Trust-Threshold Based Routing in Delay Tolerant Networks

Moonjeong Chang, Ing-Ray Chen, Fenye Bao, Jin-Hee Cho

Trust-Threshold Based Routing in 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, encounter-based routing, trust management, threshold-based routing, performance analysis.

1 Introduction

Delay 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]. Different from traditional networks such as mobile ad hoc networks, nodes in DTNs forward messages to a destination node in a *store-and-forward* manner [1] in order to cope with the absence of guaranteed end-to-end connectivity. In such environments, the key challenge is to select an appropriate "next message carrier" among all encountered nodes to maximize message delivery ratio while minimizing message overhead and delay. Further, we face additional challenges due to the lack of a centralized trust entity. The open, distributed, and dynamic inherent 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 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, 5]. 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. Several recent studies [6-9] used reputation to select message carriers among encountered nodes and encouraged cooperative behaviors using credit incentives. However, a centralized credit management system which can be a single point of failure is typically required, as it is challenging to perform distributed credit management in a DTN in the presence of selfish or malicious nodes. On the other hand, there have been several social network based approaches [10-14] to select the best message carrier in DTNs. They considered social relationship and social networking as criteria to select message carriers in DTNs. However, no consideration was given to the presence of malicious or selfish nodes.

This work extends from our earlier work [15] on trust-based routing in DTNs. Unlike prior work cited above [4-14], we integrate *social trust* and *Quality of Service (QoS) trust* into a composite trust metric for determining the best message carrier among 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 (TTBR) in DTNs would perform the best. Our approach is distributed in nature and does not require a complicated credit management system. Each node will run TTBR autonomously to assess trust of its peers using the same trust threshold setting depending on application characteristics, and consequently select 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 comparative analysis of TTBR with epidemic routing [16], social-trust-based routing (for which only social trust metrics are considered) 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 which trust-threshold based routing outperforms these baseline routing algorithms for a DTN consisting of heterogeneous mobile nodes.

2 System Model

We consider a DTN environment without a centralized trust authority. 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. Once a node becomes uncooperative, it stays as uncooperative. A malicious node acts maliciously with the intent to disrupt the main functionality of the DTN, so it can drop packets, jam 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 excluding 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 stays as a malicious node.

We consider the following energy model. The energy level of a node is related to its social encountering activities. If a node becomes uncooperative, the speed of energy consumption by the node is slowed down. If a node becomes compromised, the speed of energy consumption by the node will increase since the node may perform attacks which may consume more energy.

A node's trust value is assessed based on direct observations through monitoring, snooping, or overhearing, and indirect information. To counter whitewashing or false information attacks, a node does not use status exchange information including encounter history information because a malicious node can provide fake encounter history information to other nodes [17]. For indirect information, a node uses 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 aspects of trust relationship: *social trust* and *QoS trust*. *Social trust* is based on social relationships. We consider ***healthiness*** 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: recommender threshold denoted by T_{rec} and message forwarding threshold denoted by T_f. In this paper, the trust value of node j 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 as follows:

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

Here $w_1, w_2, w_3,$ and w_4 are weights associated with healthiness, cooperativeness, connectivity and energy, respectively with $w_1 + w_2 + w_3 + w_4 = 1$. 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. In Eq. 2, β_1 is a parameter to weigh node *i*'s own trust assessment toward node *j* at time $t+\Delta t$, and β_2 is another parameter to weigh indirect information from the recommender. Note that $\beta_1 + \beta_2 = 1$.

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

The direct trust evaluation of node *j* is given in Eq. 3 above by 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 direct observations toward node *m* over the encounter interval $[t, t+\Delta t]$. Node *i* may also leverage its past experiences with node *m* over $[0, t]$ to help assess $T_{i,m}^{encounter,X}(t+\Delta t)$, especially if the current encountering interval is short. If node *j* is not the new encounter, 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 trust decay over time. We adopt an exponential time decay factor, $e^{-\lambda_d \Delta t}$ (with $0 < \lambda_d \leq 0.1$ to limit the decay to at most 50%). Below we describe how node *i* can assess $T_{i,m}^{encounter,X}(t+\Delta t)$ based on direct observations during its encounter with node *m* over the interval $[t, t+\Delta t]$:

- $T_{i,m}^{encounter,healthiness}(t+\Delta t)$: This provides the belief of node *i* that node *m* is not compromised based on node *i*'s direct observations toward node *m* over the encounter interval $[t, t+\Delta t]$. Node *i* can monitor node *m*'s unhealthiness evidences including dishonest trust recommendation, irregular packet patterns, and abnormal traffic over the new encounter period $[t, t+\Delta t]$ or even extend the time period to $[0, t+\Delta t]$ to help assess $T_{i,m}^{encounter,healthiness}(t+\Delta t)$. It can be computed by the number of bad experiences in healthiness over the total healthiness experiences.

- $T_{i,m}^{encounter,cooperativeness}(t+\Delta t)$: This provides the degree of node *m*'s cooperativeness evaluated by node *i* based on direct observations over the encounter interval $[t, t+\Delta t]$. Node *i* can apply overhearing and snooping techniques to detect cooperativeness behavior, e.g., whether or not node *m* follows the prescribed hello or routing protocol, over the time period $[t, t+\Delta t]$ or even extend the time period to $[0, t+\Delta t]$. It can be computed by the number of bad experiences in cooperativeness over the total cooperativeness experiences.

- $T_{i,m}^{encounter,connectivity}(t+\Delta t)$: This provides the connectivity belief that node *m* will encounter node *d* (a node which may become a destination node in packet forwarding in the future). It can be computed by the number of encounters between node *m* and node *d* over 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 $T_{i,m}^{encounter,connectivity}(t+\Delta t)$ for the connectivity of node m to node d. In particular, if node i observes that node d encounters node m most frequently among all nodes over the time period $[0, t+\Delta t]$, then $T_{i,m}^{encounter,connectivity}(t+\Delta t) = 1$. This means that node i highly trusts that node m will encounter node d often and is a good candidate for packet forwarding.

- $T_{i,m}^{encounter,energy}(t+\Delta t)$: This provides the belief of node i toward node m's energy status based on direct observations toward node m. Here energy represents competence. Node i can monitor node m's transmission signal strength over $[t, t+\Delta t]$ to estimate energy status of node m.

On the other hand, for indirect trust evaluation, only 1-hop neighbors of node i will be used as recommenders for scalability. 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 at time t.

The indirect trust evaluation toward node j is given in Eq. 4 below. R_i is the 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, then there is no indirect recommendation available for node j, so node i will use its past trust toward node j obtained at time t with trust decay over Δt. If the new encounter is not node j and node i considers node c as a trustworthy recommender, 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 node c's recommendation with node i's trust toward node c. Moreover, the more recommendations from trustworthy nodes node i receives, the more accurate the trust value of node j can be. Using T_{rec} provides robustness against bad-mouthing or good-mouthing attacks since only recommendations from trustworthy nodes are considered.

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

When node i encounters node m, it can use $T_{i,m}(t)$ to decide whether or not node m can be the next message carrier to shorten message delay or improve message delivery ratio. We consider a Ω–permissible policy with T_f as the minimum trust threshold for the selection of the next message carrier. That is, node i will forward 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. This guarantees to select a trustworthy next message carrier. We consider only single-copy message routing, and buffer management is not considered in this paper.

4 Performance Model

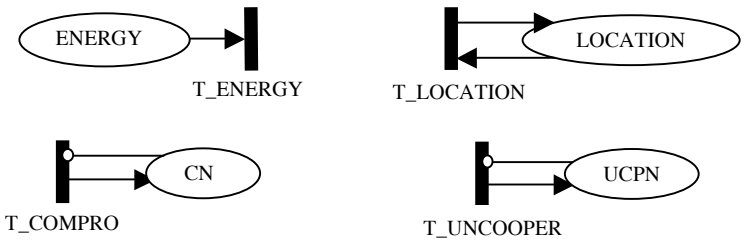


Fig. 1. SPN Model.

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 [18] due to its ability to handle a large number of states. The SPN model is shown in Fig. 1 consisting of 4 event subnets, namely, in clockwise order, energy, location, cooperativeness, 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 its trust relationships 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 radio range R. Initially nodes are randomly distributed over the operational area based on uniform distribution. Below we explain how we construct the SPN model for describing a node's ground truth status.

***Location (Connectivity)*:** 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 (i.e., north, west, south, and east) from its current location with the rate σ_0/R based on the node's speed σ_0 and radio range R. To avoid end-effects, movement is bounced back. This information along with the location information of other nodes at time t provides us the probability of two nodes encountering with each other at any time t.

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).

***Healthiness*:** We use the *compromise 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.

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 facilitates the calculation of $T_{i,j}^{X}(t+\Delta t)$ in theoretical analysis as follows. When node *i* encounters node *j*, node *i* will assess node *j* in trust property *X* to yield $T_{i,j}^{encounter,X}(t+\Delta t)$. Node *i* can directly observe node *j* during the current encounter interval $[t, t+\Delta t]$ plus it may have accumulated past direct observations toward node *j* over $[0,\ t]$ prior to the current encounter. Thus, assuming that the "cooperativeness detection mechanism" described earlier in the protocol design is effective, node *i*'s direct assessment on node *j*'s cooperativeness will be close to the ground truth cooperativeness status of node *j* at time $t+\Delta t$. Consequently, $T_{i,j}^{encounter,cooperativeness}(t+\Delta t)$ in Eq. 3 can be estimated by the probability that place *UCPN* in node *j* does not contain a token at time $t+\Delta t$. Similarly, node *i* can fairly accurately assess $T_{i,j}^{encounter,connectivity}(t)$ by consulting its encounter history with all nodes over the interval $[0, t+\Delta t]$. This quantity can be obtained by utilizing the SPN output regarding the location probability of nodes *j* and *d* at time $t+\Delta t$. For the healthiness trust component, assuming that the "healthiness detection mechanism" in the protocol design is effective, $T_{i,j}^{encounter,healthiness}(t+\Delta t)$ can be approximated by the probability that place *CN* in node *j* does not contain any token at time $t+\Delta t$. Lastly, node *i* can observe node *j*'s packet transmission signal strength over $[t, t+\Delta t]$ to estimate $T_{ij}^{encounter,energy}(t+\Delta t)$, which will be close to the ground truth energy status of node *j* and can be obtained from the SPN output by inspecting place *ENERGY*. Note that we predict $T_{i,j}^{encounter,X}(t+\Delta t)$ for theoretical analysis. In practice, node *i* would follow the protocol design to assess $T_{i,j}^{encounter,X}(t+\Delta t)$. Once $T_{i,j}^{encounter,X}(t+\Delta t)$ is obtained, node *i* can update its $T_{i,j}^{X}(t+\Delta t)$ based on Eq. 2, and subsequently, obtain $T_{i,j}(t+\Delta t)$ based on Eq. 1.

5 Results

In this section, we show numerical results and provide physical interpretation of the results obtained. For trust-threshold based routing (TTBR), we set $w_1 : w_2 : w_3 : w_4 = 0.25 : 0.25 : 0.25 : 0.25$. We setup 20 nodes with vastly different initial energy levels (in the range of [12, 24] hours) in the system. Each node moves randomly in an 8×8 operational area with mobility rate being σ_0 in the range of [1, 4] m/sec. Each of the 8×8 square regions is of the same size, with each side equal to R = 250 m. There are three types of nodes, namely, good, uncooperative and malicious nodes. A bad node is either uncooperative or malicious, or both. Good nodes have zero compromise and uncooperative rates. Uncooperative nodes have a non-zero uncooperative rate $\lambda_{uncooper}$ (i.e., once per 300 sec). Malicious nodes have a non-zero compromise rate λ_{com} in the

range of [1/480min., 1/160min.]. We set β_1:β_2=0.8:0.2 to put high trust on direct observations over indirect recommendations. The initial trust level is set to ignorance (i.e., 0.5) for all trust components due to no prior interactions among nodes. We set the decay coefficient $\lambda_d = 0.001$, and the average encounter interval Δt =5 min, resulting in $e^{-\lambda_d \Delta t} = 0.995$ to model small trust decay over time.

We consider a message forwarding case that a pair of source and destination nodes is picked randomly among good nodes in each run. We allow 30 min warm-up time for nodes to accumulate experiences about each other and start a message forwarding afterwards in each run. If a message carrier is malicious, the message is dropped (a weak attack). If the message carrier is uncooperative, the message delivery continues with 50% chance. The 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. Here, the message overhead is measured by the number of copies forwarded to reach the destination node. For the message delay and the message overhead, we only consider messages that are delivered successfully.

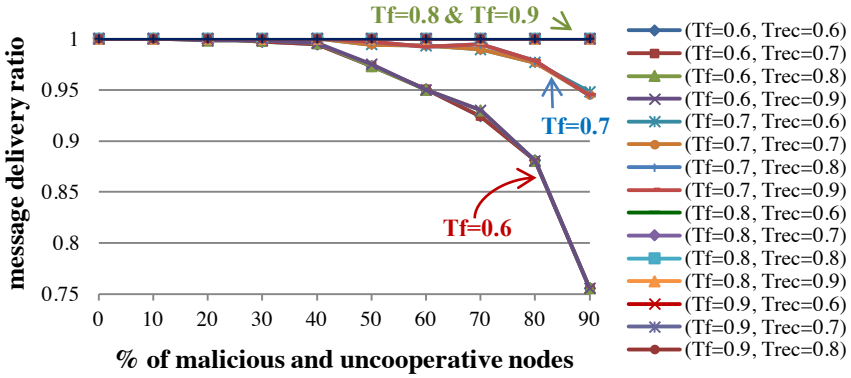
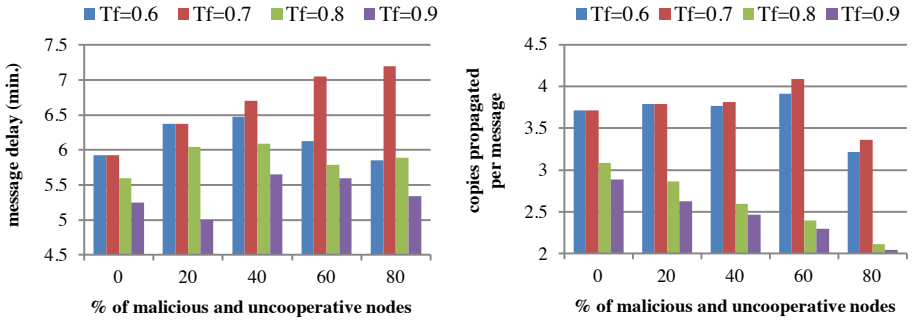


Fig. 2. Effect of T_f and T_{rec} on Message Delivery Ratio.

First of all, we investigate the optimal values of T_{rec} and T_f under TTBR in DTNs. From Figs. 2-3, we see that $T_f = 0.9$ consistently performs better than the others in terms of all performance metrics over a wide range of bad node population. This is because with $T_f = 0.9$, TTBR behaves like a "direct delivery" approach with very little copies being passed around to intermediate message carriers, resulting in a more direct route to the destination node. More specifically, as the percentage of bad nodes increases, there may be an extreme case where node i stores a message and delivers it directly to the destination node because it could not encounter any node with trust higher than T_f. This is true in our DTN scenario where nodes can encounter each other with nonzero probability due to random movement. In situations where a node's 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. From Fig. 4, we see that $T_{rec} = 0.6$ has the shortest message delay and the lowest message

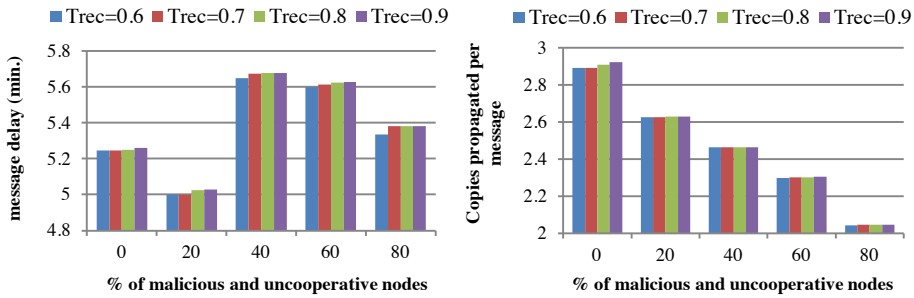
overhead over a wide range of the percentage of bad nodes when T_f is fixed at 0.9. The reason is that the recommenders are all good nodes when $0.6 \leq T_{rec} \leq 0.9$ and $T_{rec} = 0.6$ not only allows more recommenders but also provides sufficiently correct recommendations, resulting in a more accurate indirect trust assessment based on Eq. 4.

In summary, we conclude that there can exist optimal T_f and T_{rec}in TTBR to best tradeoff message delivery ratio, message delay, and message overhead, adapting to application or network environmental conditions.



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

Fig. 3. Effect of T_f on Message Delay and Message Overhead ($T_{rec} = 0.6$).



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

Fig. 4. Effect of T_{rec} on Message Delay and Message Overhead ($T_f = 0.9$).

We also perform a comparative analysis of TTBR against epidemic routing, social-trust-based routing (STBR), and QoS-trust-based routing (QTBR). For STBR and QTBR, we set $w_1 : w_2 : w_3 : w_4 = 0.5 : 0.5 : 0 : 0$ and $0 : 0 : 0.5 : 0.5$, respectively. Note that STBR and QTBR are special cases of TTBR, with STBR using only social trust metrics and QTBR using only QoS trust metrics 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.

Fig. 5 shows that the routing protocols with T_f outperform those without T_f in the delivery ratio. Also, TTBR with T_f and STBR with T_f perform better than QTBR with T_f and epidemic routing with delivery ratio approaching 1 over a wide range of bad node population. This is because TTBR and STBR are able to differentiate

trustworthy nodes from bad 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.

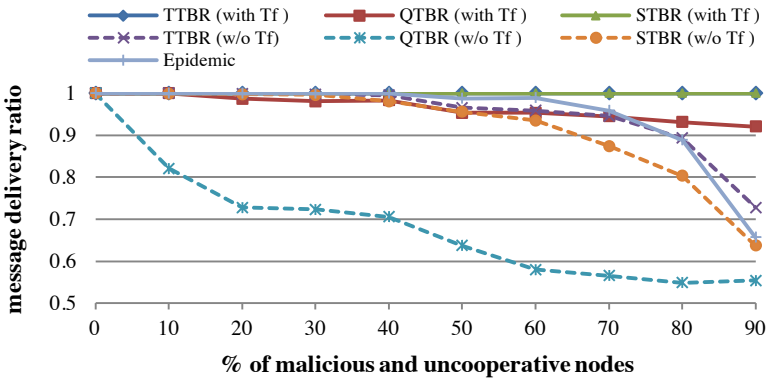


Fig. 5. Message Delivery Ratio $(T_{rec} = 0.6,\ \ T_f = 0.9)$.

Fig. 6 shows that all routing algorithms without T_f approach the ideal performance obtainable from epidemisc routing as the percentage of bad 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. Fig. 7 shows 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.

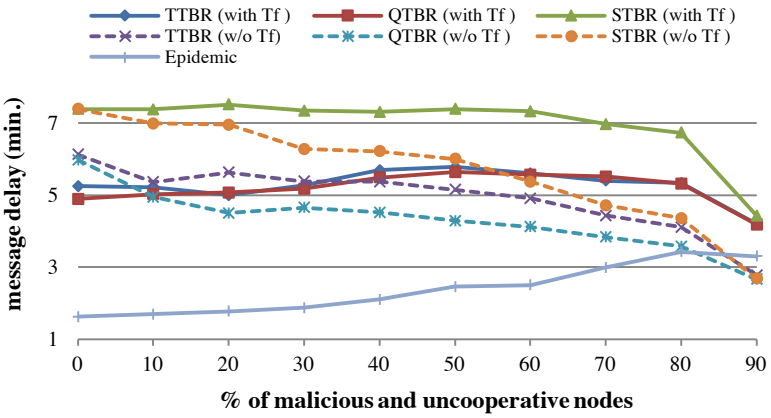


Fig. 6. Message Delay $(T_{rec} = 0.6,\ \ T_f = 0.9)$.

In Figs. 6-7, QTBR performs better than TTBR and STBR in terms of message delay and message overhead. This is because the path selected by TTBR or STBR may not be the most direct route as they attempt to avoid bad nodes when compared

with QTBR that only uses the connectivity metric and the residual energy metric as the criteria to select a message carrier. This result indicates that if the objective is to minimize message delay or message overhead, we should set the weights associated with connectivity and energy considerably higher than those for healthiness and cooperativeness for TTBR to approach the performance of QTBR in message delay or message overhead.

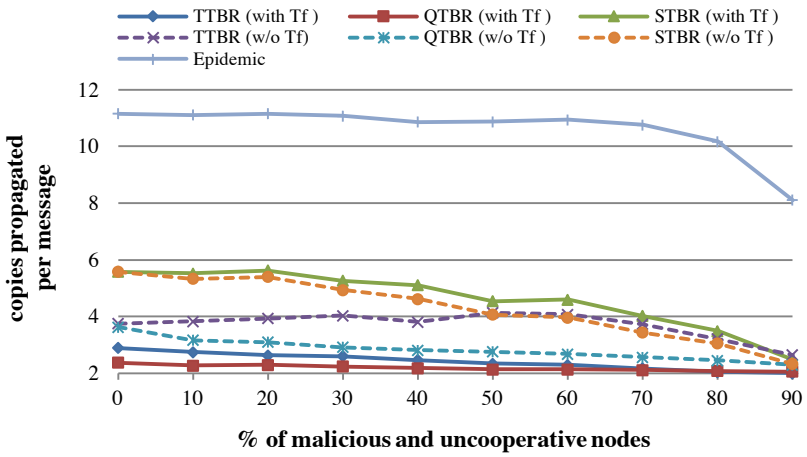


Fig. 7. Number of Copies Propagated per Message ($T_{rec} = 0.6,\ T_f = 0.9$).

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 and message delay for a significant gain in message delivery ratio. 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., STBR vs. QTBR vs. TTBR) to best balance the effect of social trust metrics vs. QoS trust metrics to maximize the delivery ratio without compromising message delay and/or message overhead requirements.

6 Conclusion

We have proposed and analyzed a trust-threshold based routing algorithm with the design objective to maximize the message delivery ratio while satisfying the 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, TTBR 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 perform a more comprehensive comparative analysis with existing trust management

protocols for DTN routing. We also plan to address quality assurance of subjective trust evaluation by extensive theoretical and experimental validation with trace data.

Acknowledgments. This work was supported by the National Research Foundation of Korea Grant funded by the Korean Government [NRF-2009-352-D00262] to Dr. Chang.

References

1. Fall, K., Farrell, S.: DTN: An Architectural Retrospective. Journal on Selected Areas in Communications, vol. 26, no. 5, pp. 828-836. IEEE press (2008)
2. Daly, M., Haahr, M.: The Challenges of Disconnected Delay Tolerant MANETs. Ad Hoc Networks, vol. 8, no. 2, pp. 241-250. Elsevier press (2010)
3. 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)
4. Jain, S., Fall, K., Patra, R.: Routing in a Delay Tolerant Network. Computer Communication Review, vol. 34, no. 4, pp. 145-158. ACM press (2004)
5. Nelson, 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)
6. Shevade, U., Song, H., Qiu, L., Zhang, Y.: Incentive-Aware Routing in DTNs. In: 16th Conference on Network Protocols, pp. 238-247. IEEE press, Orlando (2008)
7. Xu, A., Jin, Y., Shu, W., Liu, X., Luo, J.: SReD: A Secure Reputation-Based Dynamic Window Scheme for Disruption-Tolerant Networks. In: Military Communications, pp. 1-7. IEEE press, Boston (2009)
8. Chen, B., Chan, M.: MobiCent: A Credit-Based Incentive System for Disruption Tolerant Network. In: 29th Conference on Computer Communications, pp. 875-883. IEEE press, San Diego (2010)
9. 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)
10. Hui, P., Crowcroft, J., Yoneki, E.: BUBBLE Rap: Social-based Forwarding in Delay Tolerant Networks. In: MobiHoc, pp. 241-250. ACM press, Hong Kong (2008)
11. Daly, 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)
12. Bulut, E., Wang, Z., Szymanski, K.: Impact of Social Networks on Delay Tolerant Routing. In: Global Communications Conference, pp. 1804-1809. IEEE press, Hawaii (2009)
13. 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, pp. 866-874. IEEE press, San Diego (2010)
14. Mtibaa, A., May, M., Diot, C., Ammar, M.: PeopleRank: Social Opportunistic Forwarding. In: 29th Conference on Computer Communications, pp. 111-115. IEEE press, San Diego (2010)
15. Chen, I.R., Bao, F., Chang, M.J., Cho, J.H.: Trust Management for Encounter-based Routing in Delay Tolerant Networks. In: Global Communications Conference. IEEE press, Miami (2010)
16. Vahdat, A., Becker, D: Epidemic Routing for Partially Connected Ad Hoc Networks. In: Technical Report, Computer Science Department, Duke University (2000)
17. Ren, Y., Chuah, M., Yang, J., Chen, Y.: Muton: Detecting Malicious Nodes in Disruption-tolerant Networks. In: Wireless Communications and Networking conference, pp. 1-6. IEEE press, Sydney (2010)

18. Giardo, G., Fricks, R.M., Mupplala, J.K., Trivedi, K.S.: Stochastic Petri Net Package Users Manual. Department of Electrical Engineering, Duke University (1999)