# Trusted Spanning Tree for Delay Tolerant MANETs

A. Piyatumrong, P. Bouvry
Université du Luxembourg,
FSTC - CSC, Luxembourg
apivadee.piyatumrong@uni.lu,
pascal.bouvry@uni.lu

F. Guinand
Le Havre University,
LITIS, Le Havre, France
Frederic.Guinand
@univ-lehavre.fr

K. Lavagnananda
King Mongkut's University-
of Technology Thonburi,
School of Information Technology,
Bangkok, Thailand
kitt@sit.kmutt.ac.th

## Abstract

*Quality of service is an important issue in Delay Tolerant MANETs. This work aims at increasing the QoS in such networks by relying on spanning forests (DA-GRS). The existing algorithms are improved by introducing the notion of trust into the spanning forest and choosing the most robust (trustable) spanning trees among existing opportunities. The robustness/quality of the tree can be assessed based on two cost functions. Two heuristics are proposed 'G-TRUST' (a greedy-based heuristic) and 'G-TRUST BREAK' (including an automatic tree reconfiguration heuristic) and evaluated.*

## 1. Introduction

Delay Tolerant Mobile Ad Hoc MANETs (DTMs) are fluctuating networks populated by a set of moving nodes equipped with wireless communicating devices. They can spontaneously interconnect each other without any pre-existing infrastructure [1]. What makes the management of such networks difficult is their nature. DTMs are mobile, ad hoc configuring, and frequently partitioned.

The most popular wireless networking technologies available nowadays for building DTMs are Bluetooth and IEEE802.11 (WiFi). This implies that devices communicate within a limited range, and stations may move while communicating. A consequence of mobility is that the topology of such networks may change quickly and unpredictably. This dynamical characteristic constitutes one of the main obstacles for performing efficient communications. Furthermore, acquiring global information in this kind of network is difficult and impractical if not impossible. Therefore management information within this network needs to be done locally, but yet effective globally. Then, algorithms designed for DTMs have to be self-configuring, decentralized and robust to cope with both, the dynamic and the partitioned nature of the environment.

Ad hoc networks rely on 'cooperation' of a set of nodes in order to emerge and operate the network. Cooperative enforcement approach has been proposed by a number of researcher to enhance the robustness, the availability and/or the overall throughput [2] in pure Mobile Ad Hoc Networks. The main objective of those works is to cope with 'selfish node' whose deteriorate the robustness of the network by given poor collaborative efforts (e.g., not forward packet of others). In this paradigm, the term of 'trust' and 'reputation' are used as judgement values representing the cooperative level or trust level toward other stations in the community. The evaluation of trust has been proposed in many works [3], [4], [5], etc. This work proposes to use trust information to strengthen the spanning forest. We propose here some heuristics, to enhance the quality of trusted spanning tree.

The main goal of this work is to enhance the QoS running on DTMs. We introduce therefore the notion of trust into the existing algorithm used in Dynamicity Aware - Graph Relabeling System (DA-GRS) [6]. The main problem addressed in this paper is how robust trusted spanning trees (management structures) can be created in such a dynamic, decentralized and experiencing non-permanent connection manners of DTMs. G-TRUST, a simple greedy algorithm is proposed. This heuristic has been further enhanced and becomes G-TRUST BREAK which does also perform automatic tree reconfiguration. Different cost functions are proposed in this study in order to assess the robustness of the trees. It is assumed that trust information on nodes is known.

In the next section, state of the arts are presented. Following Section gives the details of the proposed model, the definition of being robust trusted spanning tree, and also the cost functions. In Section 4, all studied algorithms has been described. The experiments and results are shown in Section 5. Finally, the work is concluded and gives some future works in the Conclusions and Perspectives Section.

# 2. Related Works

## 2.1. Delay Tolerant Mobile Ad Hoc Networks (DTMs)

DTMs constitute an emerging subclass of mobile ad hoc networks (MANETs) that feature frequent and long-duration partitions [7]. With these properties, DTMs have problem in both disseminating and acquiring information, meanwhile the primary requirement of DTMs is that information are reliably delivered. For this study, the quality of service which each node can provide is the main criteria using in evaluation of trust. In a DTM network, each station can reach a subset of the other stations using wireless communication abilities. Such communication ability is typically defined by a communication range and constrained by natural obstacles (e.g. walls, buildings, etc.).

At a given moment $t$, the communication graph, $G(t)$, of such network is a pair $(V_t(G), E_t(G))$, where $V_t(G)$ is a finite set of elements, called vertices, $E_t(G)$ is a binary relation on $V_t(G)$ - a subset of pairs of elements of $V_t(G)$. The elements of $E_t(G)$ are called edges and constitute the edge set of $G(t)$. An edge between node $x_i$ and $x_j$ indicates that, at time $t$, it is possible for $x_i$ and $x_j$ to exchange information. Let's $P(t)$ is a subgraph at moment $t$, $G(t)$ may be partitioned into a set of $m$ subgraphs: $G(t) = \bigcup_{1..m} P_i(t)$ where $i \in \{1..m\}$.

## 2.2. Trust Management

In human society, trust has become the basis of almost all activities, such as communications, work, etc. People gradually form the standard of mutual trust, and they also refer to opinions of the third-party in assessing the trust. Trust can be regarded as a criterion for making a judgment under complex social conditions and can be used to guide further actions [8]. In summary, trust can be viewed as the expectation or the belief that a party will act kindly and cooperatively with the trusting party [9]. It is no surprise that some research related to security or mutual cooperation on multi-agent system paid particular attention to trust factor in various facets, [10], [11], [12].

In early stage of trust and security on MANETs, several trust and security establishments relied on cryptographic methods, authentication codes and hashing chains for their solutions. Although these schemes are effective, they are centralized system which are not applicable to DTMs (because of the dynamic movement of nodes and also lacking of pre-existing infrastructure). In the last few years, cooperation enforcement methods (avoidance the effect of selfish nodes on the networks' robustness [13]), and reputation schemes [9], [14], [15] have been proposed for trust establishment in MANETs. Recent literature suggests that the cooperation enforcement techniques are more appropriate if the primary goals are availability, robustness of the network, and the overall throughput. A comprehensive survey on cooperation enforcement can be found in [2], while detailed discussion on peer-to-peer key and trust management approach can be found in [16].

## 2.3. Dynamicity Aware - Graph Relabeling System (DA-GRS)

DA-GRS [17] is a model invented for design and analysis of decentralized applications and algorithms targeting dynamically distributed environments like DTMs. Normally, such applications and algorithms are often very difficult to set up, describe and validate. Using DA-GRS is a convenient way to design algorithms for DTMs, since its outstanding properties are localized in a dynamic working manner. In the context of the study, DA-GRS approach proposes a way of designing a decentralized algorithm for constructing and maintaining a spanning forest in DTMs, relying on a careful rule-based token management. Hence forth this concept will be referred to as 'DA-GRS' for brevity. The work in [18] described rules to handle four different scenarios, (a) tokens traversal in general case, (b) when a token meets another token, (c) partition occurs at a node which belongs to the spanning tree that possess the token, (d) partition occurs at a node which belongs to the spanning tree which does not possess the token. These rules can be viewed in DA-GRS's visual representation illustrated by Figure 1.

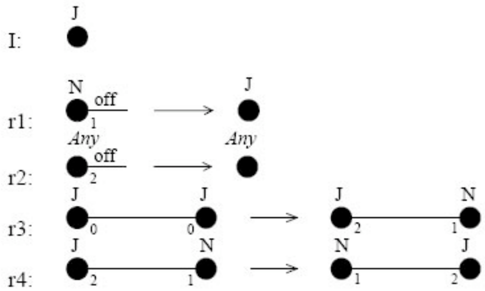

**Figure 1. Spanning forest algorithm (visual representation)**

From rules of DA-GRS, at one moment in time, there are only two tokens which can meet and merge.

# 3. Trusted Spanning Tree Model

## 3.1. Trusted Spanning Tree

In this work, QoS is enhanced by managing the notion of trust inside spanning tree. It is assumed there is no malicious node in the current work and that the trust value for nodes are known.

As DA-GRS constructs random spanning trees, quality (in term of trust) of each spanning tree ought to be assessed.

Assuming that we have a cost function $Q()$ which can evaluate the quality of trees. Let $\Gamma_i$ be the set of all possible spanning trees for $P_i$ and $Q(\gamma)$ is the quality of spanning tree $\gamma$. DA-GRS randomly selects $\gamma^{dagrs} \in \Gamma_i$. An ideal situation is to be able to select $\gamma^{optimal} \in \Gamma_i$ , or at least to select preferable $\gamma^*$ such that $Q(\gamma^{dagrs}) \leq Q(\gamma^*) \leq Q(\gamma^{optimal})$

A study of two potential candidates for such cost functions is proposed in subsection 3.3.

## 3.2. Synchronization Method

DA-GRS uses rendez-vous as the synchronization process (synchronization is made when token meet another token). This rendez-vous assumption states that at one moment in time, only two tokens can meet and merge. We propose to relax this rendez-vous assumption, allowing one master node to discover and meet potentially more than one neighbor. In order to discover more information and new members (sub-tree), the master node holding token will broadcast a packet asking if its neighbors also have token. The neighbor nodes of this master node will reply only if it has token and it is possible for having many nodes holding token at the same time. Hence, the master node can select a node to merge with from this pool of neighbor having token.

## 3.3. Robust Trusted Spanning Tree

Trust level of a node $n$, denoted by $trust(n)$, where $trust(n) \in Z^+$, defines the levels of quality of services it can provide. Whether a node $n$ can be trusted is determined by a given threshold. Let $\Theta_t = \{n' \in V_t(G) | trust(n') \leq threshold\}$ be the set of all low-trustable nodes at moment $t$. A node in a cooperative network can have low level of trust for various reasons such as low battery, poor communication signal, moving out of communication range, etc. An ideal situation is to determine an optimal trusted spanning tree among many possibilities in a given cooperative network. To date, there is no efficient algorithm which can generate an optimal spanning forest in DTMs due to their dynamic, decentralized characteristics and lack of global knowledge in such networks.

In order to determine robust trusted spanning trees, this work introduces quality measurement for trusted spanning trees by means of two cost functions. These are $weight()$ and $isolateLowTrustedNode()$. The meaning of what are robust trusted spanning trees are described below with each cost functions which are supplementary to each other. In order to summarize the quality of the created trust spanning tree, the value of functions from different studied algorithms will be compared where a higher value indicates a better quality. Figure 2(a) and (b) are examples to illustrate the idea behind these cost functions where the threshold used in this example is equal to one.

### 3.3.1 weight() function

In general case, nodes with higher trust level are more likely to be able to complete their tasks than lower ones. $weight()$ function introduced here, can be used to assess trust spanning trees with respect to this objective. Having $V(\gamma)$ as the set of all node in a spanning tree $\gamma$, the $weight()$ function of a trusted spanning tree can be determined by the following equation:

$$weight(\gamma) = \sum_{x \in V(\gamma)} trust(x) * tree\_degree(x) \qquad (1)$$

The function $tree\_degree(x)$ represents the number of direct tree neighbors, one hop neighbors whom node $x$ connected with using a tree edge. Figure 1 is used to illustrate how the $weight$ function can assess this quality. In Figure 2(a), the node with lowest trust level gets the highest tree_degree, while the node with highest level gets the lowest tree_degree (i.e the node A has a trust level of 1 and tree_degree of 3, while the node E has a trust level of 5 and tree_degree of 1), hence the $weight()$ function for this trusted spanning tree is 22. Figure 2(b) depicts the opposite (i.e. the node with highest trust level possess highest tree_degree (node E), while the node with lowest level possess lowest tree_degree (node E)). The $weight()$ function for this trusted spanning tree is 34.

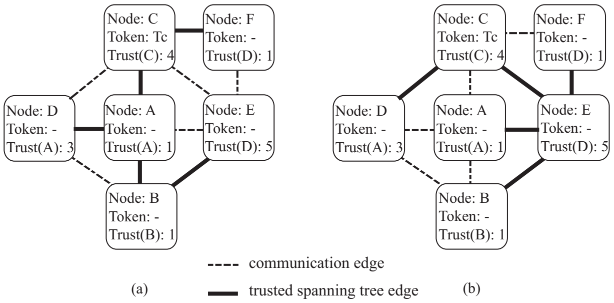

**Figure 2. An example scenario for illustrating cost functions used in this study**

Having nodes with low trust levels localized on leaves is advantageous since they would not be responsible for forwarding information to others. Furthermore, loosing them at these positions has little effect on the overall structure. On the contrary, as low trust level nodes have tendency to break away from the network, allowing them to have high degrees presents a difficult task of re-connecting the trusted spanning trees as a result of their breaking away. Therefore, in order to minimize the re-connecting task, nodes with lowest trust levels should be assigned the lowest tree_degree position in the trees. The next functions $isolateLowTrustedNode()$ is introduced as a mean to assess trusted spanning trees with respect to the objective.

### 3.3.2 isolateLowTrustedNode() function

This function indicates the efficiency of a trusted spanning tree by noting how well it can isolate non-trustable nodes. The function measures the percentile of $n'$ nodes at terminal position. The higher value of $isolateLowTrustedNode()$ function signifies better quality trusted spanning tree. Let $\Theta^*(\gamma) = \{n' \in \Theta(\gamma) | n'$ is at terminal position of $\gamma\}$ be the set of low trustable nodes being at leave position in the tree $\gamma$. The $isolateLowTrustedNode()$ function can be determined by the following equation :

$$isolateLowTrustedNode(\gamma) = \left( \frac{| \Theta^*(\gamma) |}{| \Theta(\gamma) |} \right) * 100 \quad (2)$$

Hence, the *isolateLowTrustedNode* value for Figure 1(a) is 33.33% while this value is 100% for Figure 1(b).

## 4. Trusted Spanning Trees Heuristics

### 4.1. DA-GRS

DA-GRS is used to create a lower bound of the created spanning tree in terms of trust. This lower bound may be used to make comparison with other study algorithms for further improvement purposes.

DA-GRS remains thus untouched and the resulting trees are evaluated by the proposed cost functions.. Each token moves within their own trusted spanning tree. The merging operation of two trusted spanning trees occurs when two tokens meet, using random method for selection a node to merge with (if there exist more than one acknowledgment from neighbors). After the merge is complete, a new and larger trusted spanning tree is formed, and the two tokens also merge information into one unique token .

### 4.2. Greedily Trusted Spanning Tree (G-TRUST)

G-TRUST is an extension of DA-GRS incorporating the notion of trust and using the relaxation of DA-GRS's rendez-vous. Thus, in G-TRUST, several tokens can meet simultaneously. In G-TRUST, the main point of making decision is when several tokens meet, so the master token is able to choose the most trusted node to merge.

The merging operation in G-TRUST is described below.

**Algorithm 1 Look for other trees (tokens) around token $\tau_i$**

```
$\tau^{best}$ is the most trusted token in one hop neighbourhood
if $\tau^{best} \neq \emptyset$ then
    $Merge\_With(\tau_i, \tau^{best})$ //merge the two tokens
else
    $Move\_Token(\tau_i)$ //continue to move the token randomly
end if
```

Figure 3 illustrates this improvement. In this instance, merging of two trusted spanning trees, in Figure 3(b), occurs where tokens are at nodes with highest trust level resulting in a larger and more robust trusted spanning tree than in Figure 3(a).

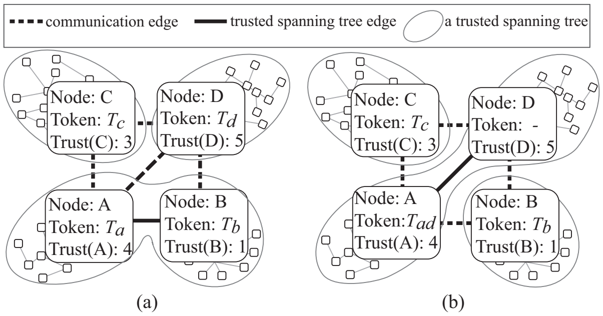

**Figure 3. An example merging using Trust DA-GRS(a) and G-TRUST(b)**

### 4.3. G-TRUST BREAK

'$BREAK$' heuristic aims at providing the opportunity to reorganized the established trees. The tree reorganization might be needed because of the fluctuation in both the communication topology and node trust values which affects the robustness of the tree itself. Hence, G-TRUST BREAK is an extension of G-TRUST enabling dynamically the trusted tree reorganization. Nodes having low trust values will provoke the reorganization. Hence, when it arrives the time to check ($T\_break$) the trustability of itself, if the trust value of a node is equal or lower than the $threshold$ of being low-trust node then it apply 'BREAK' automatically. The low-trust node will make decision which tree edge will not be break and break all the other tree edges. This remaining edge simply is the tree edge which connect the low-trust node with the highest trusted neighbor (having highest trust level among all tree neighbors). Since the breaking action is one of the rules in G-TRUST algorithm, the procedure after breaking will automatically resume the classical behavior of G-TRUST.

## 5. Experiment and Result

### 5.1. Description of the experiment

Suitable networks for simulation of any DTMs ought to comprise lay-out of nodes (e.g. citizens), environmental properties and radio propagation (communication link) which reflect real-world situations. The networks used in this work were generated by Madhoc [19], an ad-hoc networks simulator that provides mobility models allowing realistic motion of citizens in variety of environments. Two real-world mobility models, 'shopping mall' and 'highway', are used in this simulations using the parameters as found in Table 1. In Table 1 also shows the properties of 'random waypoint' mobility model which is a commonly used synthetic model for mobility.

The result showing in the next section can be separated into two experimentations. The first experiment works on finding number of times where they are more than two tokens meeting at the same time. They are all three different mobility models used in this work. In order to gain the result, 25 runs has been done and the final value is retrieved by an average among these run.

The second experiment gives the comparison among 5 studied algorithms in the two real-world mobility models. To summarized, the studied algorithms are DA-GRS (as a lower bound), G-TRUST, and G-TRUST BREAK with different $T\_break$ at every 1.25, 2.5, and 5 seconds. The simulation is done using 100 runs per algorithm per mobility model. Since the duration of every simulation is 10 second, the result value of each run is the average result occurred within the simulation time. Hence, the final result shown in the next section is the average result value from those 100 runs.

All the result comes from one biggest connected component of the network used in the study (within each network it is possible to have one or more than one connected component as the network mobile dynamically). The average number of nodes in such connected component is 37.85 nodes in highway and 76.95 nodes in shopping mall network model.

This work assumes 5 different level of $trust(n)$ where $trust(n)$ equal to 1 is the lowest trust level and the $threshold$ value which is used to determine the trustability of any node is equal to one (any node have trust level equal to one is a low-trustable node).

### 5.2. Result

Experimentation has been done to collect the number of times when there are more than two tokens meeting together. The result of 25 runs is reported in Table 2.

**Table 1. Parameterization used in Madhoc**

| | Random Waypoint | Shopping Mall | High way |
|---|---|---|---|
| Surface ($km^2$) | 0.32 | 0.32 | 1.0 |
| Node Density (per $km^2$) | 1000 | 1000 | 80 |
| Number of Nodes | 100 | 100 | 80 |
| Avg. Number of Partitions | 1 | 2.68 | 1.7 |
| Number of Connections | 541.60 | 389 | 405 |
| Average Degrees | 10.87 | 7.82 | 10.17 |
| Velocity of Nodes ($m/s^{-1}$) | 0.3-1 | 0.3-3 | 20-40 |

**Table 2. the number of times they are more than two tokens meeting at the same time**

| | Random Waypoint | Shopping Mall | High way |
|---|---|---|---|
| Average of Meeting Times | 0.96 | 0.6 | 19.24 |
| Average Number of Tokens per Meeting Times | 4.42 | 3.25 | 41.2 |

Referring to the result shown in Table 2, it suggests that $(a)$ the proposed rendez-vous relaxation of DA-GRS makes sense and enable an intelligent selection of the nodes and $(b)$ creating trusted spanning tree algorithms can utilize either random or other approach in selecting tokens to merge with.

Figure 5 and 4 illustrate results from 'highway' mobility model while figure 7 and 6 give results from 'shopping mall' mobility model. On each graph, details of each cost function applied to all studied algorithms have been shown. Table 3 and 4 show the overhead occurred in those studied algorithms applied in 'highway' and 'shopping mall' mobility model respectively.

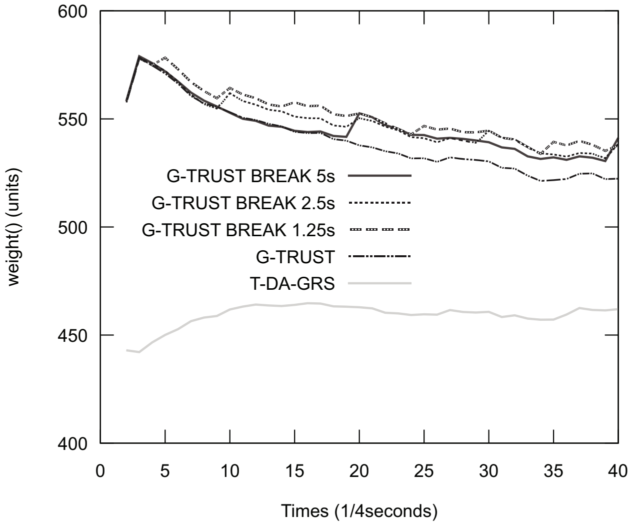

**Figure 4. Comparison of weight() measuring on all studied algorithms in 'highway' mobility model**

**Table 3. Overhead of applying 'BREAK' at each $T\_break$ comparing to G-TRUST on 'highway' model**

| | G-TRUST | G-TRUST BREAK 1.25s | G-TRUST BREAK 2.5s | G-TRUST BREAK 5s |
|---|---|---|---|---|
| Packet Used | 1978.89 | 2107.75 | 2022.58 | 1988.44 |
| Low-trusted Node Break | 0 | 36.39 | 24.09 | 14.55 |
| Created Token from Break | 0 | 79.35 | 53.27 | 34.00 |

By observing those graphs, we can see clearly that G-

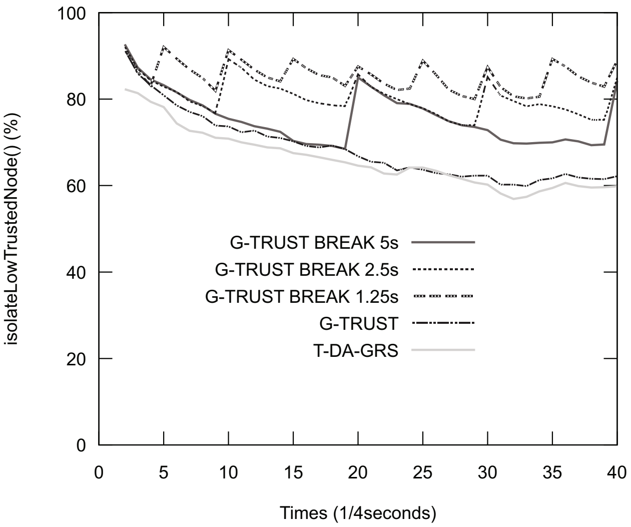

**Figure 5. Comparison of isolateLowTrustedNode() measuring on all studied algorithms in 'highway' mobility model**

TRUST can yield a better trusted spanning tree over DA-GRS in both network models (highway and shopping mall). However, the quality comparing between those BREAK algorithms and G-TRUST in shopping mall model cannot see a clear cut different result. This is a result from the typical topology of shopping mall itself. Since nodes in this model move slowly, the changing or dynamicity of the topology is very low. This means the trying on adapting spanning tree is almost useless because there is no opportunity to do so. On the other hand, nodes in highway model have high mobility and highly changing of environmental nodes. These make it worth to do BREAK.

To summarize: G-TRUST enables to select robust (trustable) trees and the quality of these trees has been measured against the random trees created by DA-GRS. However, G-TRUST has no ability to reorganizing the created tree to maintaining efficient trusted spanning tees in dynamic topology of network. Hence, 'BREAK' has been introduced to cope with those issues in G-TRUST BREAK. As can be seen from all figures shown in this section, all G-TRUST incorporating with BREAK yielded the better (in terms of trust) trees than pure G-TRUST. This confirms the importance of the ability to adapt trusted spanning tree into a changing of environment. Though, this adaptation produced more overhead in G-TRUST BREAK using $T\_break$ at 5s as illustrated in Table 3 and 4, for the others BREAK produced lower overhead than G-TRUST. This can be explained. Since the low-trusted node has been placed in a suitable for the quality of tree, there is no need to change or trying to do BREAK again and again. As soon as the tree getting more robust, the less overhead will be produced. Hence, it can be said that BREAK can both help in increasing the quality of the trusted spanning tree and reducing the overhead over time. Another important issue in using BREAK is how to choose the suitable $T\_break$ to apply the heuristic in a dynamic way. To this context of study, with provided mobility model, the comparison among results from G-TRUST BREAK depicts that the best $T\_break$ is at every 1.25 seconds.

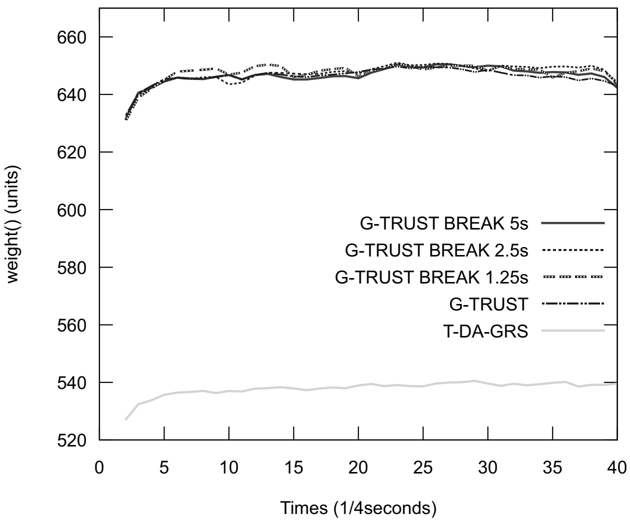

**Figure 6. Comparison of weight() measuring on all studied algorithms in 'shopping mall' mobility model**

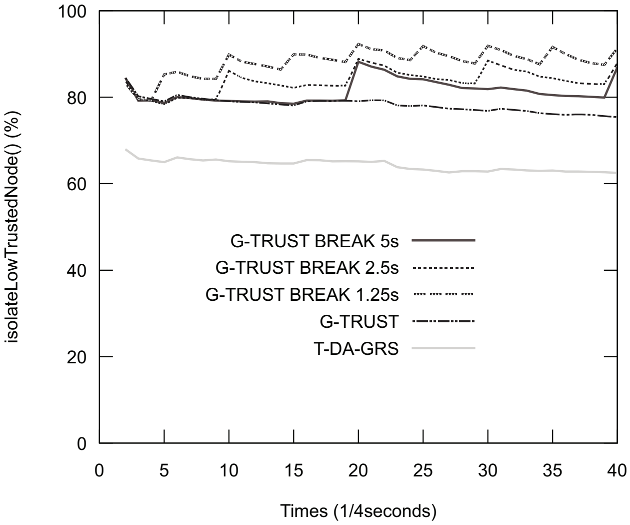

**Figure 7. Comparison of isolateLowTrustedNode() measuring on all studied algorithms in 'shopping mall' mobility model**

**Table 4. Overhead of applying 'BREAK' at each $T\_break$ comparing to G-TRUST on 'shopping mall' model**

| | G-TRUST | G-TRUST BREAK 1.25s | G-TRUST BREAK 2.5s | G-TRUST BREAK 5s |
|---|---|---|---|---|
| Packet Used | 2743.13 | 2446.61 | 2462.04 | 2787.36 |
| Low-trusted Node Break | 0 | 45.02 | 25.32 | 12.75 |
| Created Token from Break | 0 | 93.70 | 53.75 | 27.67 |

## 6. Conclusions and Perspectives

Higher quality of service in Delay Tolerant MANETs is the main target for this work. Therefore the notion of trust has been added to Spanning Forest used in DA-GRS. A relaxation of the DA-GRS notion of rendez-vous has been applied. This allowed to consider the case when several nodes meet simultaneously and to broaden the choice of trees. This opportunity has been assessed through the use of simulation on realistic mobility models. G-TRUST is an adaptation from DA-GRS consisting of a simple heuristic choosing the most trusted node when merging trees. And G-TRUST BREAK is an extension of G-TRUST for automatic and dynamic reconfiguration of trees. The quality of the two heuristics and the robustness of the resulting spanning trees has been evaluated using two different cost functions and clearly illustrates the improvement in terms of tree quality and usefulness of dynamic reconfiguration.

In terms of perspectives, future work will explore the opportunity of automatic tuning of G-TRUST BREAK : e.g. providing nodes have ability to adapt and learn from their experience and local knowledge in order for nodes to suggest the more suitable next $T\_break$. The assumption that global knowledge cannot be assumed will be kept.

## References

[1] K. Fall, "A delay tolerant network architecture for challenged internets," *Proceedings of ACM SIGCOMM 2003, Computer Communications Review*, vol. Vol 33, August 2003.

[2] G. F. Marias, P. Georgiadis, D. Flitzanis, and K. Mandalas, "Cooperation enforcement schemes for manets: a survey: Research articles," *Wirel. Commun. Mob. Comput.*, vol. 6, no. 3, pp. 319–332, 2006.

[3] K. Wrona, "Distributed security: Ad hoc networks & beyond," *presented at the Ad Hoc Network Security Pampas Workshop, RHUL, London, September 16–17, 2002.*, 2002.

[4] T. D. Huynh, N. R. Jennings, and N. R. Shadbolt, "An integrated trust and reputation model for open multi-agent systems," *Autonomous Agents and Multi-Agent Systems*, vol. 13, no. 2, pp. 119–154, 2006.

[5] S. Marti, T. J. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," in *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*, (New York, NY, USA), pp. 255–265, ACM, 2000.

[6] A. Casteigts and S. Chaumette, "Dynamicity aware graph relabeling systems (da-grs), a local computation based model to describe manet algorithms," *International Conference on Parallel and Distributed Computing Systems*, pp. 231–236, November 2005.

[7] L. Hogie, *Mobile Ad Hoc Networks: Modelling, Simulation and Broadcast-based Applications*. PhD thesis, Univerity of Le Havre, University of Luxembourg, April 2007.

[8] A. W. J. David Lewis, "Trust as a social reality," *Social Forces*, vol. 63, pp. 967–985, June 1985.

[9] T. D. Huynh, N. R. Jennings, and N. R. S. Shadbolt, "An integrated trust and reputation model for open multi-agent systems," *Autonomous Agents and Multi-Agent Systems*, vol. 13, pp. 119–154, September 2006.

[10] R. Guha, R. Kumar, P. Raghavan, and A. Tomkins, "Propagation of trust and distrust," *International World Wide Web Conference (WWW2004)*, 2004.

[11] S. Buchegger and J.-Y. L. Boudec, "Performance analysis of the CONFIDANT protocol: Cooperation of nodes — fairness in dynamic ad-hoc networks," in *MobiHOC*, IEEE, June 2002.

[12] S. David and T. J. Pinch, "Six degrees of reputation: The use and abuse of online review and recommendation systems," *First Monday*, vol. 11, March 2006.

[13] M. Seredynski, P. Bouvry, and M. A. Klopotek, "Preventing selfish behavior in ad hoc networks," in *Congress on Evolutionary Computation (CEC 2007)*, pp. 3554 – 3560, IEEE Computer Society, September 2007.

[14] S. Sukumaran and R. E. Blessing, "Reputation based localized access control for mobile ad-hoc networks," in *ADHOC-NOW*, Lecture Notes in Computer Science, pp. 197–210, Springer, 2006.

[15] Q. He, D. Wu, and P. Khosla, "Sori: A secure and objective reputationbased incentive scheme for ad-hoc networks," *WCNC2004*, 2004.

[16] J. V. D. Merwe, D. Dawoud, and S. McDonald, "A survey on peer-to-peer key management for mobile ad hoc networks," *ACM Comput. Surv.*, vol. 39, no. 1, p. 1, 2007.

[17] A. Casteigts, "Model driven capabilities of the da-grs model," *ICAS '06: Proceedings of the International Conference on Autonomic and Autonomous Systems*, p. 24, 2006.

[18] A. Casteigts, "Model driven capabilities of the da-grs model," in *Intl. Conf. on Autonomic and Autonomous Systems (ICAS)*, IEEE, 2006.

[19] L. Hogie, F. Guinand, and P. Bouvry, "The madhoc metropolitan adhoc network simulator." http://litis.univ-lehavre.fr/∼hogie/madhoc/.