Trust-based Scheme for Cheating and Collusion Detection in Wireless Multihop Networks

Normalia Samian
Faculty of Computer Science and Information Technology,
Universiti Putra Malaysia
UPM Serdang, Selangor, Malaysia 43400
normalia@upm.edu.my

Winston K. G. Seah
School of Engineering and Computer Science, Victoria
University of Wellington
Wellington, New Zealand 6140
winston.seah@ecs.vuw.ac.nz

ABSTRACT

To evaluate a node's cooperativeness/selfishness behaviour in a heterogeneous wireless environment, global reports from other nodes in wireless multihop networks (WMNs) may be required to strengthen the judgment made using local observation. However, it cannot be assumed that a node is always honest in sharing the behaviour information. A group of nodes might collude to falsely accuse/praise a particular node to gain communication benefits at the cost of other nodes' resources. This paper proposes a filtering mechanism named Trust Features-based Evidence (TFE) by formulating evidence that is based on trust features to reduce the number of liar nodes and thus, ensuring the authenticity of the shared information. Assisted by an efficient cross-checking algorithm, the TFE mechanism is able to reduce the number of cheating nodes over time and the rate of illegitimate collusion, by having detected cheaters aware that the only way to survive in the network is by presenting genuine information.

KEYWORDS

recommendation, trust, node collusion, selfishness, wireless multihop networks

ACM Reference format:
Normalia Samian and Winston K. G. Seah. 2017. Trust-based Scheme for Cheating and Collusion Detection in Wireless Multihop Networks. In *Proceedings of MobiQuitous 2017, Melbourne, VIC, Australia, November 7–10, 2017,* 10 pages.
https://doi.org/10.1145/3144457.3144486

1 INTRODUCTION

Evaluation of node behaviour can be done via two main approaches viz. local and global observation. Although not mandatory, the former is essential for a node's self-reference and can be done on a node-to-node basis using several observation/monitoring techniques such as overhearing, packet acknowledgment, and probing. However, in the event that a node either cannot access another node's behaviour information or wants to strengthen its own local assessment, the latter approach is chosen. The global observation approach works by having nodes in the network share their local observation judgments which are used to determine nodes' behaviour based on majority of votes. The globally shared behaviour information is also known as *recommendation* [23] and with the diversified nature of node behaviour in heterogeneous wireless environments dominated by wireless multihop communications, not all recommenders will share honest information; this leads to fallacy and unnecessary punishment/reward that are detrimental to network performance [4]. Various solutions have been proposed to overcome false recommendation issues with trust-based mechanisms but, to the best of our knowledge, none have addressed the issue of validating the honesty of the information shared by recommenders which was locally observed and how much they can be trusted.

We propose the Trust Features-based Evidence (TFE) mechanism that utilizes a trust module which acts as a filter for any node receiving the global report to carry out cross-checking when validating the recommender and the shared information. Thus, the proposed approach adopts distributed trust computation, but a centralized approach can also be supported on logically centralized network architectures like Software-Defined Networking (SDN) [1] by implementing TFE on the SDN controller to evaluate nodes' behaviour. The main difference between the trust module proposed in this study and existing trust-based schemes is that both recommender and its carried information are evaluated in order to measure the trust value, rather than just assessing the recommender only. In addition, we build on the proven Compare and Measure Selfishness Detection (CMSD) mechanism to ensure observation accuracy [21, 22].

Our contributions can be summarized as follows:

(i) a trust-based scheme to filter false recommendations by assigning a trust value to recommender node which is derived from trust features that reflect the forwarding/communication behaviour of the node;
(ii) the formulation of trust feature that is based on extracted information, from a proven local observation mechanism (CMSD), which consists of an observed node's forwarding details that can be evaluated as an input to assess the recommender node's honesty; and
(iii) the detection of cheaters and node collusion using an efficient cross-checking algorithm.

The remainder of this paper is organized as follows. Section II discusses related work on trust-based recommendation schemes.

MobiQuitous 2017, November 7–10, 2017, Melbourne, VIC, Australia
© 2017 Association for Computing Machinery.
ACM ISBN 978-1-4503-5368-7/17/11…$15.00
https://doi.org/10.1145/3144457.3144486

Discussion on some preliminaries and proposed scheme are presented in Section III and IV respectively. Section V presents the results from our performance evaluation before concluding in Section VI.

2 RELATED WORK

Recommendation-based approach has been widely used as a way to assess node behaviour via global judgments from other nodes. Decision on a node's behaviour is made based on majority of votes as either bad or good. However, honest recommendations cannot be assumed as nodes may be sharing false information for their own benefits which remain challenging issues to be addressed.

A node's trustworthiness level can be derived based on certain trust parameters. CONFIDANT [3] introduced a *trust manager* that cross-checks a particular misbehaviour report based on learning observation experience and deterioration test prior to making decision. However, aside from the critical assumption of being able to gather accurate malicious evidence and alarm information, this scheme is also susceptible to false accusation by colluding nodes towards a well-behaved one. Similar node collusion problems are also faced by subsequent proposed schemes, such as [8, 13]. A social-based trust approach known as Recommendation Exchange Protocol (REP) [24] takes into account the maturity of node relationship to evaluate trust level (i.e. the longer any two nodes have known each other, the higher the trust level.) The resilience of REP against false recommendation and node collusion attack is justified on the premise that a node that has longer relationship with the node it is recommending would provide a more truthful recommendation. Like ICARUS [5], where robustness towards liar nodes is evaluated based on different values of cheating probabilities, REP lacks the ability to identify/ascertain the lying behaviour itself and cheating actions are assumed to be accurately detectable.

In [6], to ensure false recommendation is low, the scheme only considers reports from recommenders that have positive evaluation on an observed node, which are presumed to have frequent mutual interactions. This work proposed the concept of *trust chain* whereby the longer the length of indirect recommenders in between of a trustor and a trustee, denotes lower trust value. However, depending on factors like path reliability and observation period, different optimal path lengths have been found to gain accurate trust value. The concept of considering only positive recommendations to evaluate node's behaviour has also been proposed in [18] with a scheme known as Collaborative Reputation (CORE). Although false rating can be reduced at certain points, but, inaccurate evaluation occurs when nodes are colluding to rate a bad node as good, which leads to trusting the wrong node.

A trust model named Trust-based Exclusion Access-control Mechanism (TEAM) proposed in [11] introduced a judiciary-based system to evaluate node's trustworthiness. Behaviour information based on local observation of each node is submitted to jury nodes for assessment prior to making final decision based on majority of the juries' votes. Although the model practiced random selection of juries to avoid node collusion, the reliability of the juries' behaviour and judgment have not been validated beforehand and thus, prone to false recommendation by the juries.

The work in [23] proposed a clustering approach to overcome dishonest recommendation by measuring trust value using social properties such as number of interactions and length of distance between the recommenders and their reported nodes. With the help of recommendation and cluster managers, confidence and deviation values are measured as a mean to detect liars more efficiently. However, this scheme is vulnerable to high communication overheads and delay with the appointment of two-tiers central managers, which have not been proven otherwise in the work. Clustering concept which works as a filter has also been applied in [25]. However, description about how the trustworthiness of the appointed cluster's managers is evaluated has not been discussed in both schemes which are of similar concerning issue highlighted for judiciary-based system in [11].

3 PRELIMINARIES

3.1 Trust Establishment

The concept of trust has been widely deployed in wireless multihop networks to enhance collaborative and cooperative communications among nodes. In the autonomous environment of wireless multihop networks (WMNs), it is important for each node in the network to build mutual trust relationships with many (if not all) other nodes to maximize the assurance of successful data transmission. However, due to diversified behaviour of the nodes, trust takes time and effort to establish, taking into consideration aspects like mobility, past interaction experience, anonymity level, history of packet forwarding, positive recommendations, etc. Hence, the information gathered might not be sufficient to build an acceptable level of initial trust bootstrapping or the trust level information stored may be outdated and does not reflect the current behaviour state of a node.

Apart from time and dynamical environment constraint, building trust is especially challenging with the presence of cheater nodes that provide dishonest information, and worsen with the scenario of illegitimate colluding nodes. Many existing works that proposed solutions to address cheaters and colluding nodes are actually lacking in identifying the behaviours accurately. If there is some sort of cheating identification, it is evaluated based on the robustness of the proposed schemes towards different numbers of cheating probabilities such as in [5] or by having more information gatherings from other nodes that have sufficient interactions with an observed node such as in [23]. But, how is the cheating action itself detected has not been fully addressed. Motivated by such issues, this study has proposed solutions to detect cheating action using trust concept by ensuring that the trust is established using significant evidences.

In this study, trust concept has been used to verify the authenticity of the behaviour information shared during global assessment by validating the trustworthiness of the recommender (i.e. observer node) and its shared information based on certain requirements. According to Li and Singhal [15], and Cho *et al.* [7], the requirements can be formulated based on evidences in the form of numerical value or tangible properties of a node such as public key and identity or any other properties that can prove a node's trustworthiness either it is self-generated or assigned by other nodes. In existing works, the trustworthiness of a node is evaluated using several parameters such as the length of trust chain [6], list of mutually connected friends [10], history of successful packet forwarding [17], and recommended reputation/trust level [12, 23]. When it comes to recommendation-based approach, typically, when the recommender node surpasses certain trust metrics requirement based on its investigated previous communication history, its recommended information such as the trust level or forwarding ratio of an observed node is also considered trusted without investigating the information itself. Hence, assessment should cater for both; the recommender and its corresponding shared information as has been proposed in this study. This can be made possible with the use of local observation mechanism like CMSD as it contains information that can be extracted as evidences to evaluate its legitimacy.

3.2 CMSD Mechanism for Local Observation

In this paper, the local observation technique used is based on the CMSD mechanism [21, 22]. The main concept of CMSD is that an observer node ID_V evaluates and compares the forwarding rate of a relay node ID_R against different incoming packets' forwarding requests (including ID_V's request). A node's behaviour is measured by using quantity of packets forwarded/sent to compute the packet forwarding speed and sending speed of a relay node R towards different incoming requests, which are used to determine the correlation coefficient, r, of the transmission rate of data packets by sender and relay node to determine cooperativeness/selfishness level [21].

The r values of an ID_R for different requestor nodes will then be used to determine its fairness, f, level. A node's behaviour is categorized based on combination of four classifications, i.e. *cooperative* (c) or *selfish* (s) and *fair* (f) or *unfair* (u), from which we can obtain four possible combinations of pairs: $(c, f)_{(a,b)}$, $(c, u)_{(a,b)}$, $(s, u)_{(a,b)}$, and $(s, f)_{(a,b)}$, where a is the identity of a relay node, ID_R and b is the identity of either an observer, ID_V, or a competitor node, ID_C. The pairs have been arranged in such a sequence to denote the decreasing level of cooperativeness and fairness from good to worst state which are represented using our proposed fairness metric, *fm*, as shown in Table 1.

Table 1: Fairness Metrics

Behaviour Pairs	Fairness Metric, *fm*
$(c, f)_{(a,b)}$	1.0
$(c, u)_{(a,b)}$	0.75
$(s, u)_{(a,b)}$	0.5
$(s, f)_{(a,b)}$	0.25

The fairness metric, *fm*, together with correlation value, r, of a relay node j are used as inputs to calculate the overall probability function of node j's forwarding effort, FR_{ij}, observed by node i by obtaining arithmetic mean $(\overline{a})$ of the two elements:

$$\overline{a} = FR_{ij} = \frac{(r_j + fm_j)}{n(r_j) + n(fm_j)} \tag{1}$$

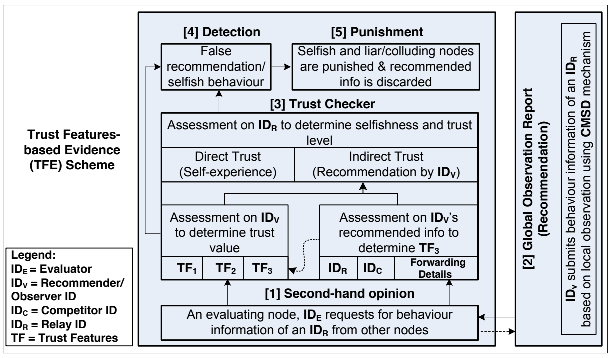
Trust Features-based Evidence (TFE) Scheme
[4] Detection
False recommendation/ selfish behaviour
[5] Punishment
Selfish and liar/colluding nodes are punished & recommended info is discarded
[3] Trust Checker
Assessment on ID_R to determine selfishness and trust level
Direct Trust (Self-experience)
Indirect Trust (Recommendation by ID_V)
Assessment on ID_V to determine trust value
Assessment on ID_V's recommended info to determine TF_3
TF_1
TF_2
TF_3
ID_R
ID_C
Forwarding Details
[1] Second-hand opinion
An evaluating node, ID_E requests for behaviour information of an ID_R from other nodes
[2] Global Observation Report (Recommendation)
ID_V submits behaviour information of an ID_R based on local observation using CMSD mechanism
Legend:
ID_E = Evaluator
ID_V = Recommender/ Observer ID
ID_C = Competitor ID
ID_R = Relay ID
TF = Trust Features

Figure 1: TFE Framework

where $n(\cdot)$ is the cardinality of the respective element set and $0 \leq \overline{a} \leq 1$. This information is submitted by an observer node i to a collector node or central agent (CA) collecting the global information which evaluates the average forwarding ratio of a node j based on this formula:

$$FR_{CA,j} = \frac{\sum_{i \in N_R}^{n(N)} FR_{ij}}{n(N_R)} \tag{2}$$

where N is a set of all nodes, and $N_R \in N$ is a subset of nodes that submit behaviour information reports. If $FR_{CA,j}$ is below the threshold value $FR_{Th} = 0.5$, node j is considered as selfish [5, 22].

4 TRUST FEATURES-BASED EVIDENCE (TFE)

4.1 Framework

This section provides overall description on the important elements of the proposed TFE scheme. As shown in Figure 1, the general framework consists of five modules, each briefly explained below:

[Module 1] Second-hand opinion: The first module starts from the process of an evaluating node, ID_E, requesting for second opinion from other nodes to strengthen its judgment of a particular observed node, ID_R, in the network. This will create a collection of global reports from other nodes about the forwarding behaviour of the observed node.

[Module 2] Global observation report (recommendation): It responds to requests from other nodes for behaviour information on a particular relay node, ID_R. Under the policy of the proposed TFE scheme, it is assumed that all nodes are adopting CMSD mechanism in performing local observation and all reporting nodes must append the observed behaviour information with additional information as listed below:

(i) ID of a recommender/observer node - ID_V

(ii) ID of a node in which forwarding comparison has been made in performing CMSD or also known as requestor/competitor node - ID_C

(iii) The observed relay node's forwarding details:

  (a) identity - ID_R

  (b) behaviour classification (e.g. unfairly selfish node)

  (c) correlation value - r

  (d) fairness metric - *fm*

  (e) forwarding ratio towards ID_V and ID_C - FR_{ij}

  (f) timestamp of CMSD operation, start & end time - t_s & t_e

[Module 3] Trust checker: Each node has its own trust checker module to carry out cross-checking tasks to validate the honesty of a recommender node and its shared information. Assessment is done on both the recommender ID_V and observed relay node(s) ID_R whereby information from ID_R will be extracted and formulated as one of the trust features (TF_3) to compute the total trust value ($TV_{ij}(t)$ - cf: Eqn. (3)) of the recommender node ID_V at time t which will become an input to evaluate selfishness and trust level ($TR_{ij}(t)$ - cf: Eqn. (11)) of an ID_R via indirect trust. Two important elements in evaluating ID_R are direct and indirect trust which makes it a hybrid trust computation module [12]. The former is obtained via past self-experience of a node with a recommender whereby evaluation on the trustworthiness of the recommender node has been made in previous communication sessions.

[Module 4] Detection: If a recommender node's trust value is found to be below a minimum threshold which also denotes false recommendation, it will be labeled as a "liar" and in the case where more than two nodes are detected to have the same trait of cheating history, they will be listed as colluding nodes.

[Module 5] Punishment: The shared selfishness/cooperativeness behaviour is validated in this module. If the shared behaviour information is obtained from an untrusted node identified in Module 4, the information will be discarded as it is considered tampered. On the other hand, genuine selfishness behaviour provided by trusted nodes will result in the selfish node being punished.

4.2 Recommendation Policy

As aforementioned, each node in this study performs local observation using CMSD mechanism and will share them with other nodes by transmitting packet containing the observed node's forwarding information (i.e. ID_R) based on format shown in Figure 2. Transmitting this packet in the network requires minimum resource consumption due to its small size, however, minimizing its transmission will help in reducing network traffic which has become the motivation for this study to only have the information transferred upon request rather than having periodical network-wide broadcast. One of the CMSD's requirements is that an observer node must compare the forwarding rate of a relay node towards its request and at least one other node (i.e. competitor node, ID_C) at the same time. It is assumed that such information is available as it is possible that a relay node tends to multiple forwarding requests from different nodes at the same time [9]. Hence, appending all the information listed in Figure 2 is possible such that inability to provide sufficient information will decrease the recommender node's trust value. Upon receiving the information packet, the node that requested for the recommended behaviour information will store them in its buffer, and later perform behaviour assessment on ID_V and ID_R. For the former, assessment on its behaviour is based on trust features.

Identity			Behaviour classification of ID_R $(c\|\|s,f\|\|u)_{a,b}$	r	fm	CMSD Timestamp	
ID_V	ID_C	ID_R		FR_{ij}		Start (t_s)	End (t_e)

ID_V – recommender node; ID_C – competitor node; ID_R – relay node; $a = ID_V$; $b = ID_C$; r – correlation value; fm – fairness metrics; FR_{ij} – forwarding ratio ; c – cooperative; s – selfish; f – fair; u – unfair

Figure 2: Packet Information.

4.3 Trust Features Formulation

4.3.1 *Assessment of recommender node, ID_V, and its shared information.*

As shown in module 3, a recommender node's behaviour is evaluated using three trust features, namely: blacklist record (TF_1), combination of direct and indirect trust (TF_2), and input from CMSD operation (TF_3). The three trust features will be assigned weight values and combined together to form a single trust value of a recommender node ID_V at time t, denoted as $TV_{ij}(t)$, as follows:

$$TV_{ij}(t) = \omega_1 * TF_1(t) + \omega_2 * TF_2(t) + \omega_3 * TF_3(t) \tag{3}$$

where $\omega_1 + \omega_2 + \omega_3 = 1$. Throughout this paper, the notation i represents a node which evaluates/recommends the behaviour/trust value of an observed/recommended node j. For TF_1, it is obtained from an evaluating node's own record of an ID_V (if any) to check whether or not it is malicious node and has been blacklisted, whereby:

$$TF_1 = \begin{cases} 0, & \text{if blacklisted.} \\ 1, & \text{if not blacklisted.} \end{cases} \tag{4}$$

This feature becomes the main condition that would affect the rest of the node's trust calculation as being malicious is more hazardous than being selfish.

Next, TF_2 is formulated based on a combination of the evaluating node's direct and indirect trust lists of an ID_V as both nodes might have accumulated communication history previously. The direct trust indicates that the evaluating node, ID_E, has a direct interaction with ID_V. The direct trust value is computed using Beta probability distribution function [2, 23] which is estimated using two parameters (α, β). We chose this method to evaluate the direct trust value because the information is unlikely to be prone to false recommendation (with the assumption that local observation is accurate), such that positive and negative behaviours can be mapped into binary representations more confidently. Since we are using the CMSD mechanism for local observation, α and β cannot be simply represented by accumulation of forwarded and dropped packets respectively as in [23]. Instead, we let α denotes cooperative behaviour of a node with forwarding ratio (Eq. (1)), $FR_{ij} \geq 0.5$, and β denotes selfish behaviour with $FR_{ij} < 0.5$. The Beta distribution function can be defined as Gamma function, $\Gamma(\cdot)$, as follows:

$$f(p|\alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)(\beta)} p^{\alpha-1}(1-p)^{\beta-1} \tag{5}$$

where $0 \leq p \leq 1$; $\alpha, \beta > 0$, and $p \neq 0$ if $\alpha < 1$ and $p \neq 1$ if $\beta < 1$, and the expectation given by:

$$f(p) = \frac{\alpha}{\alpha + \beta} \tag{6}$$

If the number of cooperative (α) and selfish (β) occurrences of a node are represented by c and s respectively, where c and $s \geq 0$, the expectation of beta probability function which also denotes the direct trust (DT_{ij}) value of an evaluator node i towards an observed (recommender) node j can be defined as:

$$f(p) = DT_{ij} = \frac{c_{ij} + 1}{(c_{ij} + 1) + (s_{ij} + 1)} \tag{7}$$

On the other hand, the indirect trust of an evaluating node ID_E towards a recommender node ID_V needs to be considered if both nodes do not have a priori direct interaction and the direct trust value is null. However, this information is only sought from the ID_E's own record which has been pre-processed in previous communication. This means that an ID_E will not be broadcasting behaviour recommendation request from other nodes during the evaluating process or might do so at another time for future evaluation. When node i requires recommendation on node j, it can be obtained from one of node j's neighbours, e.g. node h, that can directly observe j. When node i receives N recommendations on node j, the indirect trust is determined as an average of forwarding ratio using the following equation:

$$IT_{ij} = \sum_{h=1}^{N} \frac{DT_{kj}}{N} \tag{8}$$

where DT_{kj} is the direct trust node k has towards node j and $0 \leq DT_{kj} \leq 1$. We apply a stricter policy in evaluating the TF_2 of an ID_V in the sense that if an ID_E is not able to find a prompt indirect trust information in its own record to evaluate an ID_V's honesty during an evaluation period, the ID_V will be directly assigned a low trust value and the shared information might be discarded. This is to avoid high risk of getting false recommendation and also reducing delay in processing the information. Given the direct and indirect trust values, we obtain:

$$TF_2 = \omega_d * DT_{ij} + \omega_i * IT_{ij} \tag{9}$$

where $\omega_d + \omega_i = 1$ and $\omega_d > \omega_i$ due to the more reliable nature of direct trust evaluation.

The third trust feature, TF_3, is the most crucial element in computing the total trust value of a recommender node, ID_V, which is obtained from its observation information towards relay node, ID_R. To the best of our knowledge, assessing node's behaviour based on its observation information and making it as an obligation for any recommender node to present the required information has not been done in existing works. In our justification, such requirement could become a foolproof recommendation system in the sense that a node that has been doing real observation (and later willing to share) should be able to present the required information as depicted in Figure 2. However, one of the salient challenges is ensuring the validity of the shared information as a node could falsify them. Hence, in this paper, we propose a cross-check method based on the concept of trust chain which will be discussed in Section 4.4.

In validating recommendation, the very first thing to be evaluated is whether or not the shared information is sufficient according to requirements of the recommendation policy. We divided the sufficiency level, s_v, into four parts: 1) identity information, s_{v1}, 2) Behaviour classification, s_{v2}, 3) forwarding ratio, including r and fm values, s_{v3}, and 4) CMSD timestamp, s_{v4}; each holds a value of 0.25 for a total s_v value of 1, if all information is presented or else, $s_v = 0$. At this stage, we apply a strict policy whereby an ID_V needs to present all information required, otherwise, the shared information will not be evaluated which could lead to node ID_V having low trust value total. This is motivated by the fact that an honest node which has been doing genuine observation has no issue to report the information accordingly. Next is to validate the shared information by using Algorithm 1, and derive the validation value, $v_d = 1.0$ if an ID_V passes both the cheating and node collusion tests; $v_d = 0.5$ if validation is not completed; and $v_d = 0$ if node is cheating. Hence, TF_3 value can be computed as follows:

$$TF_3 = s_v * v_d \tag{10}$$

4.3.2 Assessment of recommended node, ID_R. As shown in module 3, assessment on node ID_R starts right after the trust value of an ID_V has been obtained. If $TV_{ij}(t) \geq 0.5$, then ID_V's recommendation is considered trustworthy and the shared behaviour information will be used to evaluate the indirect trust value IT_R of an ID_R. The indirect trust will be combined with direct trust value DT_R to obtain a single trust value of ID_R. The calculation of direct and indirect trust are based on Eq. (7) and Eq. (8) respectively. Hence, the total trust value (also denotes cooperativeness/selfishness level) of an ID_R is as follows:

$$TR_{ij}(t) = \omega_d * DT_{ij}(t) + \omega_i * IT_{ij}(t) \tag{11}$$

where $\omega_d + \omega_i = 1$ and $\omega_d > \omega_i$.

4.4 Cheating and Node Collusion Detection

In this section, we discuss a distributed approach for the detection of cheating (i.e. false recommendation) and colluding nodes, which can be done by any node in the network without the involvement of a central agent and the associated communication overhead. We then propose a lightweight filtering mechanism that is effective in detecting a singular or colluding cheater.

Detection of a node's cheating action is based on the following three main conditions which become our rationale that realistic cross-checking can be performed:

(i) using CMSD-based local observation, each ID_V will present two pairs of behaviour classifications ($P1$ and $P2$) on an ID_R whereby, the pairs are of ID_R's forwarding efforts towards nodes ID_V and ID_C respectively. Correct reports should follow logical classification orders as shown in Table 2. For example, when an ID_V labels an ID_R as being unfairly cooperative (c, u), it means that the ID_R is being selfish towards ID_C such that unfairly selfish (s, u) classification is expected from ID_C. Any deviation from the list denotes possible cheating occurrence which can be detected either if an ID_V itself presents conflicting pairs or if cross-comparison with another recommender's reported behaviour classifications ($P3$ and $P4$) are also conflicted.

Algorithm 1 Recommendation Validation

```
for each recommendation received do
    Check sufficiency level, $s_v$
    Assign $s_v$
    if $s_v = 1.0$ then
        Create $ID_V$ list = $\{v_1, v_2, v_3, \ldots.., v_n\}$
        Create $ID_R$ list = $\{r_1, r_2, r_3, \ldots.., r_n\}$
        Create $ID_C$ list = $\{c_1, c_2, c_3, \ldots.., c_n\}$
        Sort lists according to timestamp
        Evaluate Algorithm 2
        Obtain $v_d$
    else
        Discard information
        $s_v = 0$; $v_d = 0$
    end if
end for
Calculate $TF_3$
```

Table 2: Ideal Reports

Evaluation of	Behaviour Classification			
ID_R **towards** ID_V	(s, f)	(c, f)	(c, u)	(s, u)
ID_R **towards** ID_C	(s, f)	(c, f)	(s, u)	(c, u)

(ii) CMSD implementation also enables an ID_V to present detailed information about an ID_R as shown in Figure 2, making it hard for the ID_V to falsify information easily. For instance, the ID_V cannot simply present the behaviour classification information without ensuring that r, fm and FR_{ij} values are tallied. Subsequently, colluding action will be affected because nodes need to ensure that their false information does not look replicated.

(iii) the competitor node, ID_C, is the key element which can be cross-compared. For an ID_V to have the ability to observe the packet forwarding rate of an ID_R and the packet sending rate of an ID_C at the same time, signifies that the two nodes are adjacent neighbours of the ID_V and increase the possibility that an ID_C will also be recommending ID_R's behaviour.

Although there are many cheating scenarios can be investigated, we focus on three cases for this paper, as shown in Algorithm 2, which is explained with the help from sample cases presented in Table 3. As shown in the table, each ID_V will provide two behaviour reports based on its parallel observation on forwarding effort of an ID_R towards itself and an ID_C.

For **case 1** and **2**, ID_V and ID_C can be cross-compared due to ID_C also received the recommendation request and will share its observation report towards an ID_R. For this particular case, it is an ideal condition to perform evaluation more accurately. The difference is in the reported behaviour classification, where in **case 2**, the pair does not follow the logical order as listed in Table 2 which signifies suspicious reports. As for **case 1**, the behaviour pairs are symmetrically presented by both ID_V and ID_C, which is a sign of accurate recommendation but having a possibility of node collusion. As suggested in the above-mentioned motivation, to detect node collusion, evaluation towards forwarding details needs to be performed. If the reported values of r, fm and FR_{ij} are totally replicated, nodes are therefore perceived to have been colluding. This is done with the help from the rule we set, that the values need to be represented in two decimal points to enhance replication detection.

At this stage, we assume that nodes do not make smart calibration tricks on the values because the process will cost them communication overhead especially when more than two nodes involved in the collusion. For **case 2**, the reported behaviour pairs are conflicting, which denotes that one of the nodes is possibly cheating (given assumption local observation's accuracy is optimal). To cater this scenario, the evaluating node will check on the reported behaviour classification that an ID_V provides for both requests. Since a node with cheating intention will falsify the information, node ID_4 is the cheater because it does not present the right classification on forwarding behaviour of ID_2 towards ID_5 that is $(c, u)_{ID2,ID5}$, whereas ID_5 is honest in reporting its observation on ID_2's forwarding effort towards ID_4. Last but not least, for **case 3**, ID_C is not an ID_V (and vice versa), so, cross-comparison cannot be done directly. Thus, the evaluator node will check reports of different timestamps from the same recommender to perform evaluation. If the required information is not available, the two nodes will be assigned lower v_d values, which is 0.5 and may be verified in other assessment session. However, the final decision on recommendation honesty still depends on the other trust features to calculate the total trust of an ID_V.

Table 3: Sample Case

Case	ID_V	ID_R	ID_C	Time-stamp	Behaviour pair	Forwarding details
Case 1	ID_1	ID_2	ID_3	TS_1	P1: $(s, f)_{ID2,ID1}$ P2: $(s, f)_{ID2,ID3}$	r=0.30;fm=0.25;FR_{ij}=0.28 r=0.45;fm=0.25;FR_{ij}=0.35
	ID_3	ID_2	ID_1	TS_1	P3: $(s, f)_{ID2,ID3}$ P4: $(s, f)_{ID2,ID1}$	r=0.40;fm=0.25;FR_{ij}=0.33 r=0.25;fm=0.25;FR_{ij}=0.25
Case 2	ID_4	ID_2	ID_5	TS_2	P1: $(s, f)_{ID2,ID4}$ P2: $(s, f)_{ID2,ID5}$	r=0.22;fm=0.25;FR_{ij}=0.23 r=0.35;fm=0.25;FR_{ij}=0.30
	ID_5	ID_2	ID_4	TS_2	P3: $(c, u)_{ID2,ID5}$ P4: $(s, u)_{ID2,ID4}$	r=0.75;fm=0.75;FR_{ij}=0.75 r=0.20;fm=0.50;FR_{ij}=0.35
Case 3	ID_8	ID_2	ID_{11}	TS_3	P1: $(c, f)_{ID2,ID8}$ P2: $(c, f)_{ID2,ID11}$	r=0.80;fm=1.0;FR_{ij}=0.90 r=0.60 ;fm=1.0;FR_{ij}=0.80
	ID_9	ID_2	ID_{10}	TS_3	P1: $(s, f)_{ID2,ID9}$ P2: $(s, f)_{ID2,ID10}$	r=0.20;fm=0.25;FR_{ij}=0.23 r=0.30;fm=0.25;FR_{ij}=0.28

Algorithm 2 Cheating and Node Collusion Detection

```
From Algorithm 1
Evaluate cheating action
Check main condition
if $P1$ unmatched $P2$ || forwarding details unmatched behaviour pair then
    $ID_V$ is a cheater, $v_d = 0$
    Discard information
else
    Check Cases
    if $ID_C$ and $ID_V$ are cross-compared then
        Case 1 or Case 2
    else if $ID_C$ and $ID_V$ are not cross-compared then
        Case 3
    end if
end if
Check Case 1
if $P1$ matched $P4$ && $P2$ matched $P3$ then
    if forwarding details replicated then
        Colluding node, $v_d = 0$
    else if forwarding details not replicated then
        $ID_V$ is honest and no collusion, $v_d = 1.0$
    end if
end if
Check Case 2
if $P1$ unmatched $P4$ && $P2$ unmatched $P3$ then
    Compare selfish/cooperative label
    if $s2 == s3$ || $s4 == s1$ then
        $ID_V$ is honest, $v_d = 1.0$
    else if $s2 != s3$ || $s4 != s1$ then
        $ID_V$ is cheating, $v_d = 0$
    end if
end if
Check Case 3
Cross-compare reports from different timestamps
if available then
    Check Case 1 or Case 2
else
    Assign $v_d = 0.5$
end if
```

5 PERFORMANCE EVALUATION

The performance evaluation of our proposed scheme is divided into three parts: (1) analysis of trust value assignment, (2) analysis of false positive and false negative, and (3) network performance. The majority of our analyses will be based on scenarios in wireless multihop networks with stationary network topology which is applicable to application scenarios such as human-operated static devices in the Internet of Things (IoT) that require trust-management [14], and especially vulnerable when they are part of the cloud [19]. However, for completeness, we also consider simple scenarios involving mobile nodes.

5.1 Simulation Setup

This study utilizes the OMNeT++ simulation tool to evaluate network scenarios that utilize the Ad hoc On-Distance Vector (AODV) routing protocol [20] which has been adopted in commercial mesh networking platforms, like DigiMesh[1]. Our study uses the simulation parameters as shown in Table 4 based on similar settings in the compared scheme [5] and assumes the following:

(i) Each node has a unique and forge-resistant identifier (ID);
(ii) Every node in the network performs the same CMSD-based local observation mechanism and an observer node only monitors one other known traffic to be compared with, at one particular observation time; and
(iii) Adequate information of the observed traffic can be obtained by the observer node using promiscuous mode of listening in a contention-free channel; non-ideal channel is overcome using re-transmission mechanism.

Table 4: Simulation Parameters

Parameters	Value
Network Area	800 m x 800 m
Simulation Time	1000 seconds
Number of Nodes	50
Movement	Stationary & Gauss-Markov
Mobility Speed	10 m/s
Application	CBR
Packet Size (bytes)	512
Channel Capacity	2MB/s
Radio Propagation Range	250 meters
$\omega_1, \omega_2, \omega_3$	0.2, 0.4, 0.4
ω_d, ω_i	0.6, 0.4
Cooperative / Trust Threshold	0.5

5.2 Trust Value Assignment

We first analyze the effectiveness of our proposed scheme in assigning the right trust values to nodes in the presence of false recommendations in a static network topology. Nodes are randomly distributed in the network area and remain static throughout the 1000 seconds simulation run. The threshold value is set to 0.5 whereby a node that shares falsified information will be assigned a trust value below threshold given that detection is accurate. For simplicity, each recommender node holds value of $s_v = 1.0$ which indicates sufficient observation information is available for Algorithm 2 to be invoked. Cold-start communication is assumed for majority of nodes, with an initial trust value of $TV_{ij} = 0.5$ assigned which denotes a level of uncertainty between trust and distrust; this value will remain if a node does not get involved in any recommendation process. This also implies that, no node is blacklisted for being malicious. On the other hand, certain nodes may have varied initial trust values which are gained from previous direct communications with the evaluating node. In this analysis, five recommender nodes, ID_V $(ID : N9, N19, N29, N39, N49)$ have been set to be sharing false behaviour information (via three cases discussed earlier) about several relay nodes, ID_R $(ID : N8, N18, N27, N31, N40, N47, N50)$ which are of fairly cooperative nodes $(FR_{ij} > 0.5)$ being accused of being selfish or unfairly cooperative. The rest of the nodes are honest in providing accurate information of their local observation towards the listed nodes ID_R in which behaviour information is required by the evaluator node, ID_E.

We will see how the false recommendation will affect the reputation of the ID_R with and without the support of the TFE scheme, which is measured based on the final trust values assigned. Without utilizing the TFE scheme, the selfishness level of an ID_R is measured based on average forwarding ratio provided by recommenders with no mechanism to filter the shared information [5]. Figure 3 shows the trust values obtained by each node which is represented by y-axis, while x-axis represents node ID. With the use of the TFE scheme, nodes with false recommendation are detected and punished by decreasing their trust values to below threshold, except for one mis-detection on node $N29$ which maybe be caused by insufficient information to validate its false recommendation towards an ID_R. Subsequently, due to the mis-detection that causes false information being accepted, the trust value of cooperative node $N27$ is slightly below threshold. However, we anticipate that with more round of assessments, the actual behaviour of node $N29$ can be detected. On the other hand, without the TFE scheme, the dishonest nodes continue to persist while good cooperative nodes are wrongly accused and assigned trust values that are below threshold.

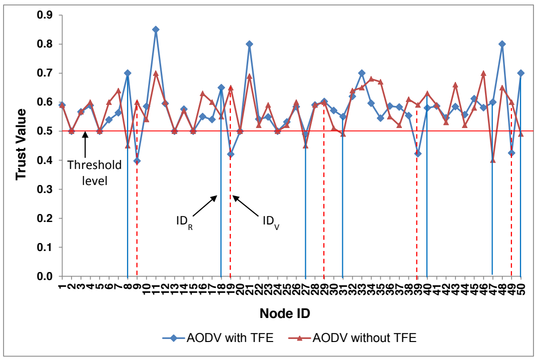

Figure 3: Trust Value Assignment (Static Topology)

Using the same scenario, we extended the evaluation to cater for a dynamic network topology, where nodes move randomly in

[1] https://www.digi.com/products/digimesh

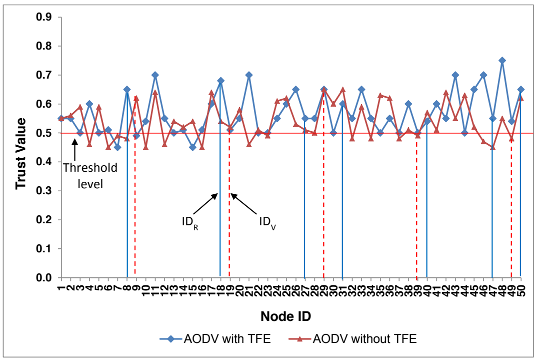

Figure 4: Trust Value Assignment (Dynamic Topology)

the network area based on the Gauss-Markov mobility model [16] which have been widely used in ad hoc networks simulation due to its highly realistic nature. Using the setting of $\alpha = 0$, nodes move in totally random pattern with speed of 10 m/s and we aim to study how node mobility can affect the behaviour detection and eventually the trust values assignment. Figure 4 shows that cheating behaviour can still be detected but the trust values for some cheaters are slightly below threshold which could be manipulated by these nodes using intermittent tactic between honest and dishonest behaviours. Node mobility aggravated the lack of sufficient information to perform accurate cross-checking to identify cheating or illegitimate collusion such that the value of v_d is set to 0.5 instead of 0 to definitively point out a cheating node, and consequently, increases the calculation of overall trust value for bad nodes. On the other hand, the trust values for majority of honest nodes are decreased which can be anticipated due to v_d's value being set to 0.5, only this time, instead of 1. Thus, it can be said that for high node mobility scenarios a different kind of detection mechanism is required and will be investigated as part of our ongoing research.

5.3 Detection Accuracy

The scheme is further evaluated based on the following detection accuracy metrics:

(i) rate of false positive - number of honest recommendation falsely identified as dishonest;
(ii) false negative - number of false recommendation wrongly identified as honest; and
(iii) true detection - number of false recommendation correctly identified;

For each metric, we compare between with and without TFE implementation. Using the mixed of three cases discussed above, we set various number of up to 25 nodes (out of 50) to be sharing false information about 10 good nodes ID_R, each of which holds initial value of $FR_{ij} = 0.7$, indicating good cooperativeness level and fluctuations in FR_{ij} values become the indication of evaluation. Every time these cheaters receive behaviour recommendation requests from any ID_E, they will accuse the nodes ID_R as being selfish with $FR_{ij} < 0.5$ and low values of its associated r and fm.

Figure 5 presents the ratio of false positive detected to the total number of recommendations over increasing number of cheaters in the network, with and without TFE implementation. The false positive rate for both implementations are increasing with the increased number of cheaters present in the network. This is because, as the number of cheaters increases, some honest recommendations cannot be fully validated using the cross-checking algorithm and would require longer detection time to be proven honest by doing repetitive comparison with reports from different time-stamp sessions. However, by having TFE scheme incorporated, the false positive is increasing at a significantly lower rate compared to the one without any filtering mechanism and have to rely heavily on majority of recommendations method.

A similar pattern is also observed in the false negative rate, where the number of false recommendations incorrectly identified as honest are increasing over higher number of cheaters as depicted in Figure 6. The slight difference is that the overall result for scheme without TFE implementation is affected with higher percentage of false negative. This can be attributed to the fact that the more dishonest nodes present in the network, the higher potential of illegitimate collusion occurring such that lower number of honest recommendations cannot convince the system of their honesty. In contrary to the scheme with TFE filtering implementation, cheaters cannot simply collude with one another without a good strategy that is hard to detect.

As aforementioned, in order for nodes to collude effectively, the colluding nodes need to communicate with one another in corroborating the falsified data. In addition, to ensure larger number of colluding nodes, these nodes need to be mobile so that they can collude and exchange false information with larger number of nodes. But since they are static, even the slightest deviations would eventually be detected because the same cheating pattern can be recognized after performing several other cross-checking sessions.

Next, we evaluate the true detection rate which is the percentage of the number of correctly identified false recommendations to the total number of actual false recommendations shared in the network. Figure 7 shows that the rate of true detection of TFE implementation is reasonably high despite reducing as the number of cheaters are increasing. The ability to achieve the true detection rate of above 80% shows that the TFE scheme is robust against the large number of cheaters, unlike the case without any filtering mechanism.

The analysis is further supported by detection rate of false recommendation on the three different cases (cf: Table 3) considered in this paper which can be labeled as:

(i) Case 1 - cheating and colluding threats;
(ii) Case 2 - tricky cheating; and
(iii) Case 3 - insufficient evidence.

The purpose of this analysis is to evaluate which case would take longer time to be identified. We study the scenario of having 25 cheater nodes in the network propagating false recommendations about 10 relay nodes and measure the rate based on the total false recommendations obtained at several different time-stamps. As shown in Figure 8, Case 1 is the fastest to be detected as it consists of information that is sufficient to evaluate cheating behaviour more accurately. Although towards the end of the simulation run the true detection rate does not reach 100% due to some mis-detection, the final rate of 91% is significantly higher than the other two cases.

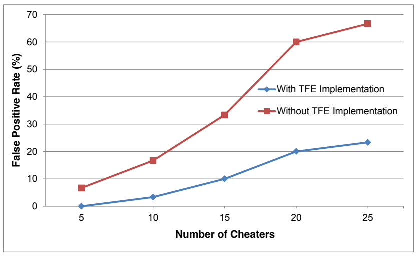

Figure 5: False Positive Rate

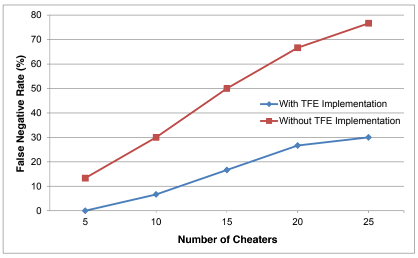

Figure 6: False Negative Rate

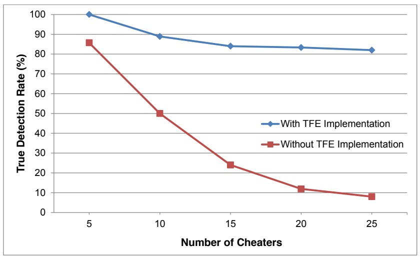

Figure 7: True Detection Rate

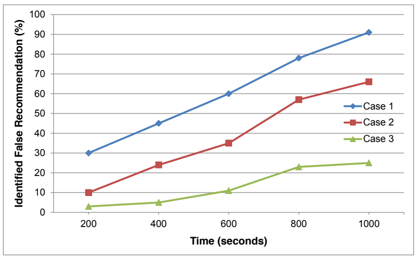

Figure 8: Detection Rate of Cases

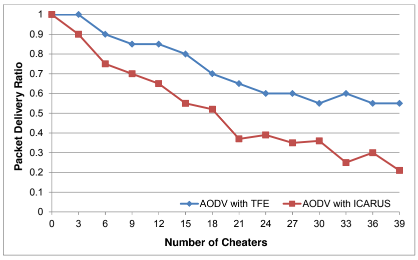

Figure 9: Packet Delivery Ratio

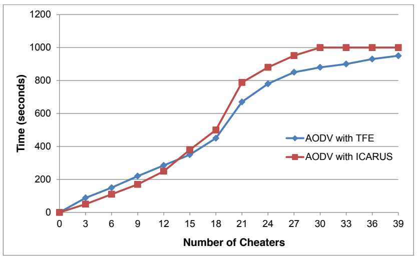

Figure 10: Average Detection Delay

Lower detection rate is found in the Case 2 scenario which is attributed to unmatched behaviour pairs such that cross-checking needs to be further extended on more detailed information which requires longer time to complete. Last but not least, Case 3 is the slowest to be detected due to the need to collect and cross-check information with different time-stamps, and faced with the possibility of not getting the required information. The delay scenarios are among the trade-offs of having CMSD-based cross-check detection such that rigorous investigation on the information that is required to be presented during recommendation phase would cause delay in certain scenarios. However, in our justification, this is an acceptable trade-off because the high complexity of the behaviour information presented would require elaborate strategies to corroborate the falsified data and makes the scheme less prone to threats, such as, bad-mouthing and large node collusion.

5.4 Network Performance

This section presents evaluation of TFE scheme with regard to network performance in terms of packet delivery ratio (PDR) and how fast cheating nodes are detected in comparison to an existing scheme, ICARUS [5]. ICARUS is a scheme which detects selfish/cheating nodes with assistance from a central agent. ICARUS has been selected as a benchmark scheme to compare against because of its centralized architecture and behaviour detection based on average forwarding ratio (AVR) in the network. When it comes to processing recommendation using the AVR method, an evaluated node is prone to bias accusation especially when a group of colluding nodes collaborate in reporting low forwarding ratio causing drop-off on the node's AVR value prior to being labeled as a bad node. In ICARUS, the mechanism provided to overcome this issue is by having a central agent aggregate information from all nodes; however, there is no concrete guarantee that all nodes would submit behaviour reports on a particular node and by rejecting reports from any node already marked as selfish. However, as mentioned in the paper, the scheme is likely prone to large number of cheaters ($> 50\%$). In addition, the paper did not specifically discuss how cheating action is detected, and rather used cheating probability from 0 to 1 to reflect the increasing level of packet rejection. Despite this condition giving the scheme an advantage of faster detection, they are actually lacking in accurate detection of cheating behaviour, yet dealing with a central agent is highly time consuming.

Besides the architectural difference between ICARUS and our TFE approach, the other key difference is the way behaviour information presented during the recommendation phase. For ICARUS, the only information shared about a particular observed relay node is its forwarding ratio value, whereas for our scheme, we use the Case 1 scenario in the implementation for simplicity which also reflects rather fair comparison considering simpler cheating scenario adopted in ICARUS. We are going to show that even our proposed scheme is fully distributive, yet detection is still effective such that network performance is comparable, if not better.

As shown in Figure 9, with the increasing number of cheating nodes in the network (the selected number is similar to ICARUS' setting), the PDR values for both parameters are decreasing, which is anticipated and requires more time to sustain the PDR at higher rate. However, our scheme is proven to be more resilient than ICARUS by maintaining higher rate of PDR. This statement can be supported by the fact that with the existence of large number of cheaters, more honest nodes are being falsely accused as selfish and getting unnecessary punishment which eventually lowers the PDR value. Our scheme has able to prevent these good nodes from getting such treatment, and thus, obtaining higher PDR.

The analysis is further extended to cater for the delay imposed towards the schemes based on the average detection time until the last false recommendation is identified under varied number of cheaters. Figure 10 shows that, in the presence of lesser cheating nodes, ICARUS implementation outperforms TFE in detecting false recommendation faster. This is because during the earlier phase, the number of honest nodes are greater than the dishonest ones, which accumulate majority of positive votes. Hence, the minority recommendation of low average forwarding ratio of a particular relay node is regarded as falsified.

However, as the number of cheaters increased, contrary scenarios start to emerge whereby TFE outperformed ICARUS by requiring lesser time to detect false recommendation with exponential increment. For ICARUS, the time growth becomes saturated upon reaching 30 cheater nodes which indicate either longer simulation time is required to detect more false recommendations or the scheme is no longer able to perform correct detection furthermore. It is worth to mention that both schemes have communication and computational processes which cause delays such that the overall time difference between the two schemes is not large. However, our TFE scheme is clearly having better performance with higher detection accuracy of false recommendation.

6 CONCLUSION AND FUTURE WORK

In this paper, we have proposed a distributive scheme to filter false recommendations during global observation session which is developed based on trust features. A significant distinct element in our trust-based scheme compared to the existing ones is that part of the trust features is formulated based on information that is extracted using a proven local observation mechanism, named CMSD. Using the extracted information that has been made compulsory to be presented by recommender nodes, and given that certain metrics are available, we are able to compute node's trust level to detect false reports, cheaters and colluding nodes to achieve significant network performance. Nonetheless, there are several potential extensions to this study.

Firstly, other than using the CMSD mechanism, adoption of a more generalised local observation technique and an alternative way to assess the reported behaviour can be considered. This would make TFE a more flexible scheme which could identify false recommendation in the event of diversified local observation techniques and shared information presented by recommenders. Secondly, this paper only considers three types of scenarios of node collusion and able to detect them. In reality, there are various more possible cases of cheating and node collusion actions which can be investigated further to test the robustness of this scheme.

Last but not least, there are several costs involved in our proposed scheme as trade-offs in performing accurate evaluation such as node's computational cost in performing cross-checking on incoming reports, memory usage to store other nodes' behaviour information and energy usage to perform complete procedures as required by the TFE modules. Despite having certain costs reduced by such as implementing an on-demand packet broadcast in the network instead of all-time network-wide flooding and setting expiration time for information storage, we however, aim to reduce these costs further down as part of our future research.

REFERENCES

[1] M. Abolhasan, J. Lipman, W. Ni, and B. Hagelstein. 2015. Software-defined wireless networking: centralized, distributed, or hybrid? *IEEE Network* 29, 4 (July 2015), 32–38.

[2] A. Ahmed, K. Abu Bakar, M. I. Channa, K. Haseeb, and A. W. Khan. 2016. A trust aware routing protocol for energy constrained wireless sensor network. *Telecommunication Systems* 61, 1 (2016), 123–140.

[3] S. Buchegger and J.-Y. L. Boudec. 2002. Performance analysis of the CONFIDANT protocol. In *Proc. of the 3rd ACM International Symposium on Mobile Ad hoc Networking and Computing (MOBIHOC).* Lausanne, Switzerland, 226–236.

[4] S. Buchegger and J.-Y. LeBoudec. 2003. The effect of rumor spreading in reputation systems for mobile ad hoc networks. In *Proc. of the Workshop on Modeling and Optimization in Mobile, Ad hoc and Wireless Networks (WiOpt).* INRIA Sophia-Antipolis, France, 1–10.

[5] D. E. Charilas, K. D. Georgilakis, and A. D. Panagopoulos. 2012. ICARUS: hybrId inCentive mechAnism for coopeRation stimUlation in ad hoc networkS. *Ad Hoc Networks* 10, 6 (2012), 976–989.

[6] J. H. Cho, A. Swami, and I. R. Chen. 2012. Modeling and analysis of trust management with trust chain optimization in mobile ad hoc networks. *Journal of Network and Computer Applications* 35, 3 (2012), 1001–1012.

[7] J. H. Cho, A. Swami, and I. R. Chen. Fourth Quarter 2011. A survey on trust management for mobile ad hoc networks. *IEEE Communications Surveys & Tutorials* 13, 4 (Fourth Quarter 2011), 562–583.

[8] Z.-K. Chong, S.-W. Tan, B.-M. Goi, and B.C.-K. Ng. 2013. Outwitting smart selfish nodes in wireless mesh networks. *International Journal of Communication Systems* 26, 9 (2013), 1163–1175.

[9] M. Conti, E. Gregori, and G. Maselli. 2006. Reliable and efficient forwarding in ad hoc networks. *Ad Hoc Networks* 4, 3 (2006), 398–415.

[10] T. Eissa, S. A. Razak, R. Khokhar, and N. Samian. 2011. Trust-based routing mechanism in MANET: design and implementation. *Mobile Networks and Applications* 18, 5 (2011), 666–677.

[11] L. H. G. Ferraz, P. B. Velloso, and O. C. M. B. Duarte. 2014. An accurate and precise malicious node exclusion mechanism for ad hoc networks. *Ad Hoc Networks* 19 (2014), 142–155.

[12] K. Govindan and P. Mohapatra. Second Quarter 2012. Trust computations and trust dynamics in mobile adhoc networks: A survey. *IEEE Communications Surveys & Tutorials* 14, 2 (Second Quarter 2012), 279–298.

[13] Q. He, D. Wu, and P. Khosla. 2004. SORI: A secure and objective reputation based incentive scheme for ad hoc networks. In *Proc. of IEEE Wireless Communications and Networking Conference (WCNC).* New Orleans, LA, USA, 825–830.

[14] I. Kounelis, G. Baldini, R. Neisse, G. Steri, M. Tallacchini, and A. Guimaraes Pereira. 2014. Building Trust in the Human Internet of Things Relationship. *IEEE Technology and Society Magazine* 33, 4 (winter 2014), 73–80.

[15] H. Li and M. Singhal. 2007. Trust management in distributed systems. *Computers* 40, 2 (2007), 45–53.

[16] B. Liang and Z. J. Haas. 1999. Predictive distance-based mobility management for PCS networks. In *Proc. of the Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM).* New York, USA, 1377–1384.

[17] Z. Liu, A. W. Joy, and R. A. Thompson. 2004. A dynamic trust model for mobile ad hoc networks. In *Proc. of the 10th IEEE International Workshop on Future Trends of Distributed Computing Systems (FTDCS).* Sushou, China, 80–85.

[18] P. Michiardi and R. Molva. 2002. CORE: A collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks. In *Proc. of the 6th Joint Working Conf. on Comms. and Multimedia Security.* Netherlands, 107–121.

[19] D. Minoli, K. Sohraby, and B. Occhiogrosso. 2017. IoT Considerations, Requirements, and Architectures for Smart Buildings Energy Optimization and Next-Generation Building Management Systems. *IEEE Internet of Things Journal* 4, 1 (Feb 2017), 269–283.

[20] C. Perkins, E. Belding-Royer, and S. Das. 2003. *Ad hoc On-Demand Distance Vector (AODV) Routing.* RFC 3561. Internet Engineering Task Force. https://www.ietf.org/rfc/rfc3561.txt

[21] N. Samian, W. K. G. Seah, and G. Chen. 2013. Quantifying selfishness and fairness in wireless multihop networks. In *Proc. of the 38th Annual IEEE Conference on Local Computer Networks (LCN).* Sydney, Australia, 270–278.

[22] N. Samian, Z. A. Zukarnain, A. Abdullah, and Z. M. Hanapi. 2015. Compare and measure selfishness detection (CMSD) mechanism: Promptness and accuracy. In *Proc. of the Seventh International Conference on Ubiquitous and Future Networks (ICUFN).* Sapporo, Japan, 851–856.

[23] A. M. Shabut, K. P. Dahal, S. K. Bista, and I. U. Awan. 2015. Recommendation Based Trust Model with an Effective Defence Scheme for MANETs. *IEEE Trans. on Mobile Computing* 14, 10 (2015), 2101–2115.

[24] P. B. Velloso, R. P. Laufer, D. D. O. O. Cunha, O. C. M. B. Duarte, and G. Pujolle. 2010. Trust management in mobile ad hoc networks using a scalable maturity-based model. *IEEE Trans. on Network and Service Management* 7, 3 (2010), 172–185.

[25] H. Yu, S. Liu, A. C. Kot, C. Miao, and C. Leung. 2011. Dynamic witness selection for trustworthy distributed cooperative sensing in cognitive radio networks. In *IEEE 13th Int'l Conf. on Communication Technology (ICCT).* Jinan, China, 1–6.