation of nodes in mobile ad hoc networks," in *Proc. of the IEEE International Conference on Communications (ICC)*. IEEE, 2007, pp. 3383–3388.

[4] Q. He, D. Wu, and P. Khosla, "SORI: a secure and objective reputation-based incentive scheme for ad-hoc networks," in *Proc. of the IEEE Wireless Communications and Networking Conference (WCNC)*, vol. 2. IEEE, 2004, pp. 825–830.

[5] G. Bigwood and T. Henderson, *Incentive-Aware Opportunistic Network Routing*. Springer, 2013.

[6] A. Lindgren, A. Doria, and O. Schelén, "Probabilistic routing in intermittently connected networks," *ACM SIGMOBILE mobile computing and communications review*, vol. 7, no. 3, pp. 19–20, 2003.

[7] W. Moreira, P. Mendes, and S. Sargento, "Opportunistic routing based on daily routines," in *Proc. of the 13th IEEE International Symposium on a World of wireless mobile and multimedia networks (WoWMoM)*. IEEE, 2012, pp. 1–6.

[8] A. Vahdat, D. Becker *et al.*, "Epidemic routing for partially connected ad hoc networks," Technical Report CS-200006, Duke University, Tech. Rep., 2000.

[9] K. A. Harras, K. C. Almeroth, and E. M. Belding-Royer, "Delay tolerant mobile networks (DTMNs): Controlled flooding in sparse mobile networks," in *Proc. of NETWORKING 2005*. Springer, 2005, pp. 1180–1192.

[10] T. Spyropoulos, K. Psounis, and C. S.Raghavendra, "Spray and wait: an efficient routing scheme for intermittently connected mobile networks," in *Proc. of the ACM SIGCOMM workshop on Delay-tolerant networking*. ACM, 2005, pp. 252–259.

[11] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and focus: Efficient mobility-assisted routing for heterogeneous and correlated mobility," in *Proc. of the 5th Annual IEEE International Conference on Pervasive Computing and Communications Workshops*. IEEE, 2007, pp. 79–85.

[12] W. Moreira Junior and P. Mendes, "Survey on opportunistic routing for delay/disruption tolerant networks," SITI, University Lusófona, Tech. Rep. SITI-TR-11-02, Feb. 2011.

[13] M. Musolesi, S. Hailes, and C. Mascolo, "Adaptive routing for intermittently connected mobile ad hoc networks," in *Proc. of the sixth IEEE International Symposium on a World of wireless mobile and multimedia networks (WoWMoM)*. IEEE, 2005, pp. 183–189.

[14] L. B. Burns B, Brock O, "MV routing and capacity building in disruption tolerant network," in *Proc. of the 24th IEEE INFOCOM*. INFOCOM, 2005.

[15] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, "Maxprop: Routing for vehicle-based disruption-tolerant networks." in *INFOCOM*, vol. 6, 2006, pp. 1–11.

[16] S. C. Nelson, M. Bakht, and R. Kravets, "Encounter-based routing in DTNs," in *Proc. of the IEEE INFOCOM*. IEEE, 2009, pp. 846–854.

[17] M. Grossglauser and M. Vetterli, "Locating mobile nodes with ease: learning efficient routes from encounter histories alone," *IEEE/ACM Transactions on Networking (TON)*, vol. 14, no. 3, pp. 457–469, 2006.

[18] R. Ramanathan, R. Hansen, P. Basu, R. Rosales-Hain, and R. Krishnan, "Prioritized epidemic routing for opportunistic networks," in *Proc. of the 1st international MobiSys workshop on Mobile opportunistic networking*. ACM, 2007, pp. 62–66.

[19] A. Balasubramanian, B. Levine, and A. Venkataramani, "DTN routing as a resource allocation problem," in *ACM SIGCOMM Computer Communication Review*, vol. 37, no. 4. ACM, 2007, pp. 373–384.

[20] P. Hui and J. Crowcroft, "Bubble rap: Forwarding in small world DTNs in ever decreasing circles," *Univ. of Cambridge, Computer Laboratory, Tech. Rep. UCAMCL-TR*, vol. 684, 2007.

[21] E. Bulut and B. K. Szymanski, "Friendship based routing in delay tolerant mobile social networks," in *Proc. of the Global Telecommunications Conference (GLOBECOM)*. IEEE, 2010, pp. 1–5.

[22] P. Hui and J. Crowcroft, "How small labels create big improvements," in *Proc. of the Fifth Annual IEEE International Conference on Pervasive Computing and Communications Workshops*. IEEE, 2007, pp. 65–70.

[23] E. M. Daly and M. Haahr, "Social network analysis for routing in disconnected delay-tolerant manets," in *Proc. of the 8th ACM international symposium on Mobile ad hoc networking and computing*. ACM, 2007, pp. 32–40.

[24] A. Mei, G. Morabito, P. Santi, and J. Stefa, "Social-aware stateless forwarding in pocket switched networks," in *Proc. of the IEEE INFOCOM*. IEEE, 2011, pp. 251–255.

[25] W. Gao and G. Cao, "User-centric data dissemination in disruption tolerant networks," in *Proc. of the IEEE INFOCOM*. IEEE, 2011, pp. 3119–3127.

[26] A. Mei, G. Morabito, P. Santi, and J. Stefa, "Social-aware stateless routing in pocket switched networks," *PARALLEL AND DISTRIBUTED SYSTEMS, IEEE Transactions on*, 2014.

[27] M. Xiao, J. Wu, and L. Huang, "Community-aware opportunistic routing in mobile social networks," *COMPUTERS, IEEE Transactions on*, vol. 25, no. 5, pp. 1200–1210, 2014.

[28] C. Boldrini, M. Conti, J. Jacopini, and A. Passarella, "HiBOp: a history based routing protocol for opportunistic networks," in *Proc. of the eighth IEEE International Symposium on a World of wireless mobile and multimedia networks (WoWMoM)*. IEEE, 2007, pp. 1–12.

[29] J. Wu and Y. Wang, "Social feature-based multi-path routing in delay tolerant networks," in *Proc. of the IEEE INFOCOM*. IEEE, 2012, pp. 1368–1376.

[30] T. Ning, Z. Yang, and X. Xie, "Incentive-aware data dissemination in delay-tolerant mobile networks," in *Proc. of the 8th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*. IEEE, 2011, pp. 539–547.

[31] G. Wu, J. Wang, L. Yao, and C. Lin, "A secure social-aware incentive scheme for delay tolerant networks," in *Proc. of the 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*. IEEE, 2013, pp. 813–820.

[32] X. Xie, H. Chen, and H. Wu, "Bargain-based stimulation mechanism for selfish mobile nodes in participatory sensing network," in *Proc. of the 6th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*. IEEE, 2009, pp. 1–9.

[33] L. Buttyán, L. Dóra, M. Félegyházi, and I. Vajda, "Barter trade improves message delivery in opportunistic networks," *Ad Hoc Networks*, vol. 8, no. 1, pp. 1–14, 2010.

[34] J. Liu and V. Issarny, "Enhanced reputation mechanism for mobile ad hoc networks," in *Trust management*. Springer, 2004, pp. 48–62.

[35] M. Y. S. Uddin, B. Godfrey, and T. Abdelzaher, "RELICS: In-network realization of incentives to combat selfishness in DTNs," in *Proc. of the 18th IEEE International Conference on Network Protocols (ICNP)*. IEEE, 2010, pp. 203–212.

[36] G. Bigwood and T. Henderson, "IRONMAN: Using social networks to add incentives and reputation to opportunistic networks," in *Proc. of the 3rd International Conference on Privacy, security, risk and trust (socialcom)*. IEEE, 2011, pp. 65–72.

[37] Q. Li, W. Gao, S. Zhu, and G. Cao, "A routing protocol for socially selfish delay tolerant networks," *Ad Hoc Networks*, vol. 10, no. 8, pp. 1619–1632, 2012.

[38] J. Miao, O. Hasan, S. B. Mokhtar, L. Brunie, and K. Yim, "An investigation on the unwillingness of nodes to participate in mobile delay tolerant network routing," *International Journal of Information Management*, vol. 33, no. 2, pp. 252–262, 2013.

[39] H. Zhou, J. Chen, J. Fan, Y. Du, and S. K. Das, "ConSub: Incentive-based content subscribing in selfish opportunistic mobile networks," *IEEE Journal on Selected Areas in Communications*, no. 99, pp. 1–11, 2013.

[40] R.-I. Ciobanu, C. Dobre, M. Dascalu, S. Trausan-Matu, and V. Cristea, "Collaborative selfish node detection with an incentive mechanism for opportunistic networks," in *Proc. of the IFIP/IEEE International Symposium on Integrated Network Management (IM)*. IEEE, 2013, pp. 1161–1166.

[41] A. Shikfa, M. Önen, and R. Molva, "Privacy and confidentiality in context-based and epidemic forwarding," *Computer Communications*, vol. 33, no. 13, pp. 1493–1504, 2010.

[42] I. Parris and T. Henderson, "Privacy-enhanced social-network routing," *Computer Communications*, vol. 35, no. 1, pp. 62–74, 2012.

[43] R. Hardin, *Trust and trustworthiness*. Russell Sage Foundation, 2002.

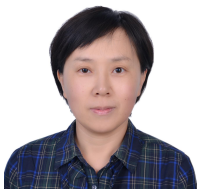
[44] P. Mundur and M. Seligman, "Delay tolerant network routing: Beyond epidemic routing," in *Proc. of the 3rd International Symposium on Wireless Pervasive Computing (ISWPC)*. IEEE, 2008, pp. 550–553.

[45] Y. Zhang and J. Zhao, "Social network analysis on data diffusion in delay tolerant networks," in *Proc. of the tenth ACM international symposium on Mobile ad hoc networking and computing*. ACM, 2009, pp. 345–346.

[46] T. Ning, Z. Yang, H. Wu, and Z. Han, "Self-interest-driven incentives for ad dissemination in autonomous mobile social networks," in *Proc. of the IEEE INFOCOM*. IEEE, 2013, pp. 2310–2318.

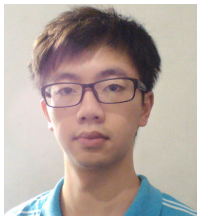
[47] A. Keränen, J. Ott, and T. Kärkkäinen, "The ONE simulator for DTN protocol evaluation," in *Proc. of the 2nd international conference on simulation tools and techniques*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2009, p. 55.

[48] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, and A. Chaintreau, "CRAWDAD data set cambridge/haggle (v. 2006-01-31)," Downloaded from http://crawdad.org/cambridge/haggle/, Jan. 2006.



Lin Yao received received Ph.D. degree from Dalian University of Technology, China in 2011. She is now an associate professor in School of Software, Dalian University of Technology (DUT), China. Her research interests include wireless sensor networks, opportunistic network.

Yanmao Man received the B.E. degree in Network Engineering from Dalian University of Technology, P. R. China, in Summber, 2015. He is currently a graduate research assistant in the Department of Computer Science (CS) at the Univeristy of North Carolina at Greensboro (UNCG). His research interests include opportunistic networks and social networks.



Zhong Huang received the B.E. degree in Network Engineering from Dalian University of Technology, P. R. China, in Summber, 2015.

Jing Deng received B.E. and Ph.D. degrees from Tsinghua University, China, in 1994 and 2002, respectively. Dr. Jing Deng is an associate professor in the Department of Computer Science (CS) at the University of North Carolina at Greensboro (UNCG). Dr. Deng is an editor of IEEE Transactions on Vehicular Technology (TVT). He is a senior member of the IEEE. His research interests include wireless network security, information assurance, mobile ad hoc networks, and wireless sensor networks.

Xin Wang is currently an associate professor of the department of Electrical and Computer Engineering.Her research interests include mobile and ubiquitous computing, wireless communications and network systems, networked sensing and fusion, networked autonomous systems, detection and estimation. She has served in executive committee and technical committee of numerous conferences and funding review panels, and is currently serving as the associate editor of IEEE Transactions on Mobile Computing.