REPORT DOCUMENTATION PAGE

Form Approved OMB NO. 0704-0188

1. REPORT DATE (DD-MM-YYYY)	2. REPORT TYPE	3. DATES COVERED (From - To)
28-09-2012	Final Report	8-Dec-2011 - 7-Sep-2012

4. TITLE AND SUBTITLE
Dynamic Trust Management for Delay Tolerant Networks and Its Application to Secure Routing

5a. CONTRACT NUMBER
W911NF-12-1-0016

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
622120

6. AUTHORS
Ing-Ray Chen, Fenye Bao, MoonJeong Chang, and Jin-Hee Cho

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES AND ADDRESSES
Virginia Polytechnic Institute & State University
Office of Sponsored Programs
Virginia Polytechnic Institute and State University
Blacksburg, VA 24060 -

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

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

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
61420-NS-II.3

12. DISTRIBUTION AVAILIBILITY STATEMENT
Approved for Public Release; Distribution Unlimited

13. SUPPLEMENTARY NOTES
The views, opinions and/or findings contained in this report are those of the author(s) and should not contrued as an official Department of the Army position, policy or decision, unless so designated by other documentation.

14. ABSTRACT
Delay tolerant networks (DTNs) are characterized by high end-to-end latency, frequent disconnection, and opportunistic communication over unreliable wireless links. In this paper, we design and validate a dynamic trust management protocol for secure routing optimization in DTN environments in the presence of well-behaved, selfish and malicious nodes. We develop a novel model-based methodology based on Stochastic Petri Net (SPN) techniques for the analysis of our trust protocol and validate it via extensive simulation. Moreover, we address

15. SUBJECT TERMS
Delay tolerant networks, dynamic trust management, secure routing, performance analysis, design and validation.

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	15. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Ing-Ray Chen
UU	UU	UU	UU		19b. TELEPHONE NUMBER 703-538-8376

Report Title

Dynamic Trust Management for Delay Tolerant Networks and Its Application to Secure Routing

ABSTRACT

Delay tolerant networks (DTNs) are characterized by high end-to-end latency, frequent disconnection, and opportunistic communication over unreliable wireless links. In this paper, we design and validate a dynamic trust management protocol for secure routing optimization in DTN environments in the presence of well-behaved, selfish and malicious nodes. We develop a novel model-based methodology based on Stochastic Petri Net (SPN) techniques for the analysis of our trust protocol and validate it via extensive simulation. Moreover, we address dynamic trust management, i.e., determining and applying the best operational settings at runtime in response to dynamically changing network conditions to minimize trust bias and to maximize the routing application performance. We perform a comparative analysis of our proposed routing protocol against existing trust-based (SReD) and non-trust based (PROPHET and epidemic) protocols. The results demonstrate that our protocol is able to deal with selfish behaviors and is resilient against trust-related attacks. Furthermore, our trust-based routing protocol can effectively trade off message overhead and message delay for a significant gain in delivery ratio. Our trust-based routing protocol operating under identified best settings outperforms SReD and PROPHET, and approaches the ideal performance of epidemic routing in delivery ratio and message delay without incurring high message or protocol maintenance overhead.

Enter List of papers submitted or published that acknowledge ARO support from the start of the project to the date of this printing. List the papers, including journal references, in the following categories:

(a) Papers published in peer-reviewed journals (N/A for none)

Received	Paper
2012/09/28 0(1	Ing-Ray Chen, Fenye Bao, MoonJeong Chang, Jin-Hee Cho. Integrated Social and QoS Trust-Based Routing in Delay Tolerant Networks, WIRELESS PERSONAL COMMUNICATIONS, (06 2012): 443. doi: 10.1007/s11277-011-0351-2

TOTAL: 1

Number of Papers published in peer-reviewed journals:

(b) Papers published in non-peer-reviewed journals (N/A for none)

Received	Paper

TOTAL:

Number of Papers published in non peer-reviewed journals:

(c) Presentations

Number of Presentations: 0.00

Non Peer-Reviewed Conference Proceeding publications (other than abstracts):

Received	Paper

TOTAL:

Number of Non Peer-Reviewed Conference Proceeding publications (other than abstracts):

Peer-Reviewed Conference Proceeding publications (other than abstracts):

Received	Paper
2012/09/28 0(2	Jin-Hee Cho, MoonJeong Chang, Ing-Ray Chen, Ananthram Swami. A Provenance-based Trust Model for Delay Tolerant Networks, 6th IFIP WG 11.11 International Conference on Trust Management (IFIPTM 2012). 2012/05/21 00:00:00, . : ,

TOTAL: 1

Number of Peer-Reviewed Conference Proceeding publications (other than abstracts):

(d) Manuscripts

Received	Paper

TOTAL:

Number of Manuscripts:

Books

Received	Paper

TOTAL:

Patents Submitted

Patents Awarded

Awards

Graduate Students

NAME	PERCENT_SUPPORTED
FTE Equivalent:	
Total Number:	

Names of Post Doctorates

NAME	PERCENT_SUPPORTED
MoonJeong Chang	1.00
FTE Equivalent:	**1.00**
Total Number:	**1**

Names of Faculty Supported

NAME	PERCENT_SUPPORTED

FTE Equivalent:
Total Number:

Names of Under Graduate students supported

NAME	PERCENT_SUPPORTED

FTE Equivalent:
Total Number:

Student Metrics

This section only applies to graduating undergraduates supported by this agreement in this reporting period

The number of undergraduates funded by this agreement who graduated during this period:	0.00
The number of undergraduates funded by this agreement who graduated during this period with a degree in science, mathematics, engineering, or technology fields:	0.00
The number of undergraduates funded by your agreement who graduated during this period and will continue to pursue a graduate or Ph.D. degree in science, mathematics, engineering, or technology fields:	0.00
Number of graduating undergraduates who achieved a 3.5 GPA to 4.0 (4.0 max scale):	0.00
Number of graduating undergraduates funded by a DoD funded Center of Excellence grant for Education, Research and Engineering:	0.00
The number of undergraduates funded by your agreement who graduated during this period and intend to work for the Department of Defense	0.00
The number of undergraduates funded by your agreement who graduated during this period and will receive scholarships or fellowships for further studies in science, mathematics, engineering or technology fields:	0.00

Names of Personnel receiving masters degrees

NAME

Total Number:

Names of personnel receiving PHDs

NAME

Total Number:

Names of other research staff

NAME	PERCENT_SUPPORTED

FTE Equivalent:
Total Number:

Sub Contractors (DD882)

Inventions (DD882)

Scientific Progress

Technology Transfer

Dynamic Trust Management for Delay Tolerant Networks and Its Application to Secure Routing

Ing-Ray Chen, Fenye Bao, MoonJeong Chang, and Jin-Hee Cho

Abstract— Delay tolerant networks (DTNs) are characterized by high end-to-end latency, frequent disconnection, and opportunistic communication over unreliable wireless links. In this paper, we design and validate a dynamic trust management protocol for secure routing optimization in DTN environments in the presence of well-behaved, selfish and malicious nodes. We develop a novel model-based methodology based on Stochastic Petri Net (SPN) techniques for the analysis of our trust protocol and validate it via extensive simulation. Moreover, we address dynamic trust management, i.e., determining and applying the best operational settings at runtime in response to dynamically changing network conditions to minimize trust bias and to maximize the routing application performance. We perform a comparative analysis of our proposed routing protocol against existing trust-based (SReD) and non-trust based (PROPHET and epidemic) protocols. The results demonstrate that our protocol is able to deal with selfish behaviors and is resilient against trust-related attacks. Furthermore, our trust-based routing protocol can effectively trade off message overhead and message delay for a significant gain in delivery ratio. Our trust-based routing protocol operating under identified best settings outperforms SReD and PROPHET, and approaches the ideal performance of epidemic routing in delivery ratio and message delay without incurring high message or protocol maintenance overhead.

Index Terms— Delay tolerant networks, dynamic trust management, secure routing, performance analysis, design and validation.

1 Introduction

A delay tolerant network (DTN) comprises mobile nodes experiencing sparse connection, opportunistic communication, and frequently changing network topology. Because of lack of end-to-end connectivity, routing in DTN adopts a *store-carry-and-forward* scheme by which messages are forwarded through a number of intermediate nodes leveraging opportunistic encountering, hence resulting in high end-to-end latency. Traditional routing protocols [6, 15, 23] for DTNs focus on exploring the mobility pattern and predicting future encounter events. However, in the presence of malicious and selfish nodes, these routing protocols still can have a low message delivery ratio. Recently, social relationship and social network [5, 10, 14] have been used for message routing in DTNs to improve routing performance, especially for DTNs with human involved, such as devices installed on vehicles, and smartphones and handheld tablets operated by humans. No consideration was given to malicious nodes, however. In this paper, we propose a trust management protocol and apply it for secure routing optimization for DTNs in the presence of malicious and selfish nodes.

- *Ing-Ray Chen, Fenye Bao and MoonJeong Chang are with the Department of Computer Science, Virginia Tech, Falls Church, VA 22043. E-mail: (baofenye, irchen, mjchang)@vt.edu.*
- *Jin-Hee Cho is with Computational and Information Sciences Directorate, U.S. Army Research Laboratory, Powder Mill Rd. Adelphi, MD 20783. E-mail: jinhee.cho@us.army.mil.*

Most existing trust management protocols in the literature consider only quality of service (QoS) metrics for trust evaluation and do not take into account the social relationship which is an important factor for trust management in DTNs with the proliferation of mobile devices carried and operated by humans. Furthermore, existing trust management protocols do not consider the issue of dynamic trust management, i.e., how can a trust protocol best respond to changing DTN environment conditions such as an increasing population of misbehaving nodes or evolving hostility or social relations such that an application (e.g., secure routing) built on top of trust management can maximize its performance.

In this paper, we *design* and *validate* a dynamic trust management protocol for secure routing performance optimization in DTNs in response to dynamically changing conditions such as increasing population of misbehaving nodes. The design part addresses the three core functions of trust management, namely, *trust composition*, *trust aggregation*, and *trust formation*. For trust composition, we combine social trust deriving from social networks and QoS trust deriving from communication networks into a composite trust metric to assess the trust of a node in DTNs. For trust aggregation, we consider both direct observations and indirect recommendations to update trust. We advocate separation of concerns for each social trust or QoS trust property selected. Our trust aggregation protocol (described below in Section 4) for aggregating trust

information of a trustee may use a distinct set of parameter settings for each trust property taking into account intrinsic properties of each trust property, so the "subjective" trust evaluation of the trustee node for that trust property is accurate. For trust formation, we investigate and identify the best way to *form* the overall trust out of the selected social and QoS trust properties in order to maximize application performance. Further, we investigate a new design concept of *application-level trust optimization* allowing an application to optimize the use of trust to classify nodes also for maximizing application performance. Our design concept of application-level trust optimization in secure routing lies in the use of a double trust threshold policy for filtering out untrustworthy trust recommenders and message forwarders.

We develop a novel model-based methodology based on Stochastic Petri Net (SPN) techniques [22] for the analysis of our trust protocol and validate it via extensive simulation. By means of a probabilistic model given the anticipated operational profile [17] as input, we describe a large number of nodes, each with its own mobility, social, and QoS behaviors. The model validated with simulation yields actual *ground truth* node status against which "subjective" trust obtained from executing the trust protocol is verified, and helps identify the best protocol settings in response to dynamically changing network conditions to minimize trust bias and to maximize the routing application performance.

We perform a comparative analysis of our proposed routing protocol with simulation validation against existing trust-based (SReD [24]) and non-trust based (PROPHET [15] and epidemic [23]) protocols. The results demonstrate that our protocol is able to deal with selfish behaviors and is resilient against trust-related attacks. In addition, the proposed routing protocol can effectively trade off message overhead and message delay for a significant gain in delivery ratio. Our trust-based routing protocol operating under identified best settings outperforms SReD and PROPHET. Further, it approaches the ideal performance of epidemic routing in delivery ratio and message delay without incurring high message or protocol maintenance overhead.

The rest of the paper is organized as follows. In Section 2, we survey existing trust management protocols and approaches to deal with misbehaved nodes in DTN routing. In Section 3, we describe the system model. In Section 4, we describe our dynamic trust management protocol. In Section 5, we develop a performance model based on SPN techniques [22] for the analysis of our trust protocol. In Section 6, we first identify the best protocol settings to minimize trust bias and to maximize the routing application performance, when given a set of parameters characterizing the operational and environmental conditions. Then we perform a comparative analysis of our proposed routing protocol against existing trust-based (SReD) and non-trust based (PROPHET and epidemic) protocols. In Section 7, we validate our trust management protocol design through extensive simulation using both synthetic and real mobility trace data. In Section 8, we demonstrate the effectiveness of dynamic trust management in response to dynamically changing network conditions to maximize DTN routing performance. Finally in Section 9, we conclude the paper and discuss future research areas.

2 Related Works

Trust management in DTNs is little explored in the literature. To the best of our knowledge, only [2, 3, 11, 13, 24] used feedback mechanisms or indirect recommendations for trust management. In addition, [2, 3] used discrepancies of indirect recommendations for adversary detection. Not leveraging direct-observation based trust/reputation deriving from social networking is a main drawback of these approaches. [24] considered three sources to estimate trust: cryptographic operation, node's behavior, and reputation. For cryptographic operations, encryption and decryption mechanisms are used to provide authentication and confidentiality and to defend outside attackers. A watchdog mechanism is adopted to detect node's behavior, and this information is combined with cryptographic operation using a weighted sum to generate a local trust value. Each node also exchanges its local trust evaluation as recommendation to others. A limitation of their work is that no consideration was given to inside attackers. [2, 3] designed an iterative trust management scheme for DTNs. They used authentication as the underlying mechanism to evaluate a node. A node exchanges its trust evaluation with others and interactively updates its trust evaluation. Inconsistent trust evaluations are identified and removed iteratively until the trust evaluation converges. However, because the iteration process is performed on each node every time trust is updated, the overhead is substantial for mobile networks with a large number of nodes. [4] proposed trust-based secure routing protocols in malicious DTNs. [18] proposed spanning tree algorithms for trust management in DTNs. However, both assumed that trust is already in place. Very recently, [8, 9] considered both direct observations and indirect recommendations for trust management and applied it to encounter-based routing. However, only a theoretical analysis was given without validation. Different from [8, 9] this work is on *design* and *validation* of dynamic trust management for trust-based secure routing in DTNs.

In the literature, *detection* and *prevention* are two widely used approaches to deal with selfish behaviors in DTN routing. Detection-based approaches [8, 9, 24] rely on monitoring and overhearing techniques to identify selfish nodes and avoid selecting selfish nodes as message carriers in DTN routing. The encounter/contact duration of a pair of nodes must last long enough to ensure the detection accuracy. Prevention-based approaches assume that nodes are rational to maximize their own interests and often use

incentives [16, 20, 21, 25] to stimulate cooperation between nodes and avoid selfish behaviors. Well-behaved nodes are awarded while selfish or uncooperative nodes are punished such that a node would not behave selfishly for the sake of its own interest. However, incentive-based approaches normally would not work for malicious nodes with sole interests to disrupt the operation of the system. Context-free protocols [16, 21] have also been proposed to hide the identity of the destination node in order to encourage nodes to participate in packet forwarding. However, in intermittently connected DTN environments, message forwarding follows the store-carry-and-forward paradigm. It is difficult, if not impossible, to establish the entire routing path by the source node without revealing the identity of the destination node to intermediate carriers during DTN routing. In this paper, we propose trust management for DTNs to deal with both malicious and selfish nodes. Our notion of selfishness follows that in the human society [16] where humans (carrying communication devices) are often socially selfish to outsiders but unselfish to friends. Our protocol to deal with selfish or malicious behaviors is detection-based. By using real trace data, we justify that the encounter frequency and contact duration are sufficient to support monitoring and overhearing detection techniques.

3 System Model

We consider a DTN environment with no centralized trusted authority. Nodes communicate through multiple hops. Every node may have a different level of energy and speed reflecting node heterogeneity. When a node encounters another node, they exchange encounter histories certified by encounter tickets [12] so as to prevent black hole attacks to DTN routing. We differentiate selfish nodes from malicious nodes. A selfish node acts for its own interests including interests to its friends, groups, or communities. 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. We consider a friendship matrix [14] to represent the social ties among nodes. Each node keeps a list of friends in its local storage. When a node becomes selfish, it will only forward messages when it is a friend of the source, current carrier, or the destination node, while a well-behaved node performs altruistically regardless of the social ties. Note that the friendship matrix is used to model the social ties and the matrix itself is not a requirement for trust evaluation. Instead, a node uses snooping and overhearing techniques to monitor selfishness of its neighboring nodes. A malicious node aims to break the basic DTN routing functionality. In addition to dropping packets, a malicious node can perform the following trust-related attacks:

1. *Self-promoting attacks*: it can promote its importance (by providing good recommendations for itself) so as to attract packets routing through it (and being dropped).
2. *Bad-mouthing attacks*: it can ruin the reputation of well-behaved nodes (by providing bad recommendations against good nodes) so as to decrease the chance of packets routing through good nodes.
3. *Good-mouthing attacks*; it can boost the reputation of bad nodes (by providing good recommendations for them) so as to increase the chance of packets routing through malicious nodes (and being dropped).

A node's trust value is assessed based on direct observations and indirect information like recommendations. Self-promoting attacks are eliminated in our trust protocol as we do not take in self-recommendations. Bad-mouthing attacks and good-mouthing attacks are mitigated by setting a trust recommender threshold T_{rec}. The trust of one node toward another node is updated upon encounter events. 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 assessing a trust property X. Later in Section 4 we will discuss these specific detection mechanisms employed in our protocol.

4 Trust Management Protocol

Our trust protocol considers trust composition, trust aggregation, trust formation and application-level trust optimization designs. For trust composition, we consider two types of trust properties: QoS trust and social trust. QoS trust is evaluated through the communication network 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. The connectivity QoS trust is about the ability of a node to encounter other nodes due to its movement patterns. The energy QoS trust is about the capability of a node to perform the basic routing function. Social trust is based on honesty or integrity in social relationships and social ties. We consider "healthiness" and social "unselfishness" to measure the social trust level of a node. The healthiness social trust is the belief of whether a node is malicious. The unselfishness social trust is the belief of whether a node is socially selfish. The selection of trust properties is application driven. In DTN routing, message delivery ratio and message delay are two important factors. We select "healthiness", "unselfishness", and "energy" in order to achieve high message delivery ratio, and we select "connectivity" to achieve low message delay.

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. We consider a trust formation model by which 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, unselfishness, connectivity, and energy as follows:

$$T_{i,j}(t) = \sum_{X}^{all} w^X \times T_{i,j}^X(t) \quad (1)$$

where X represents a trust property explored (X = healthiness, unselfishness, connectivity or energy), $T_{i,j}^X(t)$ is node i's trust in trust property X toward node j, and w^X is the weight associated with trust property X with the sum equal to 1.

In this paper, we aim to identify the best weight ratio under which the application performance (secure routing) is maximized, given an operational profile [17] as input. Before this can be achieved, however, one must address the accuracy issue of trust aggregation. That is, for each QoS or social trust property X, we must devise and validate a trust aggregation protocol executed by a trustor node to assess X of a trustee node such that the trust value computed is accurate with respect to actual status of the trustee node in X. This is achieved by devising a trust propagation protocol with tunable parameters which can be adjusted based on each trust property. Specifically, our trust aggregation protocol for node i (trustor) to assess its trust toward node j (trustee) in X upon encountering node m (encounter) at time t over an encounter interval $[t, t + \Delta t]$ is as follows:

$$T_{i,j}^X(t + \Delta t) = \beta T_{i,j}^{direct,X}(t + \Delta t) + (1 - \beta) T_{i,j}^{indirect,X}(t + \Delta t) \quad (2)$$

In Equation 2, β is a parameter to weigh node i's own trust assessment toward node j at time $t + \Delta t$ vs. indirect information from the recommender (the newly encountered node). Every trust property X has its own specific β value under which subjective $T_{i,j}^X(t)$ obtained is accurate, i.e., close to actual status of node j in X at time t. In Equation 2, $T_{i,j}^{direct,X}(t + \Delta t)$ is node i's trust in X toward node j at time $t + \Delta t$ due to "direct" observations or interaction experiences calculated by:

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

If the new encounter (node m) is node j itself, i.e., m=j, then node i can directly evaluate node j because nodes i and 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%). Node i assesses $T_{i,m}^{encounter,X}(t + \Delta t)$ based on direct observations during its encounter with node m over the interval $[t, t + \Delta t]$ as follows:

- $T_{i,m}^{encounter,healthiness}(t + \Delta t)$: Node i assesses node m's unhealthiness based on evidences manifested due to malicious attacks including self-promoting, bad-mouthing and good-mouthing attacks. Evidences of self-promoting attacks may be detected through the encounter history exchanged from node m. If the encounter history is not certified, or is certified but inconsistent with node i's encounter history matrix accumulated, it is considered as a negative experience. Evidences of bad-mouthing/good-mouthing attacks may be detected by comparing node m's recommendation toward node j with the trust value of node i toward node j itself. If the percentage difference is higher than a threshold, it is considered suspicious and thus a negative experience. These positive/negative experiences can be collected over the new encounter period $[t, t + \Delta t]$ or even over $[0, t + \Delta t]$ to assess $T_{i,m}^{encounter,healthiness}(t + \Delta t)$. It is computed by the number of positive experiences in healthiness over the total experiences in healthiness.
- $T_{i,m}^{encounter,unselfishness}(t + \Delta t)$: Node i applies overhearing and snooping techniques to detect socially selfish behaviors, e.g., whether or not node m follows the prescribed routing protocol, over $[t, t + \Delta t]$ or even extend the time period to $[0, t + \Delta t]$. This includes the behavior for executing beacon, routing, and trust protocols expected out of node m. It is computed by the number of positive experiences in unselfishness over the total experiences in unselfishness.
- $T_{i,m}^{encounter,connectivity}(t + \Delta t)$: This trust property represents the connectivity of node m to the destination node d. If the connectivity trust is high, then node m would be a good candidate for packet delivery to node d. Node i deduces node m's connectivity with node d based on its encounter matrix collected over $[0, t + \Delta t]$, including the encounter history received from m. Note that node i will only accept certified encounter history (as in [12]) to avoid black hole attacks.
- $T_{i,m}^{encounter,energy}(t + \Delta t)$: This trust property represents the capability or competence of node m to do the basic routing function. Node i monitors node m's transmission signal strength over $[t, t + \Delta t]$ and extrapolates the amount of energy left in node m.

The above detection mechanisms are for direct trust evaluation when node i encounters node m. On the other hand, for indirect trust evaluation, node i uses its 1-hop neighbors as recommenders for scalability. An application-level optimization parameter is the recommender trust threshold T_{rec} such that if $T_{i,j}(t) > T_{rec}$, node i considers node j as a more trustworthy recommender at time t. Using T_{rec} provides robustness against bad-mouthing or good-mouthing attacks since only recommendations from

more trustworthy nodes are considered. The indirect trust evaluation toward node j is given in Equation 4 below where 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 is allowed to provide its recommendation to node i for evaluating node j. In this case, node i weighs node c's recommendation, $T^X_{c,j}(t+\Delta t)$, with node i's referral trust, $T^X_{i,c}(t+\Delta t)$, toward node c.

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

When node i encounters node m, it uses $T_{i,m}(t)$ from Equation 1 to decide whether or not node m can be the next message carrier to shorten message delay or improve message delivery ratio. Another application-level optimization parameter is the minimum trust threshold T_f for the selection of the next message carrier. Node i will forward the message to node m only if $T_{i,m}(t+\Delta t) \geq T_f$ and $T_{i,m}(t)$ is in the top Ω percentile among all $T_{i,j}(t)$'s. This helps the chance of selecting a trustworthy next message carrier.

5 Performance Model

We validate our trust management designs by a novel model-based analysis methodology via extensive simulation. Specifically we develop a mathematical model based on continuous-time semi-Markov stochastic processes (for which the event time may follow any general distribution) to define a DTN consisting of a large number of mobile nodes exhibiting heterogeneous social and QoS behaviors.

We take the concept of "operational profiles" in software reliability engineering [17] as we build the mathematical model. An operational profile is what the system expects to see during its operational phase. During the testing and debugging phase, a system would be tested with its anticipated operational profile to reveal design faults. Failures are detected and design faults causing system failures are removed to improve the system reliability. The operational profile of a DTN system specifies the operational and environment conditions. Typically this would include knowledge regarding (a) hostility such as the expected % of misbehaved nodes and if it is evolving the expected rate at which nodes become malicious or selfish or even the expected % of misbehaved nodes as a function of time; (b) mobility traces providing information of how often nodes meet and interact with each other; (c) behavior specifications defining good behavior and misbehavior during protocol execution; and (d) resource information such as how fast energy is consumed.

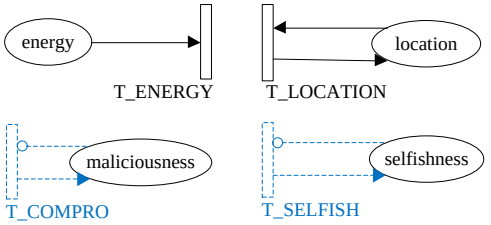


Figure 1: SPN Model for a Node in the DTN.

We develop a probability model based on Stochastic Petri Net (SPN) techniques [22] to describe a DTN, given an operational profile as input. The SPN model for a DTN node is shown in Figure 1 consisting of 4 places, namely, *energy*, *location*, *maliciousness* and *selfishness*. The underlying state machine is a semi-Markov model with 4-component states, i.e., (*energy*, *location*, *maliciousness*, *selfishness*) where energy is an integer holding the amount of energy left in the node, location is an integer holding the location of the node, maliciousness is a binary variable with 1 indicating the node is malicious and 0 otherwise, and selfishness is a binary variable with 1 indicating the node is socially selfish and 0 otherwise. A selfish node will forward a packet only if the source, current carrier or the destination is in its friend list. Each node has its own SPN model. So there are as many SPN models as they are nodes in the DTN. The operational profile specifies the % of malicious nodes and the % of social selfish nodes. Thus, some nodes will be malicious in accordance with this specification. Similarly some nodes will be selfish based on the % of selfish nodes.

The purpose of the SPN model is to yield *ground truth* status of a node in terms of its healthiness, unselfishness, connectivity, and energy status. Then we can check *subjective trust* against *ground truth* status for validation of trust protocol designs. Below we explain how we leverage the SPN model to determine 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 new area from its current location with the mobility rate σ_0/R based on the node's speed σ_0 and radio range R according to its mobility pattern (e.g., random or traces). 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 (E_0) of each node represented by a number of tokens is assigned according to node heterogeneity information. A token is taken out when transition T_ENERGY fires representing the energy consumed during protocol execution, packet forwarding and/or performing attacks in the case of a

malicious node. The rate of transition T_ENERGY indicates the energy consumption rate which varies depending on the ground truth status of the node (i.e., malicious or selfish). The operational profile specifies the energy consumption rate of a malicious node vs. a selfish node vs. a well-behaved node.

Healthiness: A malicious node is necessarily unhealthy. So we will know the ground truth status of healthiness of the node by simply inspecting if place *maliciousness* contains a token.

Unselfishness: A socially selfish node drops packets unless the source, current carrier or the destination node is in its friend list. We will know the ground truth status of unselfishness of the node by simply inspecting if place *selfishness* contains a token.

Dynamically Changing Environment Conditions: With the goal to deal with malicious and selfish nodes in DTN routing, in this paper we consider a dynamically changing environment in which the number of misbehaved nodes (malicious or selfish) is changing over time. A node becomes malicious when it is captured and turned into a compromised node. We model the capture event by a transition T_COMPRO (in dashed line) in Figure 1. Once the transition T_COMPRO is triggered, a token will be moved into the place *maliciousness* representing that this node is compromised. Similarly, once the transition T_SELFISH (also in dashed line) is triggered, a token will be moved into the place *selfishness* representing that this node becomes selfish. The transition rates of T_COMPRO and T_SELFISH are λ_c and λ_s, respectively. We will use the SPN model augmented with the two dashed line transitions in Section 8 in which we treat the subject of dynamic trust management.

Subjective Trust Evaluation: The SPN model described above yields actual or ground truth status of each node, which facilitates the calculation of $T_{i,j}^{encounter,X}(t + \Delta t)$ as follows. For X = connectivity, by making use of the *certified* encounter histories collected, node i can accurately estimate the probability that nodes j and d would encounter each other at $t + \Delta t$. Thus, $T_{i,j}^{encounter,connectivity}(t + \Delta t)$ assessed by node i would be very close to the joint probability of nodes j and d being in the same location at time $t + \Delta t$, which is obtainable from the ground truth "location" status of nodes j and d at time $t + \Delta t$. For X = energy, 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 obtainable by inspecting place *energy* in node j. For X = unselfishness, assuming that the "unselfishness detection mechanism" described earlier in the protocol design is effective, node i's direct assessment on node j's unselfishness would be close to the ground truth "unselfishness" status of node j at time $t + \Delta t$. Consequently, the value of $T_{i,j}^{encounter,unselfishness}(t + \Delta t)$ is 1 if place *selfishness* in node j does not contain a token at time $t + \Delta t$, and 0 otherwise. Lastly, for X = healthiness, assuming that the "healthiness detection mechanism" in the protocol design is effective, the value of $T_{i,j}^{encounter,healthiness}(t + \Delta t)$ from direct trust evaluation would be close to the ground truth "healthiness" status of node j at time $t + \Delta t$, i.e., it is 1 if place *maliciousness* in node j does not contain any token at time $t + \Delta t$ and 0 otherwise. The assumption that unselfishness/healthiness detection mechanism in the protocol design is effective will be validated through simulation later in this paper. 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 Equation 2, and subsequently, obtain the subjective trust of node j, $T_{i,j}(t + \Delta t)$, based on Equation 1.

Objective Trust Evaluation: The "objective" trust of node j at time t, denoted by $T_j(t)$, is also obtained from Equation 1 except that $T_j^X(t)$ is being used instead of $T_{i,j}^X(t)$. Here $T_j^X(t)$ is simply the actual status of node j in trust property X at time t obtainable from the SPN model for node j. The notion of objective trust evaluation is to validate subjective trust evaluation, that is, subjective trust evaluation is valid if the subjective trust value obtained as a result of executing our dynamic trust management protocol is close to the objective trust value obtained from actual or ground truth status.

6 Numerical Results

Our trust evaluation results have two parts. The first part is about the accuracy of trust aggregation for individual trust properties. The second part is about maximizing application performance through trust formation (by setting the best weights to trust properties) and application-level trust optimization (by setting the best recommender trust threshold $T_{rec,}$ and message carrier trust threshold T_f). Because different trust properties have their own intrinsic trust nature and react differently to trust decay over time, each trust property X has its own best set of (β, λ_d) under which $T_{i,j}^X(t)$ obtained from Equation 2 would be the most accurate, i.e., closest to actual status of node j in trust property X, or $T_j^X(t)$. Recall that a higher β value indicates that subjective trust evaluation relies more on direct observations compared with indirect recommendations provided by the recommenders and that a higher λ_d indicates a higher trust decay rate. Once we ensure the accuracy of each trust property X, we can then address the trust formation issue, i.e., identifying the best way to form the overall trust out of QoS and social trust properties and the best way to set application-level trust parameters such that the application performance (i.e., secure routing) is maximized.

Table 1: System Parameters.

Name	Value	Name	Value	Name	Value
$m \times m$	8×8	R	250m	σ_0	(0, 2] m/s
β	Design parameter	λ_d	Design parameter	E_0	12 *hours*
T_{rec}	Design parameter	T_f	Design parameter	w^X	Design parameter

Table 1 lists a set of parameters and their values (for input parameters) as prescribed by the operational profile of a DTN. We consider N = 20 nodes moving with speed randomly chosen over (0, 2] m/s in an 8×8 operational region, with each region coving R = 250m radio radius. The initial energy of each node is set to 12 hours lifetime. The parameters marked with "design parameter" are the ones whose best settings are to be determined as output, given a set of parameter values as listed in Table 1 characterizing the operational and environmental conditions of a DTN. Here we should note that a social friendship matrix [14] and the percentages of selfish and malicious nodes, although not specified in Table 1, are also given as input, which we will vary in the analysis to test their effects on design parameters. Lastly, the node compromise rate (λ_c) and node selfishness rate (λ_s) for characterizing changing DTN conditions are also not specified in Table 1. We will consider these two parameters and treat the subject of dynamic trust management in Section 8.

6.1 Best Trust Propagation Protocol Settings to Minimize Trust Bias

Here we determine the best (β, λ_d) values that yield subjective trust evaluation closest to objective trust evaluation to minimize trust bias, given a set of parameter values as listed in Table 1 characterizing the operational and environmental conditions. Since there are only two input parameters, we search the best (β, λ_d) through exhaust search, i.e., we compare subjective trust obtained through protocol execution under a given (β, λ_d) with objective trust. The best (β, λ_d) combination is the one that produces the lowest mean square error (MSE).

Table 2: Best (β, λ_d) to Minimize Trust Bias.

% of misbehaved nodes	Connectivity (β, $\lambda_d \times 10^3$)	Energy (β, $\lambda_d \times 10^3$)	Healthiness (β, $\lambda_d \times 10^3$)	Unselfishness (β, $\lambda_d \times 10^3$)
0%	(0.75, 6.0)	(0.75, 3.0)	(0.75, 0.1)	(0.75, 0.1)
10%	(0.76, 6.0)	(0.76, 2.7)	(0.76, 0.1)	(0.75, 0.1)
20%	(0.76, 6.0)	(0.76, 2.5)	(0.76, 0.1)	(0.76, 0.1)
30%	(0.76, 6.0)	(0.76, 2.2)	(0.77, 0.1)	(0.76, 0.1)
40%	(0.77, 6.0)	(0.77, 1.9)	(0.78, 0.1)	(0.77, 0.1)
50%	(0.77, 6.0)	(0.77, 1.7)	(0.78, 0.1)	(0.77, 0.1)
60%	(0.82, 6.0)	(0.82, 1.4)	(0.82, 0.1)	(0.79, 0.1)
70%	(0.83, 6.0)	(0.83, 1.1)	(0.83, 0.1)	(0.81, 0.1)
80%	(0.83, 6.0)	(0.83, 0.8)	(0.84, 0.1)	(0.82, 0.1)
90%	(0.84, 6.0)	(0.84, 0.5)	(0.85, 0.1)	(0.83, 0.1)

In Table 2, we summarize the best (β, λ_d) values for each trust property given the % of misbehaved nodes (specified in the first column with 50% selfish nodes and 50% malicious nodes) for a trustor node (i.e., node i) randomly picked toward a trustee node (i.e., node j) also randomly picked. Each (β, λ_d) entry represents the best combination under which subjective trust $T_{i,j}^X(t)$ obtained as a result of executing our trust aggregation protocol for trust property X (as prescribed by Equation 2) deviates from objective trust for property X (that is, $T_j^X(t)$) by less than 1% MSE. This substantiates our claim that there exists a distinct protocol setting in terms of (β, λ_d) for each trust property X, with X = connectivity, energy, healthiness or unselfishness. Furthermore, the best (β, λ_d) setting changes as the % of misbehaved nodes changes dynamically.

6.2 Best Trust Formation Protocol Settings to Maximize Application Performance

Next we turn our attention to the trust formation issue to optimize application performance. For the secure routing application, two most important performance metrics are message delivery ratio and delay. In many situations, however, excessive long delays are not acceptable to DTN applications. We define an objective function P_d as the percentage of messages that are delivered successfully within an application deadline d which is the maximum delay the application can tolerate. Below we set P_{30} (d = 30 minutes) as the utility function to find the best way to assign the weight w^X to X = healthiness, unselfishness, connectivity or energy. Without loss of generality and for ease of disposition, we assume that the weights assigned to social trust properties, i.e., healthiness and unselfishness, are the same each of $0.5 \times w_{social}$, and the weights assigned to QoS trust properties, i.e., connectivity and energy, are the same each of $0.5 \times w_{QoS}$ with $w_{social} + w_{QoS} = 1$. Also we assume a malicious node drops all packets. A selfish node drops part of packets it receives depending on if it knows the source, current carrier or destination node socially (whether these nodes are in its friend list).

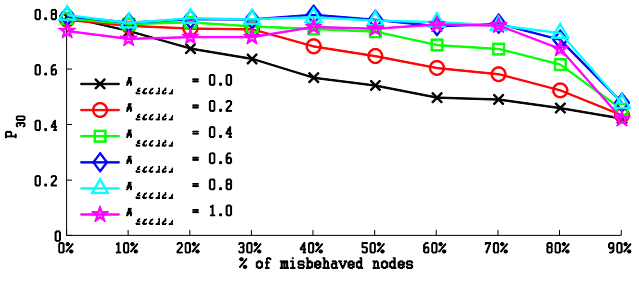


Figure 2: Effect of w_{social} on P_d.

We consider two variations of secure routing protocols: single-copy forwarding (L = 1) and double-copy forwarding (L = 2), where L is the maximum number of carriers to which a node can forward a message. Below we discuss how we identify the best setting for single-copy forwarding. The best setting for double-copy forwarding can be obtained in a similar way, but is not discussed here due to space limitation. Figure 2 shows the effect of w_{social} on P_{30}

for the secure routing application with the population percentage of misbehaved node varying in [0 - 90%]. We observe that using more social trust generally helps generate higher P_d. However, there is a cutoff point after which the effect of increasing the weight of social trust deteriorates. The reason is that using a higher weight of social trust helps the delivery ratio but could result in high message delay. Figure 2 identifies that there exists an optimal (w_{social}, w_{QoS}) set under which P_d is maximized, given the population % of misbehaved nodes as input. For example, the optimal set is w_{soical} = 0.6 and w_{QoS} = 0.4 when there is 30% of misbehaved nodes, and is w_{soical} = 0.8 and w_{QoS} = 0.2 when there is 80% of misbehaved nodes.

6.3 Best Application-Level Trust Optimization Design Settings to Maximize Application Performance

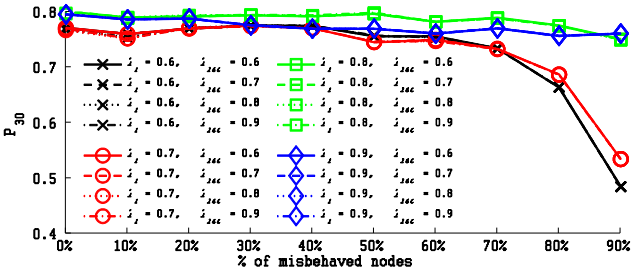
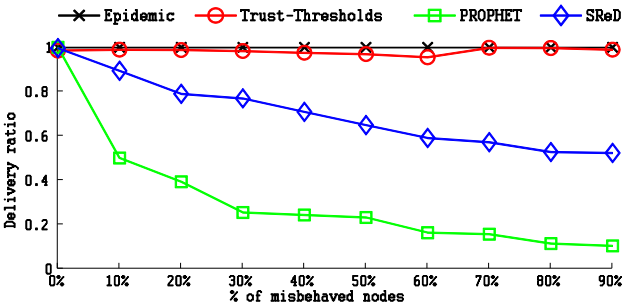


Figure 3: Effect of T_f and T_{rec} on P_d.

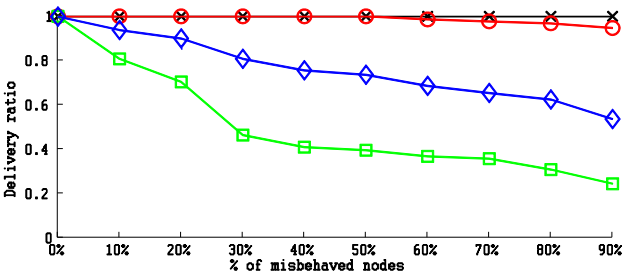
Next we apply the application-level trust optimization design in terms of the best recommender trust threshold $T_{rec,}$ and the best message carrier trust threshold T_f to maximize P_d in response to changing hostility reflected by % of misbehaved nodes. Figure 3 shows P_d vs. (T_f, T_{rec}) with the percentage of misbehaved nodes varying in [0 - 90%]. We notice that by increasing the value of T_f, the value of P_d first increases, and then decreases after a cutoff point (T_f = 0.8). The reason behind this is that using a higher value of T_f helps generate a higher message delivery ratio by choosing only the most trustworthy nodes as message carriers, but it also introduces a higher message delay. We also observe that P_d is relatively insensitive to T_{rec} especially when the value of T_{rec} is higher than 0.6 (the curves in Figure 3 with the same T_f are close to each other). The reason is that the trust value of a malicious node is very likely to be lower than ignorance (set to 0.5), so $T_{rec} \geq 0.6$ can effectively filter out false recommendations from malicious nodes. We see that the optimal (T_f, T_{rec}) to maximize P_d is (0.8, 0.6) when the percentage of misbehaving nodes is low, but is (0.9, 0.7) when the percentage of misbehaving nodes is high (> 85%). The analysis results obtained again can be used to facilitate dynamic trust management.

6.4 Comparative Analysis

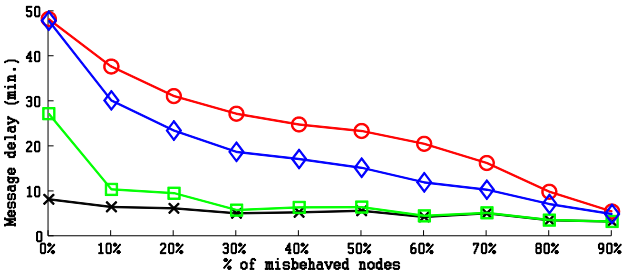
Lastly we conduct a comparative analysis, contrasting our trust-based protocol operating under the best settings identified with existing trust-based (SReD [24]) and non-trust based (PROPHET [15] and epidemic [23]) protocols. We again consider two variations: single-copy forwarding (L = 1) and double-copy forwarding (L = 2) for SReD, PROPHET, and our trust-based secure routing protocol. For our trust-based secure routing protocol, we use the best settings identified for L = 1 and L = 2 as discussed above.



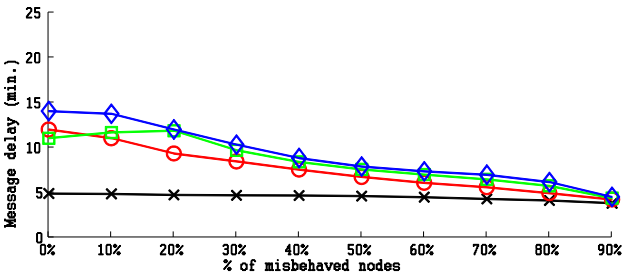
(a) Delivery Ratio (L = 1)



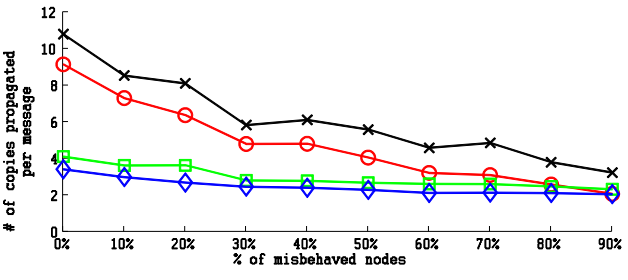
(b) Delivery Ratio (L = 2)



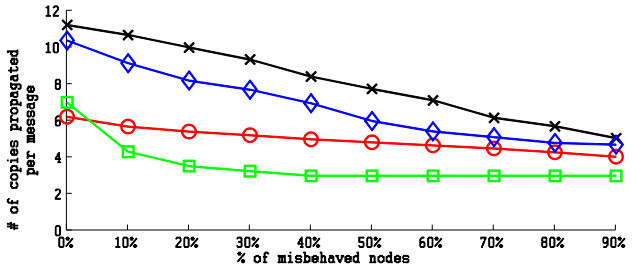
(c) Message Delay (L = 1)



(d) Message Delay (L = 2)



(e) Message Overhead (L = 1)



(f) Message Overhead (L = 2)

Figure 4: Performance Comparison (Analytical Results based on Random Mobility).

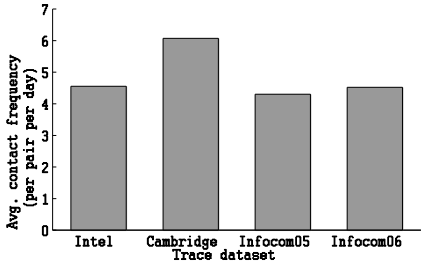
Figure 4 compares the message delivery ratio, delay, and overhead generated by our trust protocol against SReD, PROPHET, and epidemic protocols. The results demonstrate that our trust-based secure routing protocol designed to maximize P_d can effectively trade off message overhead and message delay for a significant gain in delivery ratio. In particular, our trust-based routing protocol outperforms SReD and PROPHET in delivery ratio (both when L = 1 and L = 2). Further, it approaches the ideal performance of epidemic routing in delivery ratio and message delay (when L = 2) without incurring high message overhead.

7 Simulation Validation

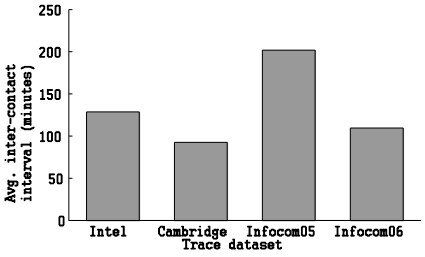
Table 3: Experiment Settings of Mobility Traces.

Trace	*Intel*	*Cambridge*	*Infocom05*	*Infocom06*
Participants	researches, interns	students, faculties	conference attendees	conference attendees
Experiment Time	4 days	5 days	3 days	4 days
Internal Devices	9 (1 stationary)	12	41	98 (20 stationary)
External Devices	119	211	233	4626

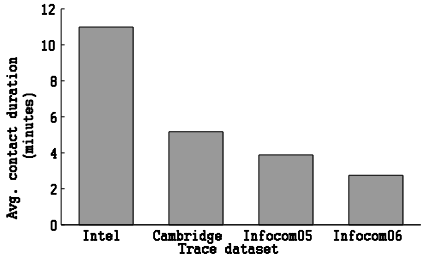
We validate analytical results through extensive simulation using ns-3 [1]. The simulated DTN environment is setup as described in Table 1. We simulate two mobility patterns: a random waypoint mobility model in a 2000 m × 2000 m operational area, with the speed in the range of (0, 2] m/s and pause time of zero, as is used in the theoretical analysis, and real mobility traces. We investigate four mobility traces from [19], namely *Intel*, *Cambridge*, *Infocom05* and *Infocom06*. We describe the experimental settings under which these mobility traces are obtained in Table 3. During the experiment, each *internal device* records the contact/encounter event with other devices (*internal* or *external*). Due to the face that the contact events between external devices are not recorded in the traces, we only consider internal devices in our simulation. In Figure 5, we summarize the contact event statistics, including the average encounter frequency, inter-encounter interval, and encounter duration for each mobility trace. We can see that each pair of nodes will encounter 4 ~ 7 times each day and each encounter event will last 3 ~ 12 minutes on average. Thus, we conclude that these mobility traces can support direct trust detection by means of monitoring and overhearing techniques when two nodes encounter each other.



(a) Frequency

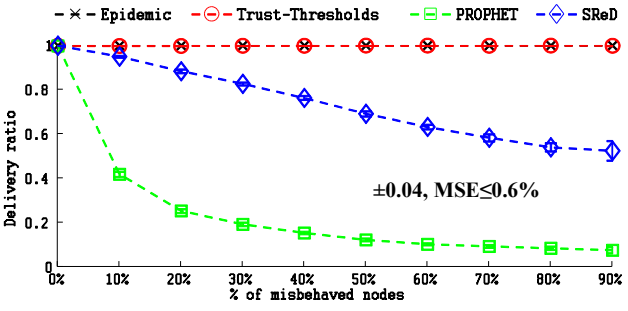


(b) Inter-Contact Interval

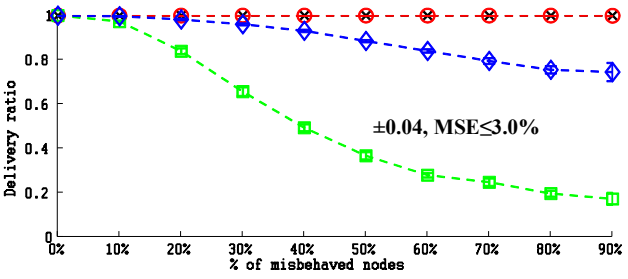


(c) Contact Duration

Figure 5: Contact Events Statistics in Mobility Traces.



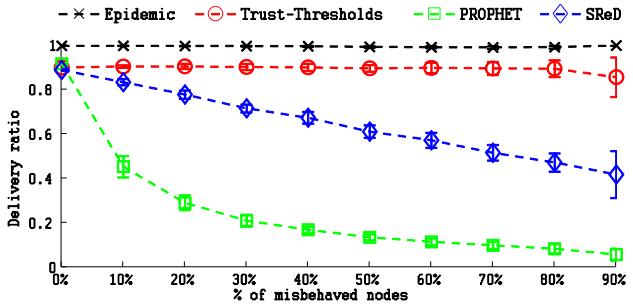
(a) Delivery Ratio (L = 1)



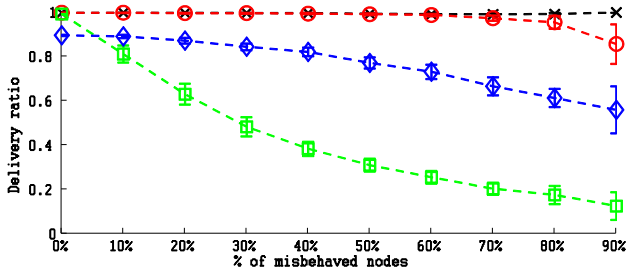
(a) Delivery Ratio (L = 2)

Figure 6: Simulation Results Corresponding to Analytical Results in Figure 4 based on Random Mobility.

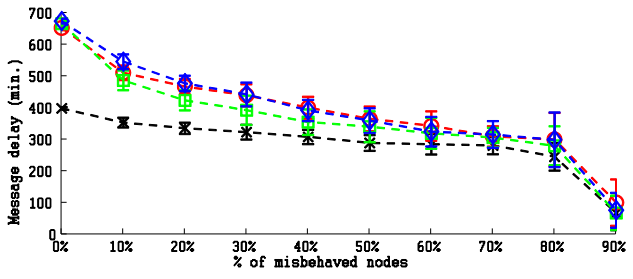
With the random movement model, we collect simulation data to validate analytical results reported earlier. Due to space limitation, we only report two figures (performance comparison in delivery ratio). Figures 6(a) and 6(b) show the simulation results of comparative performance analysis in message delivery ratio, corresponding to Figures 4(a) and 4(b) obtained earlier from analytical calculations. We follow the Monte Carlo simulation and collect observations from sufficient simulation runs with disjoint random number streams to satisfy the ±5% accuracy requirement with confidence level = 95%. The simulation results in Figures 6(a) and 6(b) are remarkably similar to the analytical results shown in Figures 4(a) and 4(b), with the average mean square error (MSE) between the simulation results vs. the analytical results less than 3%.



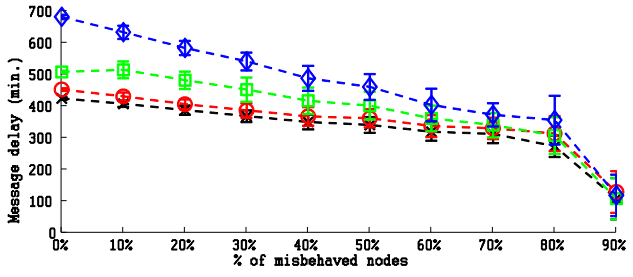
(a) Delivery Ratio (L = 1)



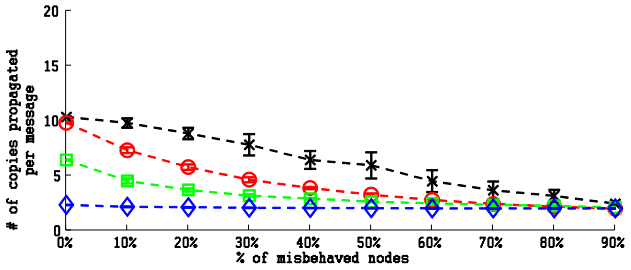
(b) Delivery Ratio (L = 2)



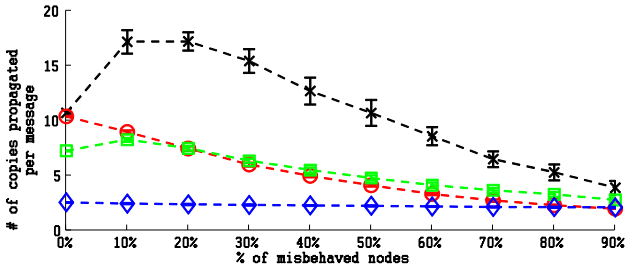
(c) Message Delay (L = 1)



(d) Message Delay (L = 2)



(e) Message Overhead (L = 1)



(f) Message Overhead (L = 2)

Figure 7: Performance Comparison of Routing Protocols based on Mobility Traces.

To test the effect of mobility patterns, Figure 7 shows the simulation results of comparing our trust-based secure routing protocol against SReD, PROPHET, and epidemic protocols, based on *infocom05* mobility traces [19]. The results of the other three mobility traces exhibit the same trend and thus are not shown here. The *infocom05* trace data contain the encounter events collected by Bluetooth devices carried by conference attendees. There are totally 41 Bluetooth devices used to record the encounter events and the experiment last about three days. We use the encounter events in the trace as the time instances to perform trust updating and message forwarding (executed by each node following our trust protocol). In each simulation run, we randomly select part of nodes as malicious or selfish nodes (varying from 0% to 90%) and generate a social friendship matrix [14]. A malicious node performs attacks to disrupt the trust of the DTN, including self-promoting, good-mouthing and bad-mouthing attacks. A selfish node forwards a message only when it is a friend of the source, current carrier, or destination. We collect the results by running 20 replications for each case with disjoint random number streams to achieve high accuracy. The error bar on each curve in Figure 7 shows the confidence interval with confidence level = 95%.

We first observe that Figures 7(a) and 7(b) obtained based on mobility trace simulation exhibit virtually the same trends as Figures 6(a) and 6(b) obtained based on random movement simulation. This supports our claim that our trust-based secure routing protocol can significantly outperform SReD and PROPHET in message delivery ratio (when L = 1 and L = 2) regardless of the node encountering pattern. We further observe that Figure 7 (simulation results based on traces) exhibits remarkably similar trends as Figure 4 (analytical results based on random movements) in

terms of ranking routing protocols in delivery ratio, delay and overhead. As both simulation results based on traces (Figure 7) and random movements (Figure 6) correlate well with analytical results (Figure 4), we conclude that the analytical results obtained and conclusions drawn for our trust-based secure routing protocol designs are valid.

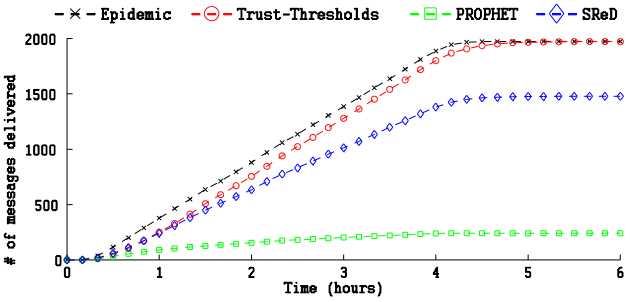
8 DYNAMIC TRUST MANAGEMENT

In this Section, we demonstrate the effectiveness of our dynamic trust management protocol in response to changing environment conditions. Without loss of generality, we consider environment changes in terms of increasing selfish and malicious nodes over time as modeled by the dashed line entities in the SPN model shown in Figure 1 with the transition rates of T_COMPRO and T_SELFISH being λ_c and λ_s, respectively. Under our dynamic trust management protocol, the best protocol settings in terms of (β, λ_d), (w_{social}, w_{QoS}), and (T_f, T_{rec}) identified in Section 6 are applied in response to dynamically changing network conditions to minimize trust bias and to maximize DTN routing performance. Specifically, at runtime, each node senses hostility changes using its trust evaluation results (trust properties in healthiness and unselfishness) toward other nodes in the DTN, and then, based on the detected % of misbehaved nodes, performs a simple table lookup (e.g., into Table 2) to determine and apply the best protocol settings in (β, λ_d), (w_{social}, w_{QoS}), and (T_f, T_{rec}) to minimize trust bias and to maximize DTN routing performance.

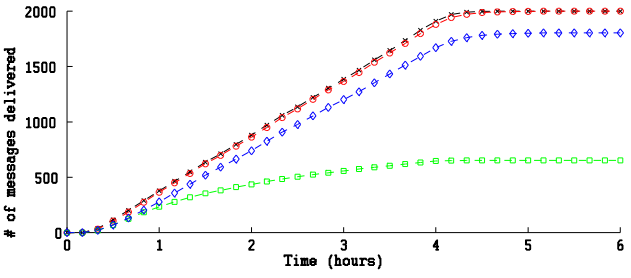
Table 4: Dynamic DTN Environments.

Mobility	***Random Way Point***	***Infocom05 Trace***
Simulation Time	6 hours	80 hours
Compromise Rate (λ_c)	0.25 / hour	0.018 / hour
Selfishness Rate (λ_s)	0.25 / hour	0.018 / hour
# of Messages per Run	2000	2000
Warm-up Time	10 minutes	8 hours
Maximum Delay	2 hours	10 hours
Routing Protocol	Trust-Based Routing, PROPHET, SReD, and epidemic Routing	
MAC & PHY	IEEE 802.11a, Ad-Hoc	
Energy Model	3V, 17.4mA TX, 5.8mA RX, 0mA IDLE	

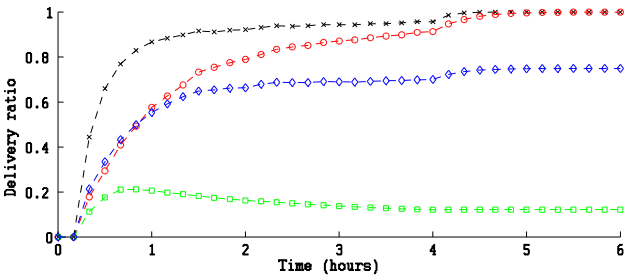
Below we perform a comparative analysis of our dynamic trust management protocol operating under best protocol settings dynamically for DTN routing against PROPHET, SReD, and epidemic routing. Similar to Section 7, we consider two mobility patterns: the random waypoint mobility model and the *infocom05* mobility trace. Table 4 describes the simulation setup for each mobility pattern. Initially, there is no malicious or selfish node in the network. As time progresses, nodes become malicious and selfish with rates λ_c and λ_s respectively. The data reported is based on the average of 2000 messages. The last message is issued a few hours (the maximum delay) before the end of simulation to ensure sufficient time for message delivery.



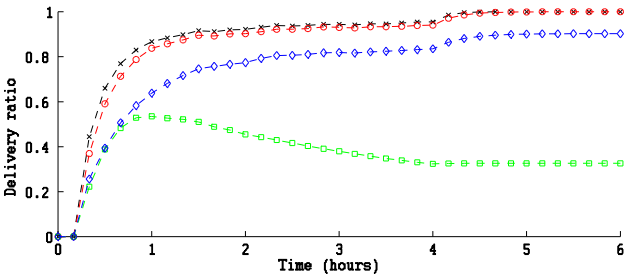
(a) # of Delivered Messages (L = 1)



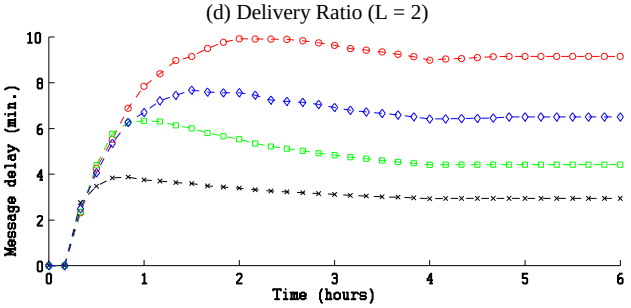
(b) # of Delivered Messages (L = 2)



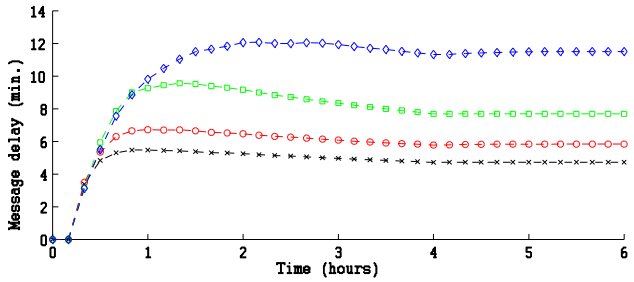
(c) Delivery Ratio (L = 1)



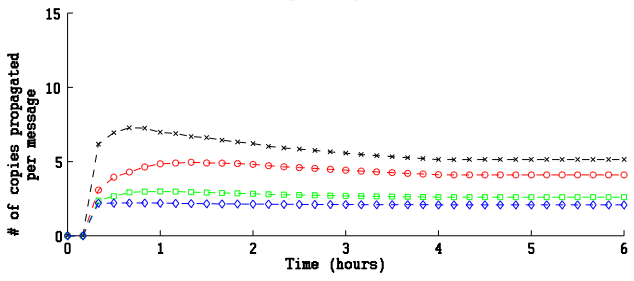
(d) Delivery Ratio (L = 2)



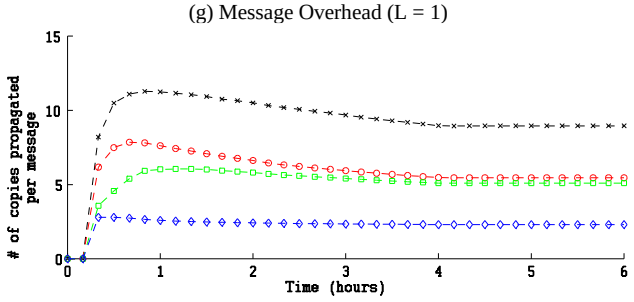
(e) Message Delay (L = 1)



(f) Message Delay (L = 2)



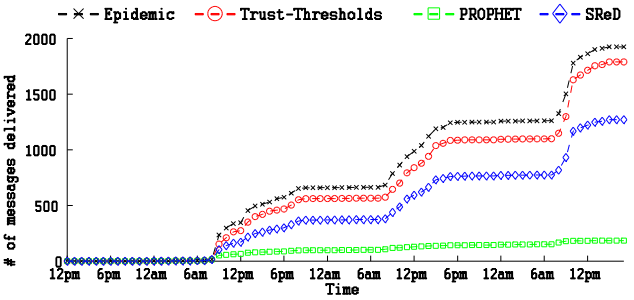
(g) Message Overhead (L = 1)



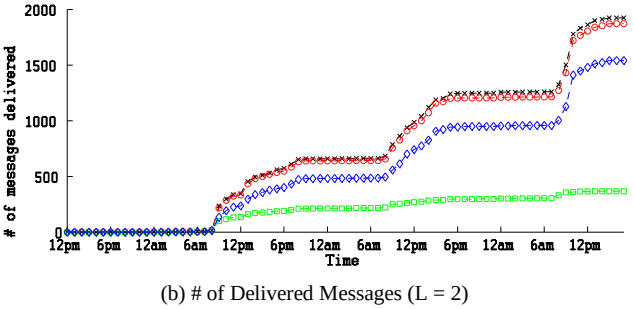
(h) Message Overhead (L = 2)

Figure 8: Performance Comparison of Routing Protocols based on Random Mobility in Dynamic DTN Environments.

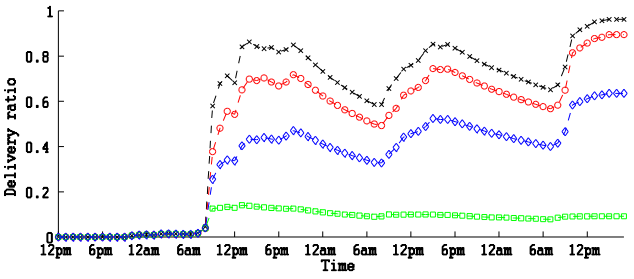
Figure 8 shows performance comparison results based on the random waypoint mobility model. We observe that both our trust-based routing protocol and epidemic routing protocol can successfully deliver messages close to 100%, while the other two protocols (PROPHET and SReD) have a significantly low delivery ratio. The reason is that our trust-based routing protocol operating under the best (β, λ_d) setting can accurately identify misbehaved nodes with minimum trust bias (through the healthiness and unselfishness trust properties), thus avoiding message forwarding to misbehaved nodes. Moreover, our trust-based routing protocol operating under the best (w_{social}, w_{QoS}) and (T_f, T_{rec}) protocol settings uses the best trust formation and application-level optimization design settings to maximize the DTN application performance in delivery ratio. We also observe that because the best protocol settings applied are geared toward maximizing the delivery ratio with P_d (delay is limited to d minutes) as the objective function, it may lead to a higher message delay (as indicated in Figure 8(e)) compared with other schemes, because only a smaller set of nodes would be selected as message carriers. However we see that when two copies (L=2) are allowed, our trust-based routing protocol approaches the ideal performance of epidemic routing in delivery ratio and message delay (Figure 8(f)) without incurring high message overhead (Figure 8(h)).



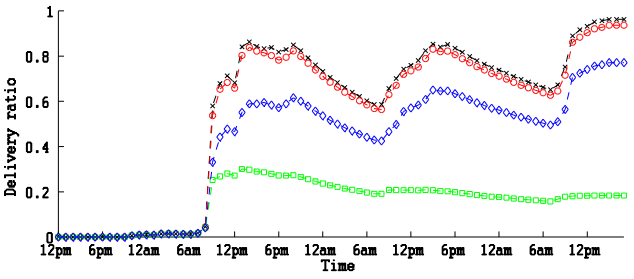
(a) # of Delivered Messages (L = 1)



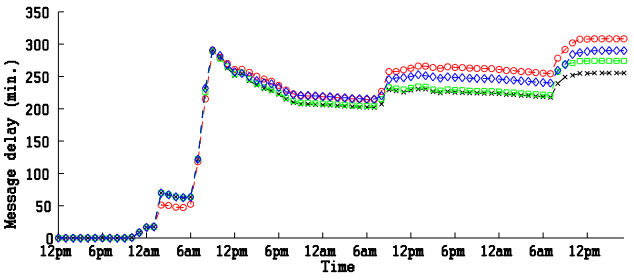
(b) # of Delivered Messages (L = 2)



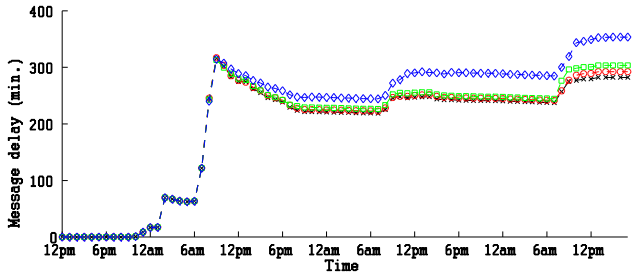
(c) Delivery Ratio (L = 1)



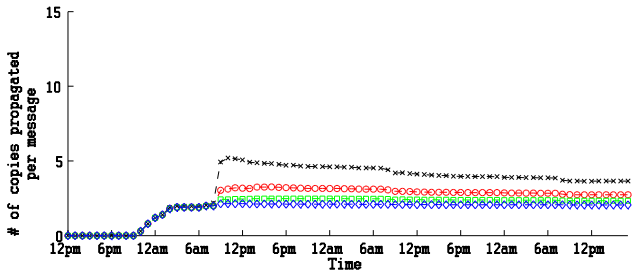
(d) Delivery Ratio (L = 2)



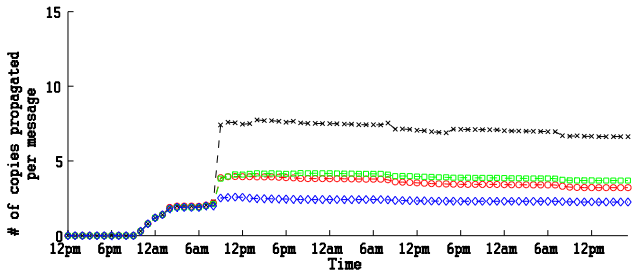
(e) Message Delay (L = 1)



(f) Message Delay (L = 2)



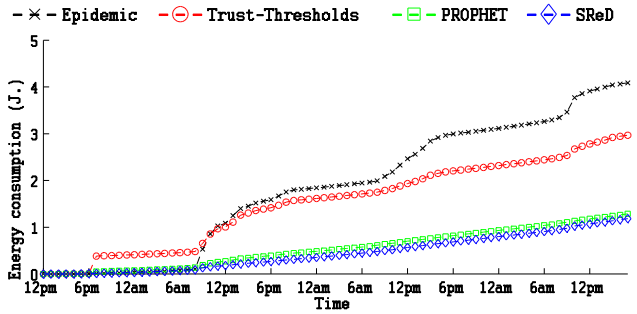
(g) Message Overhead (L = 1)



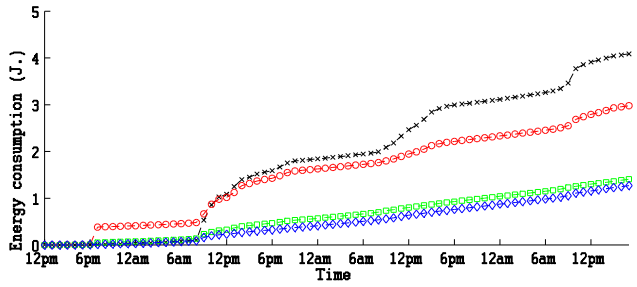
(h) Message Overhead (L = 2)

Figure 9: Performance Comparison of Routing Protocols based on Mobility Traces in Dynamic DTN Environments.

Figure 9 shows performance comparison results based on the *infocom05* mobility trace. We first observe that there are three peak periods in message delivery. This is caused by the three daytime periods in which people are active and most of the messages are delivered. Only a small fraction of the messages are forwarded and delivered during night. The curves in Figure 9 have the same performance trend as those in Figure 8, thus demonstrating the effectiveness of our dynamic trust management protocol regardless of the mobility pattern. This further validates our dynamic trust management design and its application to DTN routing in real DTN environments.



(a) Energy Consumption (L = 1)



(b) Energy Consumption (L = 2)

Figure 10: Energy Consumption of Routing Protocols based on Mobility Traces in Dynamic DTN Environments.

Lastly, we measure the protocol maintenance overhead of our trust-based routing protocol in terms of energy consumption, against PROPHET, SReD, and epidemic routing. Table 4 describes our energy model. We set the energy consumption rate in idle mode to zero to isolate its effect (since all protocols will consume the same amount of energy when a node is idle). Figure 10 shows energy consumption vs. time for all protocols. While our trust-based routing protocol consumes more energy than PROPHET and SReD because of extra overhead in trust protocol execution, it consumes significantly less energy than epidemic routing. This result, along with the performance comparative results shown in Figure 9, supports the claim that our trust-based based protocol approaches the ideal performance of epidemic routing without incurring high message or protocol maintenance overhead.

9 Conclusion

In this paper, we designed and validated a trust management protocol for DTNs and applied it to secure routing to demonstrate its utility. Our trust management protocol combines QoS trust with social trust to obtain a composite trust metric. Given an operational profile describing the network environment and node behaviors as input, our design allows the best trust setting (β, λ_a) for trust aggregation to be identified so that subjective trust is closest to objective trust for each individual trust property for minimizing trust bias. Further, our design also allows the best trust formation (w_{social}, w_{QoS}) and application-level trust setting (T_f, T_{rec}) to be identified to maximize application performance. We demonstrated how the results obtained can facilitate dynamic trust management for DTN routing in response to dynamically changing conditions at runtime.

We performed a comparative analysis of trust-based secure routing running on top of our trust management protocol with existing trust-based (SReD) and non-trust-based (PROPHET) routing protocols in DTNs. Our results backed by simulation validation demonstrate that our trust-based secure routing protocol outperforms SReD and PROPHET dynamically. Further, it approaches the ideal performance of epidemic routing in delivery ratio and

message delay without incurring high message or protocol maintenance overhead.

In the future, we plan to explore other trust-based DTN applications with which we could further demonstrate the utility of our dynamic trust management protocol design. We also plan to implement our proposed dynamic trust management protocol on top of a real DTN architecture [7] to further validate the protocol design, as well as to quantify the protocol overhead.

ACKNOWLEDGEMENTS

This work was supported in part by the Army Research Office under Grant W911NF-12-1-0016.

REFERENCES

[1] "The ns-3 Network Simulator," Nov. 2011, http://www.nsnam.org/.

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

[3] E. Ayday, H. Lee, and F. Fekri, "An Iterative Algorithm for Trust Management and Adversary Detection for Delay Tolerant Networks," *IEEE Transactions on Mobile Computing*, DOI: 10.1109/TMC.2011.160, online avaiable, 2011.

[4] E. Bulut, Z. Wang, and B. Szymanski, "Cost Effective Multi-Period Spraying for Routing in Delay Tolerant Networks," *IEEE/ACM Transactions on Networking*, vol. 18, no. 5, 2010, pp. 1530-1543.

[5] E. Bulut, Z. Wang, and B. K. Szymanski, "Impact of Social Networks on Delay Tolerant Routing," in *IEEE Global Telecommunications Conference*, Honolulu, HI, Nov. 2009, pp. 1-6.

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

[7] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss, "Delay-Tolerant Networking Architecture," *RFC 4838*, IETF, 2007.

[8] M. Chang, I.-R. Chen, F. Bao, and J.-H. Cho, "Trust-Threshold Based Routing in Delay Tolerant Networks," in *5th IFIP International Conference on Trust Management*, Copenhagen, Denmark, June 2011, pp. 265-276.

[9] I.-R. Chen, F. Bao, M. Chang, and J.-H. Cho, "Trust Management for Encounter-Based Routing in Delay Tolerant Networks," in *IEEE Global Communications Conference*, Miami, Florida, USA, Dec. 2010, pp. 1-6.

[10] 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.

[11] G. Dini, and A. L. Duca, "A Reputation-Based Approach to Tolerate Misbehaving Carriers in Delay Tolerant Networks," in *15th IEEE Symposium on Computers and Communications*, Riccione, Italy, June 2010, pp. 772-777.

[12] F. Li, J. Wu, and A. Srinivasan, "Thwarting Blackhole Attacks in Disruption-Tolerant Networks using Encounter Tickets," in *IEEE Conference on Computer Communications*, 2009, pp. 2428-2436.

[13] N. Li, and S. K. Das, "RADON: Reputation-Assisted Data Forwarding in Opportunistic Networks," in *2nd ACM International Workshop on Mobile Opportunistic Networking*, Pisa, Italy, Nov. 2010, pp. 8-14.

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

[15] A. Lindgren, A. Doria, and O. Schelen, "Probabilistic routing in intermittently connected networks," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 7, no. 3, 2003, pp. 19-20.

[16] A. Mei, and J. Stefa, "Give2Get: Forwarding in Social Mobile Wireless Networks of Selfish Individuals," in *IEEE International Conference on Distributed Computing Systems*, Genoa, Italy, June 2010, pp. 488-297.

[17] J. D. Musa, "Operational Profiles in Software-Reliability Engineering," *IEEE Software*, vol. 10, no. 2, March 1993, pp. 14-32.

[18] A. Piyatumrong, P. Bouvry, F. Guinand, and K. Lavangnananda, "Trusted Spanning Trees for Delay Tolerant Mobile Ad Hoc Networks," in *IEEE Conference on Soft Computing in Industrial Applications*, 2008, pp. 131-136.

[19] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, and A. Chaintreau. "CRAWDAD data set Cambridge/haggle (v. 2009-05-29)," May 2009; http://crawdad.cs.dartmouth.edu/cambridge/haggle.

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

[21] C. Song, and Q. Zhang, "COFFEE: A Context-Free Protocol for Stimulating Data Forwarding in Wireless Ad Hoc Networks," in *IEEE Communictions Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, Rome, Italy, June 2009, pp. 1-9.

[22] K. S. Trivedi, "Stochastic Petri Nets Package User's Manual," Department of Electrical and Computer Engineering, Duke University, 1999.

[23] A. Vahdat, and D. Becker, *Epidemic Routing for Partially Connected Ad Hoc Networks*, Technical Report, Duke University, 2000.

[24] Z. Xu, Y. Jin, W. Shu, X. Liu, and J. Luo, "SReD: A Secure Reputation-Based Dynamic Window Scheme for Disruption-Tolerant Networks," in *IEEE Military Communications Conference*, Boston, MA, Oct. 2009, pp. 1-7.

[25] H. Zhu, X. Lin, R. Lu, Y. Fan, and X. S. 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. 4628-4639.