REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
15-05-2011	Quarterly technical report	15 Feb 2011 - 14 May 2011

4. TITLE AND SUBTITLE
On Integrated Social and QoS Trust-Based Routing in Delay Tolerant Networks

5a. CONTRACT NUMBER

5b. GRANT NUMBER
N00014-10-1-0156

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Chen, Ing-Ray (VT)
Bao, Fenye (VT)
Chang, MoonJeong (VT)
Cho, Jin-Hee (ARL)

5d. PROJECT NUMBER
10PR02543-01

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
VIRGINIA POLYTECHNIC INSTITUTE AND STATE UNIVERSITY
OFFICE OF SPONSORED PROGRAMS
1880 PRATT DRIVE, SUITE 2006
BLACKSBURG, VA 24060-3325

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Office of Naval Research
875 North Randolph Street
Arlington, VA 22203-1995

10. SPONSOR/MONITOR'S ACRONYM(S)
ONR

11. SPONSORING/MONITORING AGENCY REPORT NUMBER

12. DISTRIBUTION AVAILABILITY STATEMENT
Distribution Statement A: Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT
We propose and analyze a class of integrated social and quality of service (QoS) trust-based routing protocols in mobile ad-hoc delay tolerant networks. The underlying idea is to incorporate trust evaluation in the routing protocol, considering not only QoS trust properties but also social trust properties to evaluate other nodes encountered. We prove that our protocol is resilient against bad-mouthing, good-mouthing and whitewashing attacks performed by malicious nodes. By utilizing a stochastic Petri net model describing a delay tolerant network consisting of heterogeneous mobile nodes with vastly different social and networking behaviors, we analyze the performance characteristics of trust-based routing protocols in terms of message delivery ratio, message delay, and message overhead against connectivity-based, epidemic and PROPHET routing protocols. The results indicate that our trust-based routing protocols outperform PROPHET and can approach the ideal performance obtainable by epidemic routing in delivery ratio and message delay, without incurring high message overhead. Further, integrated social and QoS trust-based protocols can effectively trade off message delay for a significant gain in message delivery ratio and message overhead over traditional connectivity-based routing protocols.

15. SUBJECT TERMS
Delay tolerant networks, opportunistic routing, trust management, trust-based routing, social networks, resiliency, performance analysis, stochastic Petri nets.

INSTRUCTIONS FOR COMPLETING SF 298

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
			SAR	16	Chen, Ing-Ray
a. REPORT U	b. ABSTRACT U	c. THIS PAGE U			19b. TELEPONE NUMBER (*Include area code*) (703) 538-8376

On Integrated Social and QoS Trust-Based Routing in Delay Tolerant Networks

Ing-Ray Chen, Fenye Bao, MoonJeong Chang
Department of Computer Science
Virginia Tech
{irchen, baofenye, mjchang}@vt.edu

Jin-Hee Cho
Computational and Information Sciences Directorate
U.S. Army Research Laboratory
jinhee.cho@us.army.mil

Abstract: We propose and analyze a class of integrated social and quality of service (QoS) trust-based routing protocols in mobile ad-hoc delay tolerant networks. The underlying idea is to incorporate trust evaluation in the routing protocol, considering not only *QoS trust* properties but also *social trust* properties to evaluate other nodes encountered. We prove that our protocol is resilient against bad-mouthing, good-mouthing and whitewashing attacks performed by malicious nodes. By utilizing a stochastic Petri net model describing a delay tolerant network consisting of heterogeneous mobile nodes with vastly different social and networking behaviors, we analyze the performance characteristics of trust-based routing protocols in terms of message delivery ratio, message delay, and message overhead against connectivity-based, epidemic and PROPHET routing protocols. The results indicate that our trust-based routing protocols outperform PROPHET and can approach the ideal performance obtainable by epidemic routing in delivery ratio and message delay, without incurring high message overhead. Further, integrated social and QoS trust-based protocols can effectively trade off message delay for a significant gain in message delivery ratio and message overhead over traditional connectivity-based routing protocols.

Keywords: Delay tolerant networks, opportunistic routing, trust management, trust-based routing, social networks, resiliency, performance analysis, stochastic Petri nets.

1. Introduction

A delay tolerant network (DTN) provides interoperable communications through mobile nodes with the characteristics of high end-to-end path latency, frequent disconnection, limited

resources (e.g., battery, computational power, bandwidth), and unreliable wireless transmission. Further, for DTNs in mobile ad hoc network (MANET) environments, we also face additional challenges due to a lack of centralized trusted entity and this increases security vulnerability [5]. For a sparse MANET DTN, mobility-assisted routing based on *store-carry-and-forward* method has been used. That is, a message carrier forwards a message to an encountered node until the message reaches a destination node. In MANET DTN environments, it is important to select a trustworthy node as a next message carrier among all encountered nodes to minimize the delay for a message to reach a destination node as well as to maximize the message delivery ratio. In this paper, we consider a MANET DTN in the presence of selfish and malicious nodes and propose a family of trust-based routing protocols to select a highly trustworthy next message carrier with the goal of maximizing the message delivery ratio without incurring a high delay or a high message overhead.

In the literature, DTN routing protocols based on encounter patterns have been investigated [2][10][11]. However, if the predicted encounter does not happen, then messages would be lost for single-copy routing, or flooded for multi-copy routing. Moreover, these approaches could not guarantee reliable message delivery due to the presence of selfish or malicious nodes. The vulnerability of DTN routing to node selfishness was well studied in [7][17]. Several recent studies [12][14][15] considered using reputation in selecting message carriers among encountered nodes for DTNs. Nevertheless, [12][15] assumed that a centralized entity exists for credit management, and [14] merely used reputation to judge if the system should switch from reputation-based routing to multipath routing when many selfish nodes exist.

There is very little research to date on the social aspect of trust management for DTN routing. Social relationship and social networking were considered as criteria to select message carriers in a MANET DTN [6][8]. However, no consideration was given to the presence of malicious or selfish nodes. Very recently, [9] considered routing by socially selfish nodes in DTNs, taking into consideration the willingness of a socially selfish node to forward messages to the destination node because of social ties. Unlike prior work cited above, in this paper, we integrate *social trust* and *Quality of Service (QoS) trust* into a composite trust metric for determining the best node among the new encounters for message forwarding, extending from our preliminary work [16]. We consider *honesty* and *unselfishness* for social trust to account for a node's trustworthiness for message delivery, and *connectivity* for QoS trust to account for a node's

capability to quickly deliver the message to the destination node. By assigning various weights associated with these QoS and social trust properties, we form a class of DTN routing protocols, from which we examine two versions of the trust management protocol in this paper: an equal-weight QoS and social trust management protocol (called trust-based routing for short) and a QoS trust only management protocol (called connectivity-based routing for short). We analyze and compare the performance characteristics of trust-based routing and connectivity-based routing protocols with epidemic routing [13] and PROPHET [18] for a DTN consisting of heterogeneous mobile nodes with vastly different social and networking behaviors. The results indicate that our trust-based routing protocols outperform PROPHET and can approach the ideal performance obtainable by epidemic routing in delivery ratio and message delay, without incurring high message overhead. Further, integrated social and QoS trust-based protocols can effectively trade off message delay for a significant gain in message delivery ratio and message overhead over connectivity-based routing protocols.

2. System Model

We consider a MANET DTN environment with no centralized trusted authority. Nodes communicate through multi-hops. Every node may have a different level of energy and speed reflecting node heterogeneity. We differentiate selfish nodes from malicious nodes. A selfish node acts for its own interest. So it may drop packets arbitrarily just to save energy but it may decide to forward a packet if it has good social ties with the destination node. A malicious node acts maliciously with the intention to disrupt the main functionality of the DTN, so it can drop packets, jam the wireless channel, perform bad-mouthing attacks (provide negative recommendations against good nodes), perform good-mouthing attacks (provide positive recommendations for other colluding malicious nodes) and even forge packets. In the paper, we will use the terms a malicious node, a compromised node, and a bad node interchangeably.

We consider the following model to describe a node's behaviors. If a node is selfish, the speed of energy consumption is slowed down and vice versa. If a node is compromised, the speed of energy consumption will increase since the node may have a chance to perform attacks which may consume more energy, e.g., disseminating bogus messages. We also consider redemption mechanism for a selfish node to have a second chance. That is, a selfish node may become unselfish again, especially when its energy is still high compared with its peers. We

assume that each node has a pair of pre-distributed public/private keys which can be used for packet authentication and preventing spoofing attacks.

A node's trust value is assessed based on direct observations and indirect information like recommendations. The trust of one node toward another node is updated upon encounter events. Our trust metric consists of two trust types: *QoS trust* and *social trust*. *QoS trust* is evaluated through the communication by the capability of a node to deliver messages to the destination node. We consider ***connectivity*** to measure the QoS trust level of a node. Social trust is based on social relationships. We consider ***unselfishness*** and ***honesty*** to measure the social trust level of a node. Different from most existing encounter-based routing protocols which considered only connectivity, we consider social trust in addition to QoS trust in order to select more trustworthy message carriers among encountered nodes. It is worth noting that unselfishness traditionally has been considered as a QoS trust metric [3] to measure the extent to which a node cooperates with other nodes to conform to protocol execution. Here we consider unselfishness as a social trust metric to measure if a node is socially willing to route packets passed to it in a DTN, thereby modeling the social behavior exhibited by a selfish node. We define a node's trust level as a real number in the range of [0, 1], with 1 indicating complete trust, 0.5 ignorance, and 0 complete distrust.

There is no centralized intrusion detection system (IDS) as it may be infeasible to implement an efficient IDS in a DTN environment because of the sparseness of nodes and small chances for certain nodes to encounter or connect to each other. Each node will execute the trust protocol independently and will perform its *direct trust* assessment toward an encountered node based on specific detection mechanisms designed for detecting a trust property X, with X=connectivity, unselfishness, or honesty. In the paper we will discuss these specific detection mechanisms employed in our protocol.

3. Trust Management for Message Routing

The trust value of node j as evaluated by node i at time t, denoted as $T_{i,j}(t)$, is computed by a weighted average of connectivity, honesty, and unselfishness trust components. Specifically node i will compute $T_{i,j}(t)$ by:

$$T_{i,j}(t) = w_1 T_{i,j}^{e-connectivity}(t) + w_2 T_{i,j}^{d-connectivity}(t) + w_3 T_{i,j}^{honesty}(t) + w_4 T_{i,j}^{unselfis\ hness}(t) \quad (1)$$

where $w_1 : w_2 : w_3 : w_4$ is the weight ratio with $w_1 + w_2 + w_3 + w_4 = 1$. Of these trust components (or properties) in Equation 1, $T_{i,j}^{e-connectivity}(t)$ is about node i's belief in node i's encounter connectivity to node j, representing the delay of node i passing the message to node j, $T_{i,j}^{d-connectivity}(t)$ is about node i's belief in node j's connectivity to the destination node d, representing the delay of node j passing the message to node d, $T_{i,j}^{honesty}(t)$ is about node i's belief in node j's honesty, and $T_{i,j}^{unselfis\ hness}(t)$ is about node i's belief in node j's unselfishness.

The reason of considering both e-connectivity and d-connectivity trust properties in our protocol is given as follows. The end-to-end delay from node i's perspective consists of the e-connectivity delay from node i to node j (the next carrier) and the d-connectivity delay from node j to node d (the destination node). Thus, both connectivity metrics are needed. Suppose d-connectivity is only trust metric for connectivity. If node i encounters node j and discovers that node j's d-connectivity delay is higher than another node's (say node m's) d-connectivity delay, then node i will decide not to pass the message to node j. This would be a wrong decision in case node m's e-connectivity delay + d-connectivity delay actually is higher than node j's e-connectivity delay (which is zero upon encounter) + d-connectivity delay. Here we note two special cases: (1) if node j is the currently encountered node, then $T_{i,j}^{e-connectivity}(t)$ is one, representing that the e-connectivity delay is zero; (2) if node d is the currently encountered node, then both $T_{i,j}^{e-connectivity}(t)$ and $T_{i,j}^{d-connecti\ vity}(t)$ are one, representing that both the e-connectivity delay and d-connectivity delay are zero.

In message forwarding in DTNs, two most important performance metrics are message delivery ratio and delay. The rationale of using these four trust metrics is to rank nodes such that high $T_{i,j}^{e-connectivity}(t)$ and $T_{i,j}^{d-connectivity}(t)$ represent low end-to-end delay, while high $T_{i,j}^{honesty}(t)$ and $T_{i,j}^{unselfis\ hness}(t)$ represent high delivery ratio. We set $T_{i,j}^{e-connectivity}(0)$, $T_{i,j}^{d-connectivity}(0)$, $T_{i,j}^{honesty}(0)$ and $T_{i,j}^{unselfis\ hness}(0)$ to ignorance (0.5) since initially there is no information exchanged among nodes.

We define a minimum trust threshold T_{min} also set to ignorance (0.5) such that if $T_{i,j}(t) > T_{min}$, node i will consider node j as "trustworthy" (or plainly as a good node) at time t. When node i encounters another node, say node m, it exchanges its encounter history with node m.

Moreover, if node i believes that node m is a good node, i.e., $T_{i,m}(t+\Delta t) > T_{min}$, where Δt is the encounter period, node i will use node m as a recommender to update its beliefs toward other nodes. Specifically, node i will update its trust toward node j upon encountering node m at time t for a duration of Δt as follows:

$$T_{i,j}^{X}(t+\Delta t) = \beta_1 \, T_{i,j}^{direct\,,\,X}(t+\Delta t) + \beta_2 \, T_{i,j}^{indirect\,,\,X}(t+\Delta t) \tag{2}$$

Here X refers to a trust property (e-connectivity, d-connectivity, honesty, or unselfishness) with:

$$T_{i,j}^{direct\,,\,X}(t+\Delta t) = \begin{cases} T_{i,m}^{encounter\,,X}(t+\Delta t), if\ m = j \\ T_{i,j}^{X}(t), if\ m \neq j \end{cases} \tag{3}$$

$$T_{i,j}^{indirect\,,\,X}(t+\Delta t) = \begin{cases} T_{i,m}^{X}(t), if\ m = j \\ T_{i,j}^{X}(t), if\ m \neq j\ and\ T_{i,m}(t+\Delta t) \leq T_{min} \\ T_{i,m}^{X}(t+\Delta t) \times T_{m,j}^{X}(t+\Delta t), if\ m \neq j\ and\ T_{i,m}(t+\Delta t) > T_{min} \end{cases} \tag{4}$$

In Equation 2, β_1 is a weight parameter to weigh node i's own trust assessment toward node j at time $t+\Delta t$, i.e., "self-information," and β_2 is a weight parameter to weigh indirect information from the recommender, i.e., "other-information," with $\beta_1 + \beta_2 = 1$.

In Equation 3 for the direct trust calculation of node j, if the new encounter (node m) is node j itself, then node i can directly evaluate node j. We use $T_{i,m}^{encounter\,,X}(t+\Delta t)$ to denote the assessment result of node i toward node m in trust property X based on node i's past experiences with node m up to time $t+\Delta t$. This means that the value of $T_{i,m}^{encounter\,,X}(t+\Delta t)$ is assessed based on node i's direct observations to node m collected while they encountered with each other (including the current encounter) over the time period $[0, t+\Delta t]$. Later in Section 5, we will describe how this can be obtained for each trust property X. If the new encounter is not node j, then there is no new direct information can be gained about node j, so node i will just use its past trust toward node j obtained at time t.

In Equation 4 for the indirect trust calculation of node j, if the new encounter is node j itself, then there is no indirect recommendation for node j, so node i will just use its past trust obtained at time t. If the new encounter is not node j, then node m can provide its recommendation to node i for evaluating node j, if node i considers node m as trustworthy, i.e., $T_{i,m}(t+\Delta t) > T_{min}$. In this case, we must take into account node i's belief in node m in the calculation of

$T_{i,j}^{indirect,\ X}(t+\Delta t)$. This models the decay of trust as trust is derived from a distant node as indirect information. On the other hand, if node i does not consider node m as a good node because of $T_{i,m}(t+\Delta t) \leq T_{min}$, then node i refuses to take recommendations from node m about node j, and will just use its past trust information about node j obtained at time t. The policy that recommendations from a newly encounter node are accepted only if the newly encountered node is considered a good node provides robustness against bad-mouthing or good-mouthing attacks.

$T_{i,j}(t)$ in Equation 1 can be used by node i (if it is a message carrier) to decide, upon encountering node m, if it should forward the message to node m with the intent to shorten the message delay or improve the message delivery ratio. We consider a Ω–permissible policy in this paper, i.e., node i will pass the message to node m if $T_{i,m}(t)$ is in the top Ω percentile among all $T_{i,j}(t)'s$. We experiment with various values of Ω to trade message delivery ratio with message latency.

4. Protocol Resiliency

Below we provide a formal proof that our trust management protocol is resilient against malicious attacks, including whitewashing attacks (a bad node washing away its bad reputation by gaining high trust upon encountering with another node), bad-mouthing attacks (a bad node providing bad recommendations toward a good node to ruin its reputation), and good-mouthing attacks (a bad node providing good recommendations for a colluding bad node to raise its reputation).

4.1 Resiliency to Whitewashing Attacks

Definition 1: A bad node, say node m, upon encountering node i at time t for an encounter interval Δt, is said to perform a whitewashing attack successfully against node i if $T_{i,m}(t) \leq T_{min}$ and $T_{i,m}(t+\Delta t) > T_{min}$.

Lemma 1: Our protocol is resilient against whitewashing attacks.

Proof: When node i encounters node m at time t for a duration of Δt, according to our protocol $T_{i,m}(t+\Delta t) = \beta_1 T_{i,m}^{encounter}(t+\Delta t) + \beta_2 T_{i,m}(t)$, of which $T_{i,m}(t) \leq T_{min}$ is given (in the *if* part) and $T_{i,m}^{encounter}(t+\Delta t) \leq T_{min}$ is true because node i will be able to observe node m's bad behavior directly based on node i's past experiences with node m up to time $t+\Delta t$, including the

current encounter. Taking the fact that $\beta_1 + \beta_2 = 1$, we obtain $T_{i,m}(t + \Delta t) \leq T_{min}$. Thus, it is impossible that a bad node can successfully perform a whitewashing attack.

4.2 Resiliency to Bad-Mouthing Attacks

Definition 2: A bad node, say node m, upon encountering node i at time t for an encounter interval Δt, is said to perform a bad-mouthing attack successfully against a good node, say node j, if $T_{i,j}(t) > T_{min}$ and $T_{i,j}(t + \Delta t) \leq T_{min}$.

Lemma 2: Our protocol is resilient against bad-mouthing attacks.

Proof: The proof hinges on proving $T_{i,m}(t + \Delta t) \leq T_{min}$ and therefore node i will refuse to take recommendations from node m about node j. Utilizing the proof to Lemma 1 and the fact that $T_{i,m}(t) \leq T_{min}$ is true (because we set the initial trust value to ignorance, i.e., $T_{i,m}(0) = T_{min}$, making it impossible for a bad node to gain trustworthy status at time t), we know $T_{i,m}(t + \Delta t) \leq T_{min}$ is true. Consequently, node i will not take recommendations from node m about node j. According to our protocol, $T_{i,j}(t + \Delta t) = \beta_1 T_{i,j}(t) + \beta_2 T_{i,j}(t)$. This leads to $T_{i,j}(t + \Delta t) > T_{min}$ because $\beta_1 + \beta_2 = 1$ and $T_{i,j}(t) > T_{min}$ is given (in the *if* part). Therefore, it is impossible that a bad node can successfully perform a bad-mouthing attack.

4.3 Resiliency to Good-Mouthing Attacks

Definition 3: A bad node, say node m, upon encountering node i at time t for an encounter interval Δt, is said to perform a good-mouthing attack successfully for a bad node, say node k, if $T_{i,k}(t) \leq T_{min}$ and $T_{i,k}(t + \Delta t) > T_{min}$.

Lemma 3: Our protocol is resilient against good-mouthing attacks.

Proof: Following the proof to Lemma 1, we know that $T_{i,m}(t + \Delta t) \leq T_{min}$ is true. Hence, node i refuses to take recommendations from node m about node k and $T_{i,k}(t + \Delta t) = \beta_1 T_{i,k}(t) + \beta_2 T_{i,k}(t)$ according to our protocol. This leads to $T_{i,k}(t + \Delta t) \leq T_{min}$ because $\beta_1 + \beta_2 = 1$ and $T_{i,k}(t) \leq T_{min}$ is given (in the *if* part). Therefore, it is impossible that a bad node can successfully perform a good-mouthing attack.

5. Performance Model

We analyze the performance of the proposed trust-based routing protocol for DTN message forwarding by a probability model based on stoehastie Petri net (SPN) teehniques [4] due to its ability to handle a large number of states.

5.1 SPN Model to Yield Ground Truth

We develop an SPN model to yield dynamie ground truth information of nodes in the example DTN described in Section 2. The SPN model is shown in Figure 1. The SPN model describes a node's lifetime in the presence of selfish and malicious nodes. It is used to obtain eaeh node's information (e.g., conneetivity, honesty, and unselfishness) and to derive the trust relationship with other nodes in the system.

Without loss of generality, we eonsider a square-shaped operational area eonsisting of $m \times m$ sub-grid areas with the width and height equal to the radio range (R). Initially nodes are randomly distributed over the operational area based on uniform distribution. A node randomly moves to one of four loeations in four direetions (i.e., north, west, south, and east) in accordanee with its mobility rate. To avoid end-effeets, movement is wrapped around (i.e., a torus is assumed). The SPN model produces the probability that a node, say node i, is in a particular location L at time t. This information along with the location information of other nodes at time t (derived from these nodes' SPN models) provides us the probability of two nodes encountering with eaeh other, and how often two nodes exehange encounter histories to update $T_{i,j}^{X}(t)$.

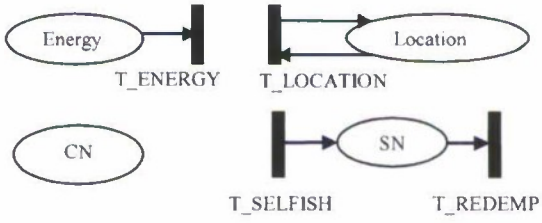


Figure 1: SPN Model.

Below we explain how we eonstruet the SPN model for deseribing a node's behavior in terms of its location, energy, honesty, and selfishness status.

Location: Transition T_LOCATION is triggered when the node moves to a randomly selected area out of four different directions from its current location with the rate being calculated as σ/R based on its speed σ and wireless radio range R.

Connectivity: Connectivity of node i to node j is measured by the time-averaged probability that node i and node j are within one-hop during $[0, t + \Delta t]$. This can be obtained by knowledge of location probabilities of node i and node j during $[0, t + \Delta t]$.

Energy: Place *Energy* represents the current energy level of a node. An initial energy level of each node is assigned according to node heterogeneity information. A token is taken out when transition T_ENERGY fires. The transition rate of T_ENERGY is adjusted on the fly based on a node's state. It is lower when a node is selfish to save energy; it is higher when the node is compromised so that it performs attacks more and consumes energy more. We use the energy model in [3] to adjust the rate to consume one token in place *Energy* based on a node's state.

Honesty: A node is either good or bad. We distinguish a bad (or compromised) node from a good node by placing a token in place *CN*. A bad node can perform various attacks including white washing, good mouthing and bad mouthing attacks, thus exhibiting dishonest behaviors. When a node encounters another node, it will perform a direct trust assessment of the encountered node in the *honest* trust property based on specific detection mechanisms devised for detecting dishonesty (to be described later).

Selfishness: Place *SN* represents whether a node is selfish or not. If a node becomes selfish, a token goes to *SN* by triggering T_SELFISH. We model a node's selfish behavior as a function of its remaining energy. Specifically, the transition rate to T_SELFISH is given by:

$$rate(T\_SELFISH) = \frac{f(E_{remain})}{\Delta t} \tag{5}$$

where Δt is the duration between two encountering events over which a node may decide to become selfish. The form $f(y) = \alpha_1 y^{-\varepsilon_1}$ follows the demand-pricing relationship in Economics [1] to model the effect of its argument y on the selfishness behavior, such that $f(E_{remain})$ models the behavior that a node with a higher level of energy is less likely to be selfish. Similarly a selfish node may become unselfish again through transition T_REDEMP. The redemption rate is modeled in a similar way as:

$$rate(T\_REDEMP) = \frac{g(E_{consumed})}{\Delta t} \quad (6)$$

where $g(y) = \alpha_2 y^{-\varepsilon_2}$ and $E_{consumed}$ is the amount of energy consumed as given by $E_0 - E_{remain}$ and Δt is the encountering interval over which a selfish node may decide to become unselfish again. $g(E_{consumed})$ models the behavior that a node with a lower level of energy will more likely stay selfish to further save its energy considering its own individual benefit.

5.2 Trust Assessment

Leveraging the SPN model described which yields ground truth information of node *i*'s status, we can calculate $T_{i,j}^{X}(t + \Delta t)$ as follows. In practice, $T_{i,j}^{X}(t + \Delta t)$ is obtained by node *i* by following the protocol execution at runtime. The computational procedure devised below is to predict $T_{i,j}^{X}(t + \Delta t)$ that would be obtained by node *i*. Our assertion is that the detection mechanisms used by a node for trust assessment of property *X* of an encountered node will be effective and fairly accurate. Thus, $T_{i,m}^{encounter\ ,X}(t + \Delta t)$ assessed by node *i* will be close to ground truth. Consequently, $T_{i,m}^{encounter\ ,X}(t + \Delta t)$ is predicted to be the same as the ground truth status of node *m* in trust property *X*, as provided from the SPN output. Below we discuss specific detection mechanisms used by node *i* to assess node *m* upon encounter to satisfy the assertion.

- $T_{i,m}^{encounter\ ,\ e-connectivity}(t + \Delta t)$: This refers to the belief of node *i* about its connectivity to node *m* based on node *i*'s encountering experiences. The specific detection mechanism used is counter-based. That is, node *i* keeps track of the numbers of encounters it has had with all other nodes in the DTN up to time $t + \Delta t$ and computes $T_{i,m}^{encounter\ ,\ e-connectivity}(t + \Delta t)$ by the ratio of the number of encounters between node *i* and node *m* to the maximum number of encounters between node *i* and any other node during $[0, t + \Delta t]$.
- $T_{i,m}^{encounter\ ,\ d-connectivity}(t + \Delta t)$: This refers to the belief of node *i* about the connectivity between node *m* and node *d* based on node *i*'s encountering experiences. The specific detection mechanism used is also counter-based. It can be computed by the ratio of the number of encounters between node *m* and node *d* to the maximum number of encounters between node *d* and any other node over the time period $[0, t + \Delta t]$ all based on node *i*'s observations. Note that node *i* can observe node *m* encountering node *d* only if both node *m* and node *d* are within 1-hop range of node *i*. Thus, by consulting its encounter history with

all nodes, node i will be able to calculate $T_{i,m}^{encounter\ ,d-connectivity}(t+\Delta t)$ for the connectivity of node m to node d.

- $T_{i,m}^{encounter\ ,\ honesty}(t+\Delta t)$: This refers to the belief of node i that node m is honest based on direct observation experiences with node m during encounters. Since a compromised node will perform attacks and exhibit dishonest behaviors, the specific mechanisms used are anomaly detection or intrusion detection techniques [18][19]. Specifically, node i monitors node m's dishonest evidences including dishonest trust recommendation, irregular packet patterns, and abnormal traffic while they encountered including the current encounter. Then it computes $T_{i,m}^{encounter\ ,honesty}(t+\Delta t)$ by the ratio of the number of bad honesty experiences to the total number of honesty experiences.
- $T_{i,m}^{encounter\ ,\ unselfis\ hness}(t+\Delta t)$: This refers to the belief of node i that node m is willing to deliver messages. In traditional MANETs, a node's selfishness can be detected by using snooping and overhearing techniques. However, in DTNs messages are delivered in a store-and-forward fashion, thus snooping and overhearing may not be feasible. Our specific detection mechanism for detecting unselfishness is signature-based, leveraging the private/public keys for message authentication. Specifically, when node i encounters node j and passes a message to node j, if node j is not selfish it will forward the message and acknowledge node i with the same message signed with its private key. Afterwards, when node i and node m encounter each other, they exchange message signatures and verify each exchanged message signature by the receiver's public key. Since each message is unique, a bad node cannot apply replication attacks. An unselfish node therefore will have more verified message signatures than a selfish node. Node i then computes $T_{i,m}^{encounter\ ,\ unselfis\ hness}(t+\Delta t)$ by the ratio of the number of verified message signatures received from node m to the maximum number of verified message signatures received from any other node.

As a result of applying the above detection mechanisms for trust property X, $T_{i,m}^{encounter\ ,\ X}(t+\Delta t)$ obtained by node i would be close to the ground truth status of node m at time t which can be easily obtained from the SPN model output. In particular, $T_{i,m}^{encounter\ ,honesty}(t+\Delta t)$ in Equation 3 is simply equal to the probability that place CN does

not contain a token at time $t + \Delta t$, and $T_{i,m}^{encounter\ ,\ unselfis\ hness}(t + \Delta t)$ is simply equal to the probability that place *SN* does not contain a token at time $t + \Delta t$, both of which can be computed easily from the SPN model output. Similarly, $T_{i,m}^{encounter\ ,\ e-connectivity}(t + \Delta t)$ is simply equal to the time-averaged probability that node i and node m are within one-hop during $[0, t + \Delta t]$ and $T_{i,m}^{encounter\ ,\ d-connectivity}(t + \Delta t)$ is equal to the time-averaged probability that node m and node d are within one-hop during $[0, t + \Delta t]$, both of which can be obtained by utilizing the SPN model output regarding the node location probability. Once $T_{i,m}^{encounter\ ,X}(t + \Delta t)$ is obtained at each encounter time, node i computes $T_{i,j}^{X}(t + \Delta t)$ based on Equation 2, and subsequently, obtains $T_{i,j}(t + \Delta t)$ based on Equation 1.

6. Results

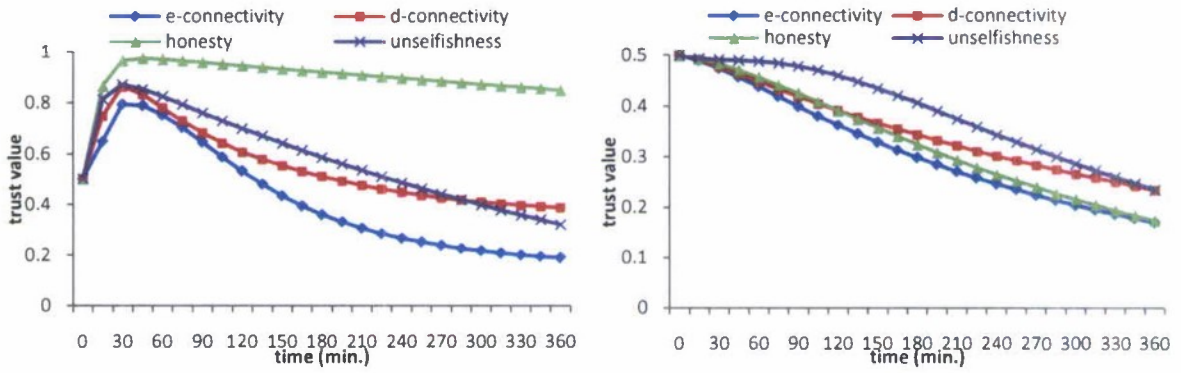
Table 1: Default Parameter Values Used.

Param	Value	Param	Value	Param	Value	Param	Value
$m \times m$	8×8	R	$250m$	T_{min}	0.5	σ	[0, 2] *m/s*
α_1	4	α_2	0.5	$\varepsilon_1, \varepsilon_2$	1.6	β_1:β_2	0.8:0.2
E_0	[12, 24] *hrs*	Ω	90%	Δt	300 *s*	N	20

Below we show numerical results and provide physical interpretation of the results obtained. Table 1 lists the default parameter values used. For trust-based routing, we set $w_1 : w_2 : w_3 : w_4 = 0.25 : 0.25 : 0.25 : 0.25$ for e-connectivity: d-connectivity: honesty: unselfishness, while for connectivity-based routing, we set $w_1 : w_2 : w_3 : w_4 = 0.5 : 0.5 : 0 : 0$. We setup $N = 20$ nodes with vastly different initial energy levels in the system moving randomly in a 8×8 operational region with the speed of each node in the range of [0, 2] *m/s*, and with each area covering 250 *m* radio radius. There are two sets of nodes, namely, good nodes and bad nodes, and we vary the percentage of bad nodes to test their effect on the performance of our protocol. A good node may become selfish to save energy and unselfish again after redemption, with the selfish rate defined based on Equation 5 and redemption rate defined by Equation 6. The initial trust level is set to ignorance (i.e., 0.5) for all trust components since initially nodes do not know each other. We also set T_{min} to 0.5 so that a node will take recommendations from a newly encountered node only when its trust level toward the newly encountered node exceeds ignorance.

To reveal which trust component might have a more dominant effect, we show $T_{i,j}^{e-connectivity}(t)$, $T_{i,j}^{d-connectivity}(t)$, $T_{i,j}^{honesty}(t)$ and $T_{i,j}^{unselfis\ hness}(t)$ for node i (a good

node) evaluating node j randomly picked. Other nodes exhibit similar trends and thus only one set of results is shown. Figure 2(a) is for the case in which node j is a good node. We see that all trust components exhibit the same trend. A good node initially picks up its trustworthy status (with its trust level greater than T_{min}) due to favorable direct evaluations by those nodes it encounters and interacts with, who in turn pass on their positive recommendations to other nodes they encounter. All trust components after their respectively maximum values then decline as time progresses because malicious negative recommendations from bad nodes performing bad-mouthing attacks gradually pick up advantages against positive recommendations from good nodes. Among all trust components, the honesty trust component is expected to contribute the most to the trustworthy status of a good node. This is reflected in Figure 2(a) which shows honesty dominates other trust components.



(a) Node j is a good node. (b) Node j is a bad node.

Figure 2: Comparing $T_{i,j}^{X}(t)$ as a Function of Time.

Figure 2(b) shows $T_{i,j}^{e-connectivity}(t)$, $T_{i,j}^{d-connectivity}(t)$, $T_{i,j}^{honesty}(t)$ and $T_{i,j}^{unselfis\ hness}(t)$ as a function of time for the case in which node j is a bad node. Here again all trust components exhibit the same trend. However, the trust values decrease monotonically over time. Contrary to a good node, a bad node never has any chance to attain trustworthy status, with the rapid decline of honesty and unselfishness especially contributing to a bad node's trust decline. The result that a bad node's status is always untrustworthy as demonstrated in Figure 2(b) substantiates our claim that our protocol is resilient against whitewashing, bad-mouthing, and good-mouthing attacks by malicious nodes.

Next we consider a message forwarding scenario in which in each run we randomly pick a source node s and a destination node d. The source and destination nodes picked are always good nodes. There is only a single copy of the message initially given to node s. We let the system run for 30 min. to warm up the system and start the message forwarding afterward in each run. During a message-passing run, every node i updates its $T_{i,j}(t)$ for all j's based on Equation 1. In particular, the current message carrier uses $T_{i,j}(t)$ to judge if it should pass the message to a node it encounters at time t. If the message carrier is malicious, the message is dropped (a weak attack). If the message carrier is selfish, the message delivery continues with 50% of the chance. A message delivery run is completed when the message is delivered to the destination node, or the message is lost before it reaches the destination node. Data are collected for 1500 runs from which the message delivery ratio, delay and overhead performance measurements are calculated.

We compare trust-based routing and connectivity-based routing against two baseline routing protocols, namely, epidemic routing [13] and PROPHET [18], in terms of message delivery ratio, delay and overhead performance metrics. Assuming sufficient buffer space, epidemic routing achieves the best performance in delivery ratio and message delay at the expense of the worst performance in delivery overhead (in the number of message copies generated). Thus, epidemic routing provides the upper bound performance in delivery ratio and message delay, and the lower bound performance in delivery overhead against which trust-based routing can be compared. We use PROPHET as another baseline routing protocol to demonstrate the effectiveness of trust-based routing protocols in all three performance metrics.

In epidemic routing [13], a node forwards a copy of the message to any node it encounters. Thus, a node consumes more energy because it propagates many redundant message copies to the network. Compared with trust-based routing and connectivity-based routing, however, epidemic routing saves energy because it does not have the overhead of trust management. When using the SPN model to describe a node executing epidemic routing, we adjust the energy consumption rate to transition T_ENERGY in the "Energy" subnet of the SPN model to account for a different energy consumption rate.

In PROPHET [18], when two nodes encounter, they exchange a *delivery predictability* vector. If the *delivery predictability* of the current message carrier is lower than that of the newly encountered node, then the message is passed from the current message carrier to the newly encountered node as the next carrier. The *delivery predictability* is a probabilistic metric

indicating the encounter frequency between two nodes and is updated when two nodes encounter each other. It is similar to the *d-connectivity* trust property used in our protocols in predicting the delay of the next carrier encountering the destination node. For ease of disposition, we will loosely refer *delivery predictability* as *d-connectivity*. PROPHET is a multi-copy routing protocol by which each node still keeps its message copy for future transmission after it sends the message to a carrier. For fair comparison, we consider a version of PROPEHT, called PROPHET_S, where each node removes its message copy after forwarding it to another node and will not perform any more message forwarding. The energy consumption model of each node in PROPHET_S is similar to trust-based routing protocols considering only *d-connectivity*. Therefore, in PROPHET_S the energy consumption rate to transition T_ENERGY in the SPN model remains the same as in our trust-based routing protocols.

Figure 3 shows the message delivery ratio as a function of the percentage of compromised and selfish nodes in the DTN for trust-based and connectivity-based routing protocols. For performance comparison, we also show the delivery ratio obtained from epidemic routing and PROPHET_S. Here we see that trust-based routing outperforms connectivity-based routing in delivery ratio and its performance approaches the maximum achievable performance obtainable from epidemic routing. This is attributed to the ability of trust-based protocols being able to differentiate trustworthy nodes from selfish and bad nodes and select trustworthy nodes to relay the message. The result demonstrates the effectiveness of incorporating social trust into the decision making process for DTN message routing. Among all protocols, PROPHET_S performs the worst in message delivery ratio. In particular, PROPHET_S is significantly worse than trust-based routing because it does not consider honesty and unselfishness for routing decisions. PROPHET_S is also considerably worse than connectivity-based routing because it considers only d-connectivity instead of both e-connectivity and d-connectivity, and the message is passed to a newly encountered node as long as the new encounter's d-connectivity is better than that of the current message carrier. This results in a longer route from the source node to the destination node with a higher chance to run into a malicious node or selfish node to drop the message.

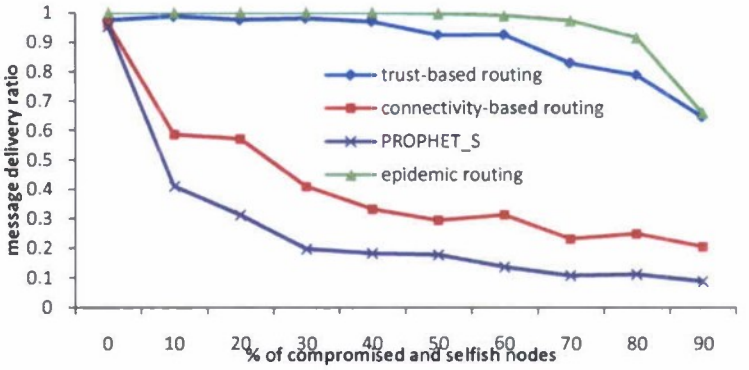


Figure 3: Performance Comparison in Message Delivery Ratio.

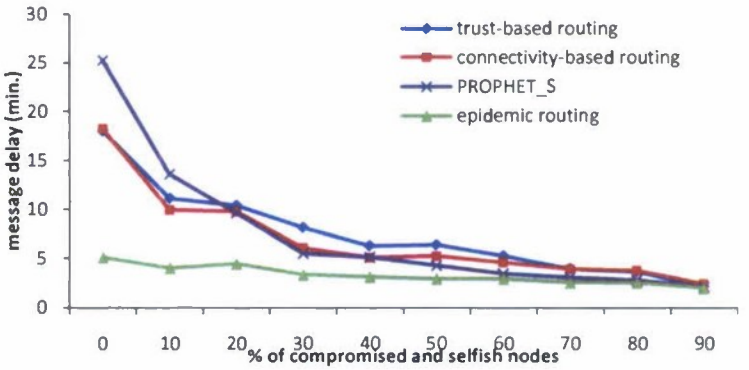


Figure 4: Performance Comparison in Message Delay.

Figure 4 shows the average delay experienced per message considering only those messages delivered successfully. Here we first note that in general connectivity-based routing performs better than trust-based routing because connectivity-based protocols use the delay to encounter the next message carrier (e-connectivity) and the delay for the next message carrier to encounter the destination node (d-connectivity) as the criteria to select a message carrier. The result suggests that if delay is of primary concern, we should set the weights associated with e-connectivity and d-connectivity (QoS trust metrics) higher than those for honesty and unselfishness (social trust metrics), as connectivity-based routing does (by setting $w_1 : w_2 : w_3 : w_4 = 0.5 : 0.5 : 0 : 0$). This will have the effect of trading off high delivery ratio for low delay. Figure 4 also shows that connectivity-based routing approaches the ideal performance obtainable from epidemic routing as the percentage of malicious and selfish nodes increases. We also observe that in general PROPHET_S, being a protocol using d-connectivity for routing, performs better than trust-based routing but worse than connectivity-based routing. The reason

that PROPHET_S performs worse than connectivity-based routing is that it only compares d-connectivity values of two encountering nodes for routing decisions, which is not effective in minimizing the end-to-end delay. We note that this effect is especially pronounced when the population of malicious and selfish nodes is low, since in this condition PROPHET_S even performs worse than trust-based routing which considers both e-connectivity and d-connectivity as part of its trust composition. A main reason for this performance deterioration of PROPHET_S in message delay when the population of malicious and selfish nodes is low is that in this condition most new encounters would be good nodes, so the effect of connectivity dominates the effect of node maliciousness/selfishness for deciding the next message carrier, and PROPHET_S comparing d-connectivity values of two encountering nodes for routing decisions is not effective in minimizing the end-to-end delay.

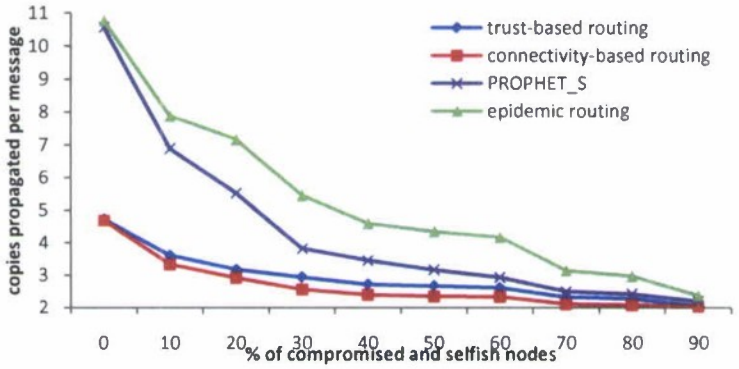


Figure 5: Performance Comparison in Message Overhead.

Figure 5 compares the three protocols in message overhead measured by the number of copies forwarded to reach the destination node for those messages successfully delivered. We see that trust-based protocols perform comparably with connectivity-based protocols and both protocols outperform epidemic routing and PROPHET_S considerably in message overhead. The reason that trust-based protocols use slightly more message copies than connectivity-based routing protocols is that the path being selected by trust-based protocols may not be the most direct route in order to avoid selfish or malicious nodes. The reason that both trust-based routing and connectivity-based routing outperform PROPHET_S, especially when the population of malicious and selfish nodes is low, is that as explained earlier PROPHET_S tends to generate a longer route, thus resulting in more message copies being propagated.

In summary, from Figures 3-5, we see that trust-based protocols can effectively trade off message delay (Figure 4) for a significant gain in message delivery ratio (Figure 3) and message overhead (Figure 5) over connectivity-based routing, epidemic routing, and PROPHET_S.

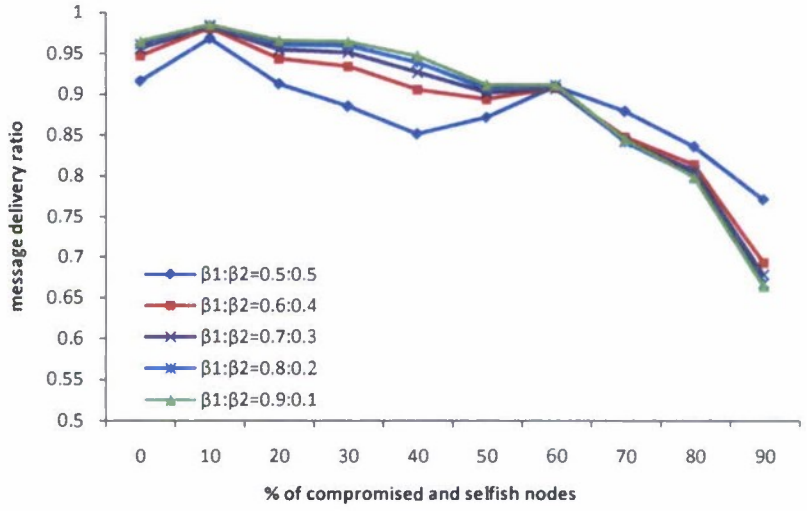


Figure 6: Effect of β_1:β_2 on Delivery Ratio of Trust-based Routing.

By comparing the performance of trust-based routing (w_1: w_2: w_3: w_4 = 0.25:0.25:0.25:0.25) and connectivity-based routing (w_1: w_2: w_3: w_4 = 0.5:0.5:0:0), we have demonstrated the effect of parameters w_1: w_2: w_3: w_4 on system performance. Figure 6 investigates the effect of β_1:β_2, on performance of trust-based protocols with β_1:β_2 varying from 0.5:0.5 to 0.9:0.1. We observe that as β_1:β_2 increases (using a higher weight on direct trust), the message delivery ratio increases if the population of malicious/selfish nodes is low; otherwise, the delivery ratio decreases. This result means that when the population of malicious/selfish nodes is low, one should use a higher ratio of β_1:β_2 to improve the protocol performance. We attribute this to the fact that when the population of malicious/selfish nodes is low, it is easy for any newly encountered node to qualify as a recommender and provide a trust recommendation toward all other nodes in the DTN. However, because of trust decay of indirect recommendations, i.e., due to the product term in Equation 4, the indirect trust value received will likely decrease. Consequently, a good node may unnecessarily underestimate the trust values of other good nodes in the system. To avoid this, it is better to place a higher weight on *direct trust* if there are a lot of good nodes around to serve as recommenders. Here, we note that when given knowledge of the percentage of malicious and

selfish nodes, the sensitivity analysis performed above helps identify the best ratio of β_1:β_2 to maximize the protocol performance.

7. Conclusion

In this paper, we have proposed and analyzed a class of trust-based routing protocols in delay tolerant networks. The most salient feature of our protocol is that we consider not only connectivity (QoS trust) but also honesty and unselfishness (social trust) properties into a composite trust metric for decision making in DTN routing dynamically. We formally proved that our protocol is resilient against whitewashing, bad-mouthing, and good-mouthing attacks by malicious nodes. We further substantiated the claim with numerical results demonstrating that a malicious node will never attain trustworthy status. Our performance analysis results demonstrate that by properly selecting weights associated with QoS and social trust metrics for trust evaluation, our trust management protocols can achieve the ideal performance level in delivery ratio and delay obtainable by epidemic routing, especially as the percentage of malicious and selfish nodes increases. In particular, trust-based protocols that consider both social and QoS trust can effectively trade off message delay for a significant gain in message delivery ratio and message overhead over connectivity-based routing, epidemic routing, and PROPHET routing protocols.

In the future, we plan to investigate other forms of message passing such as multi-copy message forwarding and other forms of attacks by malicious nodes such as jamming, forgery, and DoS attacks. We also plan to consider other trust metrics such as technical competence, betweenness centrality, similarity, and social ties (strength) [6].

References

[1] M. Aldebert, M. Ivaldi, and C. Roucolle, "Telecommunications Demand and Pricing Structure: an Economic Analysis," *Telecommunication Systems*, vol. 25, no. 1-2, Jan. 2004, pp. 89-115.

[2] J. Burgess, B. Gallagher, D. Jensen, and B.N. Levine, "Maxprop: Routing for Vehicle-Based Disruption-Tolerant Networking," *IEEE INFOCOM 2006*, Barcelona, Spain, April 2006, pp. 1-11.

[3] J.H. Cho, A. Swami and I.R. Chen, "Modeling and Analysis of Trust Management for Cognitive Mission-driven Group Communication Systems in Mobile Ad Hoc Networks," *7th IEEE/IFIP Int. Symp. Trusted Computing and Communications*, Vancouver, Canada, Aug. 2009.

[4] G. Ciardo, R.M. Fricks, J.K. Muppala and K.S. Trivedi, *Stochastic Petri Net Package Users Manual*, Department of Electrical Engineering, Duke University, 1999.

[5] E.M. Daly and M. Haahr, "The Challenges of Disconnected Delay Tolerant MANETs," *Ad Hoc Networks*, vol. 8, no. 2, March 2010, pp. 241-250.

[6] E.M. Daly and M. Haahr, "Social Network Analysis for Information Flow in Disconnected Delay-Tolerant MANETs," *IEEE Transactions on Mobile Computing*, vol. 8, no. 5, May 2009, pp. 606-621.

[7] M. Karaliopoulos, "Assessing the Vulnerability of DTN Data Relaying Schemes to Node Selfishness," *IEEE Communications Letters*, vol. 13, no. 12, 2009, pp. 923-925.

[8] E. Bulut, Z. Wang and B.K. Szymanski, "Impact of Social Networks on Delay Tolerant Routing," *IEEE Globecom 2009*, Hawaii, USA, Nov. 2009, pp. 1804-1809.

[9] Q. Li, S. Zhu, and G. Cao, "Routing in Socially Selfish Delay Tolerant Networks," *IEEE Infocom 2010*, San Diego, CA, March 2010, pp. 1-9.

[10] S. Jain, K. Fall, and R. Patra, "Routing in a Delay Tolerant Network," *ACM Computer Communication Review*, vol. 34, no. 4, Oct. 2004. pp. 145-158.

[11] S.C. Nelson, M. Bakht and R. Kravets, "Encounter-based Routing in DTNs," *IEEE Infocom 2009*, Rio De Janeiro, Brazil, Apr. 2009, pp.846-854.

[12] U. Shevade, H. Song, L. Qiu, and Y. Zhang, "Incentive-Aware Routing in DTNs," *16th IEEE Conf. on Network Protocols*, Orlando, FL, USA, Oct. 2008, pp. 238-247.

[13] A. Vahdat and D. Becker, "Epidemic Routing for Partially Connected Ad Hoc Networks," Technical Report, Computer Science Department, Duke University, 2000.

[14] Z. Xu, et al. "SReD: A Secure Reputation-Based Dynamic Window Scheme for Disruption-Tolerant Networks," *IEEE Military Communications Conf.*, Oct. 2009, pp. 1-7.

[15] H. Zhu, X. Lin, R. Lu, Y. Fan, and X. Shen, "SMART: A Secure Multilayer Credit-Based Incentive Scheme for Delay-Tolerant Networks", *IEEE Transactions on Vehicular Technology*, vol. 58, no. 8, Oct. 2009, pp. 4628-4638.

[16] I.R. Chen, F. Bao, M.J. Chang, and J.H. Cho, "Trust Management for Encounter-based Routing in Delay Tolerant Networks," *IEEE Global Communications Conference*, Miami, USA, Dec. 2010.

[17] C. Crepeau, C.R. Davis and M. Maheswaran, "A Secure MANET Routing Protocol with Resilience against Byzantine Behaviours of Malicious or Selfish Nodes," *21st International Conference on Advanced Information Networking and Applications Workshops*, Niagara Falls, Ontario, Canada, May 2007.

[18] A. Lindgren, A. Doria, O. Schelen, "Probabilistic Routing in Intermittently Connected Networks," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 7, no. 3, Jul. 2003, pp. 19-20.

[19] E. Ayday, H. Lee, and F. Fekri, "Trust Management and Adversary Detection for Delay Tolerant Networks," *IEEE Military Communications Conf.*, Oct. 2010, pp. 1788-1793.

[20] M. Chuah, P. Yang, and J. Han, "A Ferry-based Intrustion Detection Scheme for Sparsely Connected Ad Hoc Networks," *4th Annual International Conference on Mobile and Ubiquitous Systems: Networking & Services*, Aug. 2007, pp. 1-8.