REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
15-02-2011	Quarterly technical report	15 Nov 2010 - 14 Feb 2011

4. TITLE AND SUBTITLE
Trust-Threshold Based Routing in Mobile Ad Hoc Delay Tolerant Networks

5a. CONTRACT NUMBER

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

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Chang, MoonJeong (VT)
Chen, Ing-Ray (VT)
Bao, Fenye (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 a trust-threshold based routing protocol for delay tolerant networks, leveraging two trust thresholds for accepting recommendations and for selecting the next message carrier for message forwarding. We show that there exist optimal trust threshold values under which trust-threshold based routing performs the best in terms of message delivery ratio, message delay and message overhead. By means of a probability model, we perform a comparative analysis of trust-threshold based routing against epidemic, social-trust-based and QoS-trust-based routing. Our results demonstrate that trust-threshold based routing operating under proper trust thresholds can effectively trade off message delay and message overhead for a significant gain in message delivery ratio. Moreover, our analysis helps identify the optimal weight setting to best balance the effect of social vs. QoS trust metrics to maximize the message delivery ratio without compromising message delay and/or message overhead requirements.

15. SUBJECT TERMS
Delay tolerant networks, message routing, trust management, social trust, QoS trust, trust-threshold based routing, performance analysis.

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

Trust-Threshold Based Routing in Mobile Ad Hoc Delay Tolerant Networks

MoonJeong Chang[1], Ing-Ray Chen[1], Fenye Bao[1] and Jin-Hee Cho[2]

[1]Department of Computer Science,
Virginia Tech, 7054 Haycock Road, Falls Church, VA 22043, USA
{mjchang, irchen, baofenye}@vt.edu
[2]Computational and Information Sciences Directorate,
U.S. Army Research Laboratory, 2800 Powder Mill Road, Adelphi, MD 20783, USA
jinhee.cho@us.army.mil

Abstract. We propose a trust-threshold based routing protocol for delay tolerant networks, leveraging two trust thresholds for accepting recommendations and for selecting the next message carrier for message forwarding. We show that there exist optimal trust threshold values under which trust-threshold based routing performs the best in terms of message delivery ratio, message delay and message overhead. By means of a probability model, we perform a comparative analysis of trust-threshold based routing against epidemic, social-trust-based and QoS-trust-based routing. Our results demonstrate that trust-threshold based routing operating under proper trust thresholds can effectively trade off message delay and message overhead for a significant gain in message delivery ratio. Moreover, our analysis helps identify the optimal weight setting to best balance the effect of social vs. QoS trust metrics to maximize the message delivery ratio without compromising message delay and/or message overhead requirements.

Keywords: Delay tolerant networks, message routing, trust management, social trust, QoS trust, trust-threshold based routing, performance analysis.

1 Introduction

Delay/Disruption Tolerant Networks (DTNs) are self-organizing wireless networks with the characteristics of large latency, intermittent connectivity, and limited resources (e.g., battery, computational power, bandwidth) [1, 2]. Different from the traditional networks such as mobile ad hoc networks, the nodes in DTNs forward messages to a destination in a *store-carry-and-forward* manner [1, 2] in order to cope with the absence of guaranteed end-to-end connectivity. That is, an intermediate node stores a message received from a sender and carries it, and then forwards it to an encountered node which continues the store-carry-and-forward process until the message reaches the destination node. In such environments, the key challenge is to select an appropriate "next message carrier" among all encountered nodes to maximize the message delivery ratio while minimizing message overhead and delay. Further, we face additional challenges due to a lack of centralized trust entity. The open, distributed, and dynamic nature of DTNs also induces security vulnerability [2, 3]. In this paper, we consider a DTN in the presence of malicious and uncooperative

nodes and propose a method for the selection of trustworthy message carriers with the goal of maximizing the message delivery ratio without compromising message delay or message overhead in the context of DTN routing.

Most current DTN routing protocols are based on encounter patterns [4-7]. The problem is that if the predicted encounter does not happen, then messages would be lost for single-copy routing, or flooded for multi-copy routing. Moreover, in the presence of selfish or malicious nodes, these approaches still could not guarantee reliable message delivery. The vulnerability of DTN routing to node selfishness was well studied in [8]. Several reeent studies [9-13] used reputation to select message carriers among encountered nodes and encouraged cooperative behaviors using credit incentives. However, a eentralized eredit management system which can be a single point of failure is typically required, as it is challenging to perform distributed eredit management in a DTN in the presence of selfish or malieious nodes.

The rapid proliferation of miniaturized wireless deviees sueh as mobile phones, smart phones, and PDAs makes humans beeome deviee-carriers. Since eommunications between sueh devices are possible only when in elose proximity, their eontacts are elosely related to the social relationship or interactions [14]. From this perspeetive, recently there have been several social network based approaehes [15-21] to select the best message carrier in DTNs. [15-19] considered soeial relationship and social networking as eriteria to seleet message earriers in DTNs. However, no eonsideration was given to the presence of malicious or selfish nodes; [20] 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; [21] considered social trust based on friendship, familiarity, and similarity in order to thwart Sybil attacks in DTNs.

This work extends from our earlier work [22] on trust-based routing in DTNs. We also take social networking into eonsideration in designing DTN routing protocols. However, unlike prior work cited above, we integrate *social trust* and *Quality of Service (QoS) trust* into a composite trust metrie for determining the best node among the new encounters for message forwarding. In this work, we propose the design notion of trust thresholds for determining the trustworthiness of a node acting as a recommender or as the next message carrier, and analyze the best thresholds under which trust-threshold based routing in DTNs would perform the best. Our approaeh is distributed in nature and does not require a complicated credit management system. Each node will run the proposed trust-threshold based routing protocol individually to assess trust of its peers using the same trust threshold setting, and consequently seleet trustworthy nodes as carriers for message routing. Without loss of generality, we consider *healthiness* and *cooperativeness* for social trust to account for a node's trustworthiness for message delivery, and *connectivity* and *energy* for QoS trust to account for a node's QoS capability to quickly deliver the message to the destination node. We perform a eomparative analysis of the resulting trust-threshold based routing algorithm with epidemic routing [23], social-trust-based routing (for which only social trust metrics are eonsidered) and QoS-trust-based routing (for which only QoS trust metrics are considered) and identify conditions including the best trust thresholds to be used under whieh trust-threshold based routing outperforms these baseline routing algorithms for a DTN eonsisting of heterogeneous mobile nodes with vastly different social and networking behaviors.

2 System Model

We consider a DTN environment without a centralized trust authority. Nodes communicate through multi-hop wireless links. Every node may have a different level of energy and speed reflecting node heterogeneity. We differentiate uncooperative nodes from malicious nodes. An uncooperative node acts to maximize its own benefit regardless of the global benefit of the DTN. So it may drop packets arbitrarily just to save energy but it may decide to forward a packet if it has a good social tie with the 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, and even forge packets. As soon as a malicious node is detected, the trust value of the malicious node will be set to zero and thus exclude it as a message carrier for message forwarding. A node initially may be healthy but become compromised because of being captured for example. Once a node is compromised, it is a malicious node. In the paper, we will use the terms malicious node and compromised node interchangeably.

We consider the following energy model. The energy level of a node is related to the social encountering activities of the node. If a node becomes uncooperative, the speed of energy consumption is slowed down. If a node becomes compromised, the speed of energy consumption will increase since the node may perform attacks which may consume more energy. Since we assume that wireless devices can be carried by people, the residual energy level does not affect a node's speed.

A node's trust value is assessed based on direct observations through monitoring, snooping, and overhearing, and indirect information like recommendations. To counter whitewashing or false information attacks, nodes do not use status exchange information including encounter history information because a malicious node can provide fake encounter history information to other nodes [24, 25]. For indirect information, we use recommendations obtained only from 1-hop neighbors to cope with fragile connectivity and sparse node density in DTNs. The trust of one node toward another node is updated upon an encounter event. Our trust metric consists of two trust types: *social trust* and *QoS trust*. *Social trust* is based on social relationships. We consider ***healthiness*** (or honesty) and ***cooperativeness*** to measure the social trust level of a node. Social network structure-based properties such as similarity, centrality, and betweenness are not considered because we do not use trust encounter histories exchanged to avoid self-promoting or false information attacks by malicious nodes. *QoS trust* is evaluated through the communication networks by the capability of a node to deliver messages to the destination node. We consider ***connectivity*** and ***energy*** to measure the QoS trust level of a 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.

3 Trust-Threshold Based Routing

Our trust-threshold based routing algorithm builds upon the notion of peer-to-peer trust evaluation at runtime. A node will evaluate its peers dynamically and will use trust thresholds as criteria to determine if it can trust a node as a recommender or as a message carrier. Two trust thresholds are used: the recommender threshold denoted

by T_{rec} and the message forwarding threshold denoted by T_f. 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 healthiness, cooperativeness, connectivity, and energy selected as the social and QoS trust components in this paper. Specifically node i will compute $T_{i,j}(t)$ by:

$$T_{i,j}(t) = w_1 T_{i,j}^{healthiness}(t) + w_2 T_{i,j}^{cooperativeness}(t) + w_3 T_{i,j}^{connectivity}(t) + w_4 T_{i,j}^{energy}(t) \quad (1)$$

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, $T_{i,j}^{healthiness}(t)$ is about node i's belief in node j's honesty; $T_{i,j}^{cooperativeness}(t)$ is about node i's belief in node j's cooperativeness; $T_{i,j}^{connectivity}(t)$ is about node i's belief in node i's connectivity to node j, representing the delay of node i passing the message to node j; $T_{i,j}^{energy}(t)$ is about node i's belief in node j's energy. In message forwarding in DTNs, two most important performance metrics are message delivery ratio and message delay. The rationale of using these four trust metrics is to rank nodes such that high $T_{i,j}^{connectivity}(t)$ and $T_{i,j}^{energy}(t)$ represent low delay, while high $T_{i,j}^{healthniess}(t)$ and $T_{i,j}^{cooperativeness}(t)$ lead to high delivery ratio. We set $T_{i,j}^{healthiness}(0)$, $T_{i,j}^{cooperativeness}(0)$, $T_{i,j}^{connectivity}(0)$ and $T_{i,j}^{energy}(0)$ to ignorance (0.5) since initially there is no information exchanged among nodes. Specifically, node i will update its trust toward node j upon encountering node m at time t for the duration $[t, t + \Delta t]$ as follows:

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

Here X refers to a trust property (i.e., healthiness, cooperativeness, connectivity, or energy). In Equation 2, β_1 is a 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 another parameter to weigh indirect information from the recommender, i.e., "other-information," with $\beta_1 + \beta_2 = 1$. Here we note that typically $\beta_1 > \beta_2$ because cognitive nodes in a DTN tend to trust direct observations more than indirect recommendations.

3.1 Direct Observation Evaluation

$$T_{i,j}^{direct,X}(t + \Delta t) = \begin{cases} T_{i,m}^{encounter,X}(t + \Delta t), & if\ m = j \\ e^{-\lambda_d \Delta t} \times T_{i,j}^{X}(t), & if\ m \neq j \end{cases} \quad (3)$$

The direct trust evaluation of node j is given in Equation 3 above in which if the new encounter (node m) is node j itself, then node i can directly evaluate node j because node i and node j are 1-hop neighbors. 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 $t + \Delta t$. If the new encounter is not node j, then no new direct information can be gained about node j. So, node i will use its past trust toward node j obtained at time t decayed over the time interval Δt to model decay of trust over time. We adopt an exponential time decay factor, $e^{-\lambda_d \Delta t}$ ($0 < \lambda_d \leq 0.1$ to limit

the decay to at most 50%). Below we describe how direct trust evaluation for each trust component value $T_{i,j}^{direct,X}(t)$ can be obtained based on direct observations:

- $T_{i,j}^{healthiness,\ direct}(t)$: This provides the belief of node i that node j is not compromised based on node i's direct observations toward node j. Node i can monitor node j's unhealthiness evidences including dishonest trust recommendation, false self-reporting [26], and abnormal traffic over the time period $[0, t]$ to estimate $T_{ij}^{healthiness,\ direct}(t)$. It could be computed by the number of bad experiences in healthiness over the total healthiness experiences.
- $T_{i,j}^{cooperativeness,\ direct}(t)$: This provides the degree of node j's cooperativeness as evaluated by node i based on direct observations over the time period $[0, t]$. Node i can apply overhearing or snooping techniques to detect cooperativeness behaviors and may give recent interaction experiences a higher priority over old experiences in estimating $T_{i,j}^{cooperativeness,\ direct}(t)$. It could be computed by the number of bad experiences in cooperativeness over the total cooperativeness experiences.
- $T_{i,j}^{connectivity,\ direct}(t)$: This provides the probability of encountering node j by node i at time t. It can be computed by the number of encounters between nodes i and j over the maximum number of encounters between node i and any other node over the time period $[0, t]$.
- $T_{i,j}^{energy,\ direct}(t)$: This provides the belief of node i toward node j's energy status based on direct observations toward node j. Node i can overhear or even monitor node j's packet transmission activities over the time period $[0, t]$ to estimate energy consumption of node j and compute $T_{i,j}^{energy,\ direct}(t)$ as the percentage of energy remaining in node j.

3.2 Indirect Information Evaluation

We use recommendations only from 1-hop neighbors because nodes in a DTN may not be able to connect to remote nodes due to fragile connectivity or sparse node density. Here we note that node i will not do indirect trust evaluation toward a newly encountered node, say node m, because node i and node m would be within 1-hop upon encounter, so node i will do direct trust evaluation toward node m instead, as discussed in Section 3.1.

We define the recommender trust threshold T_{rec} such that if $T_{i,j}(t) > T_{rec}$, node i will consider node j as a "trustworthy" recommender (or plainly as a good node) at time t. If node i believes that a neighbor, say node c, is a good node, i.e., $T_{i,c}(t + \Delta t) > T_{rec}$, node i will use node c as a recommender to update its beliefs toward other nodes.

$$T_{i,j}^{indirect,\ X}(t + \Delta t) = \begin{cases} e^{-\lambda_d \Delta t} \times T_{i,m}^{X}(t), & if\ m = j \\ e^{-\lambda_d \Delta t} \times T_{i,j}^{X}(t), & if\ m \neq j\ and\ |R_i| = 0 \\ \dfrac{\sum_{c \in R_i} \{T_{i,c}^{X}(t + \Delta t) \times T_{c,j}^{X}(t + \Delta t)\}}{\sum_{c \in R_i} T_{i,c}^{X}(t + \Delta t)}, & if\ m \neq j\ and\ |R_i| > 0 \end{cases} \quad (4)$$

The indirect trust evaluation toward node j is given in Equation 4 above where R_i is a set containing node i's 1-hop neighbors with $T_{i,c}(t + \Delta t) > T_{rec}$, and $|R_i|$ indicates the cardinality of R_i. If the new encounter is node j itself, then there is no indirect recommendation for node j, so node i will use its past trust toward node j obtained at time t decayed over the time interval Δt to model decay of trust over time. If the new encounter is not node j and node i considers node c as trustworthy, i.e., $T_{i,c}(t + \Delta t) > T_{rec}$, then node c can provide its recommendation to node i for evaluating node j. In this case, node i weighs the recommendation provided by node c by normalizing it with referral trust. Moreover, the more recommendations node i receives from trustworthy nodes, the more accurate the trust value of node j can be. Our recommender trust threshold design provides robustness against bad-mouthing or good-mouthing attacks since only recommendations from good nodes are taken into consideration.

3.3 Message Routing

$T_{i,j}(t)$ 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 and a forwarding trust-threshold (T_f) in this paper, i.e., node i will pass the message to node m if $T_{i,m}(t + \Delta t) \geq T_f$ as well as $T_{i,m}(t)$ is in the top Ω percentile among all $T_{i,j}(t)$'s. Here, T_f is defined as a minimum trust threshold for the selection of the next message carrier. The reason for using T_f is to guarantee that a next message carrier is trustworthy. The performance metrics of interest are message delivery ratio, message delay and message overhead. We consider only single-copy message routing and buffer management is not considered in this paper. Below we develop a performance model to identify the best message forwarding threshold T_f (for accepting the next message carrier) and the best recommendation threshold T_{rec} (for accepting a recommender) to optimize performance of trust-threshold based routing in DTNs, as well as for performance comparison with baseline message routing protocols.

4 Performance Model

We develop a probability model to analyze the performance of the proposed trust-threshold based routing protocol for DTN message forwarding. The probability model is based on stochastic Petri net (SPN) techniques [27] due to its ability to handle a large number of states. The SPN model is shown in Fig. 1 consisting of 5 event subnets, namely, in clockwise order, energy, location, cooperativeness, intrusion detection, and compromise. The purpose of the SPN model is to yield the ground truth status of a node (i.e., healthiness, cooperativeness, connectivity, and energy) in the presence of uncooperative and malicious nodes and to derive the trust relationship with other nodes in the system. Without loss of generality, we consider a square-shaped operational area consisting 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

locations in four directions (i.e., north, west, south, and east) in accordance with its mobility rate. To avoid end-effects, movement is bounced back. The *location subnet* produces the probability that node i is in a particular location L at time t. This information along with the location information of other nodes at time t provides us the probability of two nodes encountering with each other.

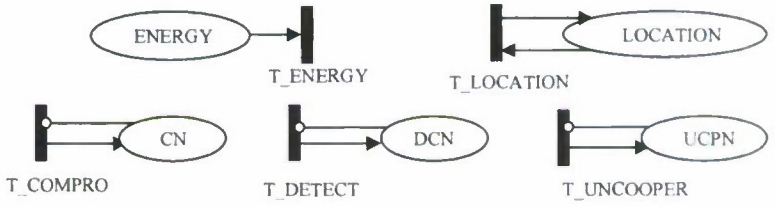


Fig. 1. SPN Model.

Below we explain how we construct the SPN model for describing a node's ground truth status in terms of its location, energy level, degree of healthiness (i.e., whether or not a node is compromised or/and detected), and degree of cooperativeness.

Energy: We use the *energy subnet* to describe the energy status of a node. 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 rate of transition T_ENERGY indicates the energy consumption rate which depends on the ground truth status of the node (i.e., uncooperativeness and healthiness).

Location: We use the *location subnet* to describe the location status of a node. 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 calculated as σ_0/R based on the node speed σ_0 and wireless radio range R.

Connectivity: We use the *connectivity subnet* to measure connectivity of node i to node j by the time-averaged probability that node i and node j are within one hop during $[t\text{-}n\Delta t, t]$, modeling not only chances, but also recency of encountering events between node i and node j. This can be obtained by knowledge of location probabilities of node i and node j during $[t\text{-}n\Delta t, t]$. Without considering recency, the interval would be $[0, t]$.

Healthiness: We use the *compromise subnet* and the *intrusion detection subnet* to describe the healthiness status of a node. A node becomes compromised when transition T_COMPRO fires and then a token is put in place *CN* to represent the node has been captured and compromised. The rate to T_COMPRO is λ_{com}, the per-node compromising rate given as input to the SPN model. Our model is generic in handling intrusion detection as follows. In case an *intrusion detection system* (IDS) exists, it would be characterized by a false negative probability P_{fn} and a false positive probability P_{fp} given as input to the SPN model. If the node is compromised and it is detected by the IDS, transition T_DETECT fires and a token moves to place *DCN*. The transition rate to T_DETECT is given by $(1 - P_{fn})/T_{IDS}$ where T_{IDS} is the IDS

detection interval and $1 - P_{fn}$ is the probability that the IDS correctly detects the compromised node. If the node is good but is falsely identified as a bad node, transition T_DETECT also fires and a token moves to place *DCN*. The transition rate to T_DETECT is given by P_{fp}/T_{IDS} and P_{fp} is the probability that the IDS incorrectly diagnoses a good node as a bad node. Thus, the transition rate to T_DETECT is a weighted sum of these two transition rates, conditioning on if the node is compromised or not, which we can easily determine from the SPN model. In case an IDS does not exist, this intrusion detection subnet would not exist and can be removed from the SPN model.

Cooperativeness: We use the *cooperative subnet* to describe the cooperative status of a node. Place *UCPN* indicates whether a node is uncooperative or not. If a node becomes uncooperative, a token goes to *UCPN* by triggering T_UNCOOPER. The transition rate to T_UNCOOPER is $\lambda_{uncooper,}$ the per-node uncooperative rate given as input to the SPN model.

The SPN model described above yields the "ground truth" status of each node, which would allow us to calculate $T_{i,j}^{X}(t)$ as follows. When node i encounters node j, node i will assess node j in trust property X to yield $T_{i,j}^{encounter,\ X}(t)$ based on its past experiences up to time t. Because node i has prior close interaction experiences with node j (including the current encounter), node i has good knowledge about whether node j is cooperative or not through snooping and overhearing. Hence, node i's direct assessment in node j's cooperativeness at the encounter time t is the same as or close to the ground truth cooperativeness status of node j at time t. Consequently, $T_{i,j}^{encounter,\ cooperativeness}(t)$ in Equation 3 is simply equal to the probability that place *UCPN* in node j does not contain a token at time t, which we can compute easily from the SPN model. Similarly, node i can fairly accurately assess $T_{i,j}^{encounter,\ connectivity}(t)$ by consulting its encounter history with node j over $[t - n\Delta t, t]$. This quantity can be obtained by utilizing the SPN output regarding the node location probability at time t. For the healthiness trust component, in case an IDS exists, node i knows that node j is malicious when it is detected and a message is announced to the system, i.e., when node j's place *DCN* (in Fig.1) is not zero. Thus, we can compute $T_{i,j}^{encounter,\ healthiness}(t)$ by the probability that place *DCN* in node j does not contain any token at time t. In case an IDS does not exist, we can approximate $T_{i,j}^{encounter,\ healthiness}(t)$ by the probability that place *CN* in node j does not contain any token at time t. Lastly, node i can overhear or even monitor node j's packet transmission activities over the time period $[0, t]$ to estimate $T_{ij}^{encounter,\ energy}(t)$, which would be close to the ground truth energy status of node j and can be obtained easily from the SPN output by inspecting place *ENERGY*. Once $T_{i,j}^{encounter,\ X}(t)$ is obtained, node i can update its $T_{i,j}^{X}(t)$ based on Equation 2, and subsequently, can obtain $T_{i,j}(t)$ based on Equation 1.

5 Results

In this section, we show numerical results and provide physical interpretation of the results obtained. Table 1 lists the default parameter values used. For trust-threshold based routing, we set $w_1{:}w_2{:}w_3{:}w_4 = 0.25{:}0.25{:}0.25{:}0.25$ for healthiness: cooperativeness: connectivity: energy. We setup 20 nodes with vastly different initial energy levels in the system moving randomly in a 8×8 operational region with the mobility rate of each node being σ_0 in the range of [1,4] m/sec, and with each area covering 250 m radio radius. There are two sets of nodes, namely, good nodes and bad nodes (i.e., uncooperative and/or malicious nodes). Good nodes are the ones with the compromise rate being zero and the uncooperative rate being zero. Uncooperative nodes have a non-zero uncooperative rate $\lambda_{uncooper}$, and once they become uncooperative they stay as uncooperative. Compromised nodes have a non-zero compromise rate λ_{com} in the range of [1/480min, 1/160min]. We assume an IDS exists with the false negative/positive probability being 1%. For indirect trust evaluation, we use recommendations only from 1-hop neighbors whose trust is higher than the recommender threshold T_{rec}. We set $\beta_1{:}\beta_2$=0.8:0.2 to place higher trust on direct observations. 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 set λ_d, the decay coefficient, to 0.001 (such that $e^{-\lambda_d \Delta t} = 0.995$) to model small trust decay with time. Trust-threshold routing is performed based on the algorithm described in Section 3 using the message forwarding threshold (T_f) being applied to all nodes in a DTN.

Table 1. Parameters and Their Default Values.

Parameter	Value	Parameter	Value	Parameter	Value	Parameter	Value
$m \times m$	8×8	R	250m	$1/\lambda_{uncooper}$	300 s	λ_d	0.001
P_{fn}, P_{fp}	1%	Δt	300 s (5 min)	σ_0	[1, 4] m/s	Ω	90%
$\beta_1{:}\beta_2$	0.8:0.2	E_0	[12, 24] hrs	$1/\lambda_{com}$	[160, 480] min	T_{IDS}	300s

5.1 Optimal Trust Thresholds for Routing in DTNs

In this section, we investigate the optimal values of T_{rec} and T_f under trust-threshold based routing in DTNs. Note that T_f and T_{rec} are the minimum trust thresholds for the selection of a next message carrier and for the selection of recommenders, respectively. First, 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 allow nodes to accumulate experiences 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 uncooperative, 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 2000 runs from which the message delivery ratio, delay and overhead performance measurements are calculated.

Fig. 2 shows the effect of T_f and T_{rec} on message delivery ratio as the percentage of malicious and uncooperative nodes varies. We vary T_f from 0.6 to 0.9 and T_{rec} from 0.6 to 0.9 to cover a wide range of possible values. We see that the message delivery ratio becomes higher as T_f increases. Specifically, as the percentage of malicious and uncooperative nodes increases, the message delivery ratio becomes lower with T_f=0.6 or 0.7, while the message delivery ratio approaches 1 with T_f=0.8 or 0.9. The reason is that trust-threshold based routing behaves like a "direct delivery" approach as T_f increases, the effect of which is especially pronounced when there is a high malicious/uncooperative node population. More specifically, a carrier is likely to hold the message until it runs into a trustworthy node. There may be an extreme case where node i can store a message until it encounters the destination node because it could not encounter a node with trust greater than T_f. We also observe that T_f dominates T_{rec} in message delivery ratio which we observe is relatively insensitive to T_{rec}. This is mainly because using fewer (e.g., when T_{rec} = 0.8) or more recommenders (e.g., when T_{rec}= 0.6) to provide recommendations does not affect the indirect trust evaluation outcome much, as long as the recommenders are good nodes.

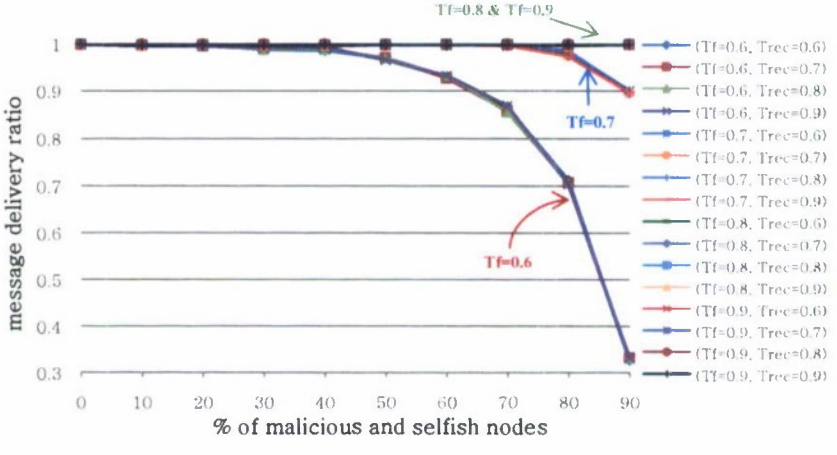
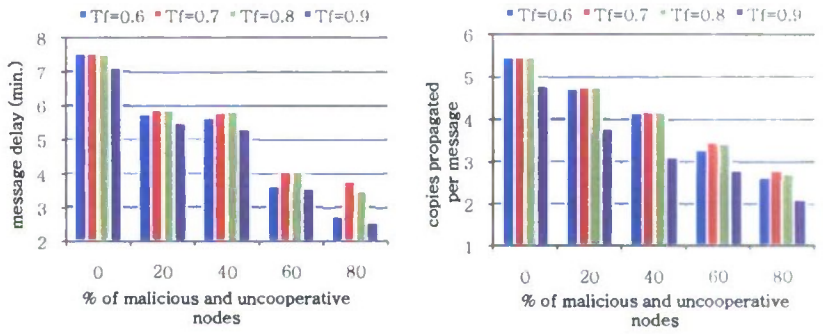


Fig. 2. Effect of T_f and T_{rec} on Message Delivery Ratio.

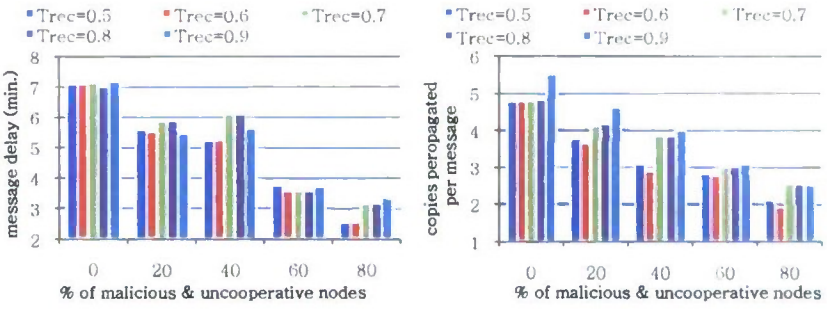
Fig. 3(a) and 3(b) show the message delay and message overhead (measured by the number of copies propagated per message), respectively, as a function of the percentage of malicious and/or uncooperative nodes, with T_f varying in the range of [0.6, 0.9] and T_{rec} fixed at 0.6 to isolate its effect. Here we only consider messages that are delivered successfully. We first observe that both the message delay and the message overhead decrease as the malicious/uncooperative node population increases

because of the smaller probability of encountering trustworthy nodes in message forwarding. In general, we see $T_f = 0.9$ consistently performs better than the others in terms of message delay and message overhead over a wide range of malicious/uncooperative node population. We attribute it to the fact that with $T_f = 0.9$, trust-threshold based routing behaves like "direct delivery" with very little copies being passed around to intermediate message carriers, resulting in a more direct route to reach the destination node. This is true in our DTN scenario where nodes can encounter each other with nonzero probability due to random movement. In situations where node movement is not random and the encountering probability may be zero or very small among certain nodes, $T_f = 0.9$ may not necessarily always perform the best. Our model helps identify the best T_f that minimizes the message delay/overhead.



(a) Message Delay. (b) Copies Propagated per Message.

Fig. 3. Effect of T_f on Message Delay and Message Overhead.



(a) Message Delay. (b) Copies Propagated per Message.

Fig. 4. Effect of T_{rec} on Message Delay and Message Overhead.

Fig. 4(a) and 4(b) show the message delay and message overhead, respectively, as a function of the percentage of malicious and/or uncooperative nodes, with T_{rec} varying in the range of [0.5, 0.9] and T_f fixed at 0.9 to isolate its effect. Here we see that $T_{rec} = 0.6$ performs slightly better than the other T_{rec} values in terms of the

message delay and the number of copies propagated per message. The reason is that the recommenders are all good nodes when $0.6 \leq T_{rec} \leq 0.9$ and $T_{rec} = 0.6$ allows more good recommenders to provide indirect recommendations, thus proving a more accurate indirect trust assessment. We also observe that $T_{rec} = 0.6$ has the shortest message delay and the lowest message overhead over a wide range of the percentage of malicious and uncooperative nodes.

In summary, we conclude that there exist optimal message forwarding threshold T_f and recommender threshold T_{rec} in trust-threshold based routing to best tradeoff message delivery ratio, message delay, and message overhead, adapting to application or network environment characteristics.

5.2 Comparative Performance Analysis of Trust-Threshold Based Routing

In this section, we perform a comparative analysis of trust-threshold based routing against epidemic routing [23], social-trust-based routing, and QoS-trust-based routing. For social-trust-based routing, we set $w_1 : w_2 : w_3 : w_4 = 0.5 : 0.5 : 0 : 0$, and for QoS-trust-based routing, we set $w_1 : w_2 : w_3 : w_4 = 0 : 0 : 0.5 : 0.5$. Here we note that social-trust-based routing and QoS-trust-based routing are special cases of trust-threshold based routing, with social-trust-based routing using only social trust metrics (healthiness and cooperativeness) and QoS-trust-based routing using only QoS trust metrics (connectivity and energy) for trust evaluation. Thus, the design concept of trust thresholds also applies to them. To show the effect of T_f, we evaluate the performance of these two routing algorithms with and without T_f. The dashed line is used for the "without T_f" case, while the solid line is for the "with T_f" case using the optimal T_f value identified in Section 5.1. Epidemic routing does not use the design parameter T_f, so only a solid line is shown for epidemic routing.

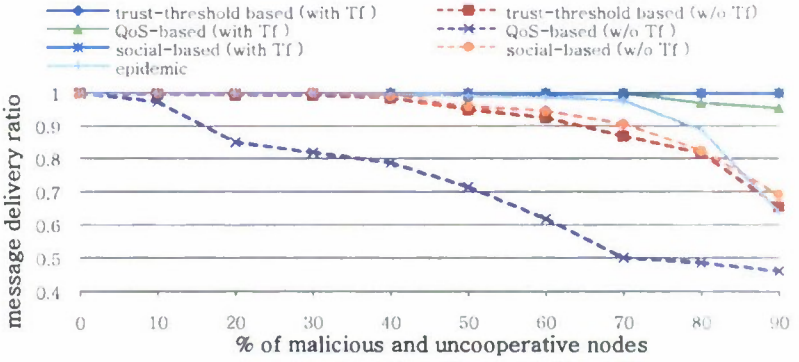


Fig. 5. Message Delivery Ratio ($T_{rec} = 0.6$, $T_f = 0.9$).

Fig. 5 shows the message delivery ratio as a function of the percentage of malicious and uncooperative nodes in a DTN. We see that the routing protocols with T_f outperform those without T_f in the delivery ratio. Also trust-threshold based routing with T_f and social-trust routing with T_f perform better than QoS-trust-based routing with T_f and epidemic routing, with the delivery ratio approaching 1 over a

wide range of malicious/uncooperative node population. This is attributed to the ability of trust-threshold based routing and social-trust-based routing being able to differentiate trustworthy nodes from uncooperative and malicious nodes and select trustworthy nodes to relay the message. We also note that performance of epidemic routing deteriorates when there is a high bad node population because it does not select trustworthy message carriers. This result demonstrates the effectiveness of incorporating social trust into the decision making process for DTN message routing, as well as using T_f to select the next message carrier to yield high delivery ratio.

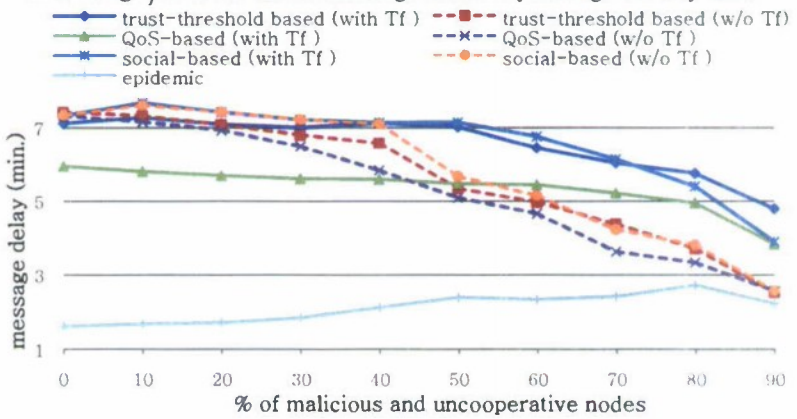


Fig. 6. Message Delay ($T_{rec} = 0.6,\ T_f = 0.9$).

Fig. 6 shows the average message delay experienced per message considering only those messages delivered successfully as a function of the percentage of malicious and uncooperative nodes. Here we first note that the message delay for all routing algorithms except epidemic routing decreases as the percentage of malicious and uncooperative nodes increases. This is because the probability of being able to forward the message to a good node decreases as more bad nodes exist in the system. Epidemic routing is insensitive to this because it is flood-based in nature and will try all possible routes to reach the destination node. As a result, the delay of epidemic routing represents the ideal smallest possible delay experienced for routing a message. Fig. 6 shows epidemic routing indeed performs the best among all in terms of delay. It also shows that with similar reasoning, all routing algorithms without T_f approach the ideal performance obtainable from epidemic routing as the percentage of malicious and uncooperative nodes increases.

Another result is that QoS-trust-based routing performs better than trust-threshold based routing and social-trust-based routing in terms of message delay. This is because QoS-trust-based routing only uses the connectivity QoS metric (representing the delay to encounter the next message carrier) and the level of the residual energy metric as the criteria to select a message carrier. This result indicates that if the objective is to minimize the message delay, we should set the weights associated with connectivity and energy (QoS trust metrics) considerably higher than those for healthiness and cooperativeness (social trust metrics) for trust-threshold based routing to approach the performance of QoS-trust-based routing in message delay.

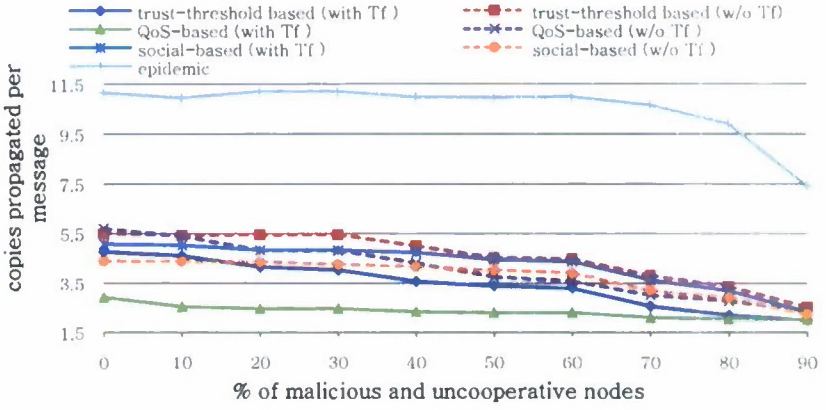


Fig. 7. Number of Copies Propagated per Message ($T_{rec} = 0.6$, $T_f = 0.9$).

Finally, Fig. 7 shows the message overhead measured by the number of copies forwarded to reach the destination node for those messages successfully delivered. We see that all trust-based routing algorithms, with or without T_f, outperform epidemic routing considerably in message overhead because trust is being utilized to regulate message forwarding. In particular, QoS-trust-based routing with T_f (the bottom curve) performs the best among all routing protocols. This result again reassures the effectiveness of our trust threshold design. The reason that trust-threshold based routing and social-trust-based routing use more message copies than QoS-trust-based routing is that the path selected by trust-threshold based routing or social-trust-based routing may not be the most direct route as they attempt to avoid uncooperative or malicious nodes. The result also suggests that if we want to minimize message overhead, we should set the weights associated with connectivity and energy (QoS trust metrics) considerably higher than those for healthiness and cooperativeness (social trust metrics) for trust-threshold based routing, in order to approach the performance of QoS-trust-based routing in message overhead.

In summary, from Figs. 5-7, we see that our proposed trust-threshold based routing algorithm operating under identified optimal T_f values can effectively trade off message overhead (Fig. 7) and message delay (Fig. 6) for a significant gain in message delivery ratio (Fig. 5). Moreover, our analysis results reveal that there exists an optimal weight setting in terms of $w_1 : w_2 : w_3 : w_4$ (e.g., social-trust-based vs. trust-threshold based vs. QoS-trust-based routing) to best balance the effect of social trust metrics vs. QoS trust metrics to maximize the message delivery ratio without compromising message delay and/or message overhead requirements.

6 Conclusion

In this paper, we have proposed and analyzed a trust-threshold based routing algorithm with the design objective to maximize the message delivery ratio while satisfying message delay and message overhead requirements. Our algorithm

leverages a trust management protocol incorporating both social and QoS trust metrics for peer-to-peer trust evaluation, as well as trust thresholds for selecting recommenders for indirect trust evaluation and for selecting the next message carrier for message forwarding. Our performance analysis results demonstrate that when operating under proper trust thresholds and social vs. QoS trust weight settings as identified in the paper, trust-threshold based routing can effectively trade off message delay and message overhead for a significant gain in message delivery ratio to achieve the design objective. In the future we plan to extend the research to investigate the best way to compose the overall trust metric from QoS and social trust components (not necessarily limited to the two QoS and two social trust components considered in this paper) and identify the optimal settings of design parameters, when given a DTN-based application characterized by a set of operational and environmental variables specifying the energy consumption model, the application failure model, the node compromise model, the social and QoS behavior model, the mobility model (random vs. random waypoint vs. multi-group-based vs. traces), and the application requirements (e.g., message delivery ratio, delay and overhead requirements for message routing applications).

References

1. Fall, K.: A Delay-Tolerant Network Architecture for Challenged Internets. In: ACM Special Interest Group on Data Communication, pp. 27-34. ACM Press, Karlsruhe (2003)
2. Fall, K., Farrell, S.: DTN: An Architectural Retrospective. IEEE Journal on Selected Areas in Communications, vol. 26, no. 5, pp. 828-836. IEEE press (2008)
3. Daly, E.M., Haahr, M.: The Challenges of Disconnected Delay Tolerant MANETs. Ad Hoc Networks, vol. 8, no. 2, pp. 241-250. Elsevier press (2010)
4. Burgess, J., Gallagher, B., Jensen, D., Levine, B.N.: Maxprop: Routing for Vehicle-Based Disruption-Tolerant Networking. In: 25th Conference on Computer Communications (INFOCOM), pp. 1-11. IEEE press, Barcelona (2006)
5. Jain, S., Fall, K., Patra, R.: Routing in a Delay Tolerant Network. ACM Computer Communication Review, vol. 34, no. 4, pp. 145-158. ACM press (2004)
6. Nelson, S.C., Bakht, M., Kravets, R.: Encounter-based Routing in DTNs. In: 28th Conference on Computer Communications, pp. 846-854. IEEE press, Rio De Janeiro (2009)
7. Lindgren, A., Doria, A., Schelen, O.: Probabilistic Routing in Intermittently Connected Networks. ACM SIGMOBILE Mobile Computing and Communications Review, vol. 7, no. 3, pp. 19-20. ACM press (2003)
8. Karaliopoulos, M.: Assessing the Vulnerability of DTN Data Relaying Schemes to Node Selfishness. IEEE Communications Letters, vol. 13, no. 12, pp. 923-925. IEEE press (2009)
9. Shevade, U., Song, H., Qiu, L., Zhang, Y.: Incentive-Aware Routing in DTNs. In: 16th IEEE Conference on Network Protocols, pp. 238-247. IEEE press, Orlando (2008)
10. Xu, A., Jin, Y.,Shu, W., Liu, X., Luo, J.: SReD: A Secure Reputation-Based Dynamic Window Scheme for Disruption-Tolerant Networks. In: IEEE Military Communications, pp. 1-7. IEEE press, Boston (2009)
11. Zhu, H., Lin,X., Lu,R., Fan,Y., Shen,X.: SMART: A Secure Multilayer Credit-Based Incentive Scheme for Delay-Tolerant Networks. IEEE Transactions on Vehicular Technology, vol. 58, no. 8, pp. 4628-4638. IEEE press (2009)
12. Chen, B., Chan, M.: MobiCent: A Credit-Based Incentive System for Disruption Tolerant Network. In: 29th Conference on Computer Communications (INFOCOM), pp. 875-883. IEEE press, San Diego (2010)

13. Lu, R., Lin, X., Zhu, H., Shen, X.: Pi: A Practical Incentive Protocol for Delay Tolerant Networks.IEEE Transactions on Wireless Communications, vol. 9, no. 4, pp.1483-1493. IEEE press (2010)
14. Sastry, N., Sollins, K., Crowcroft, J.: Delivery Properties of Human Social Networks. In: 28th Conference on Computer Communications (INFOCOM), pp. 2586-2590. IEEE press, Rio De Janeiro (2009)
15. Hui, P., Crowcroft,J., Yoneki, E.: BUBBLE Rap: Social-based Forwarding in Delay Tolerant Networks. In: ACM MobiHoc 2008, pp. 241-250. ACM press, Hong Kong SAR (2008)
16. Daly, E.M., Haahr, M.: Social Network Analysis for Information Flow in Disconnected Delay-Tolerant MANETs. IEEE Transactions on Mobile Computing, vol. 8, no. 5, pp. 606-621. IEEE press (2009)
17. Bulut, E., Wang, Z., Szymanski, B.K.: Impact of Social Networks on Delay Tolerant Routing. In: IEEE Global Communications Conference (GLOBECOM), pp. 1804-1809. IEEE press, Hawaii (2009)
18. Hossmann, T., Spyropoulos, T., Legendre, F.: Know Thy Neighbor: Towards Optimal Mapping of Contacts to Social Graphs for DTN Routing. In: 29th Conference on Computer Communications (INFOCOM), pp. 866-874. IEEE press, San Diego (2010)
19. Mtibaa, A., May, M., Diot, C., Ammar, M.: PeopleRank: Social Opportunistic Forwarding. In: 29th Conference on Computer Communications (INFOCOM), pp. 111-115. IEEE press, San Diego (2010)
20. Li, Q., Zhu, S., Cao, G.: Routing in Socially Selfish Delay Tolerant Networks. In: 29th Conference on Computer Communications (INFOCOM), pp. 857-865. IEEE press, San Diego (2010)
21. Trigunovic, S., Legendre, F., Anastasiades, C.: Social Trust in Opportunistic Networks. In: 29th Conference on Computer Communications (INFOCOM), pp. 1-6. IEEE press, San Diego (2010)
22. Chen, I.R., Bao, F., Chang, M.J., Cho, J.H.: Trust Management for Encounter-based Routing in Delay Tolerant Networks. In: IEEE Global Communications Conference (GLOBECOM). IEEE press, Miami (2010)
23. Vahdat, A., Becker, D: Epidemic Routing for Partially Connected Ad Hoc Networks. In: Technical Report, Computer Science Department, Duke University (2000)
24. Li, F., Wu, J., Srinivasan, A.: Thwarting Black Hole Attacks in Disruption-tolerant Networks using Encounter Tickets. In: 28th Conference on Computer Communications (INFOCOM), pp. 2428-2436. IEEE press, Rio De Janeiro (2009)
25. Ren, Y., Chuah, M., Yang, J., Chen, Y.: Muton: Detecting Malicious Nodes in Disruption-tolerant Networks. In: IEEE Wireless Communications and Networking conference (WCNC), pp.1-6. IEEE press, Sydney (2010)
26. Liu, K., Abu-Ghazaleh, N., Kang, D.: Location Verification and Trust Management for Resilient Geographic Routing. Journal of Parallel and Distributed Computing, vol. 67, no. 2, pp. 215-228. Elsevier press (2007)
27. Giardo, G., Fricks, R.M., Mupplala, J.K., Trivedi, K.S.: Stochastic Petri Net Package Users Manual. Department of Electrical Engineering, Duke University (1999)