# Trust in Ad Hoc Networks: A New Model Based on Clustering Algorithm

Ali Mansouri[1] and Mohamed Salim Bouhlel[2]
*(Corresponding author: Ali Mansouri)*

Higher Institute of Applied Languages and Computer Sciences of Beja[1]
Boulevard of the Environment B.P. 340 Beja 9000, Tunisia
Higher Institute of Biotechnology of Sfax (ISBS)[2]
Soukra Road km 4, B.P. 1175, 3038 Sfax, Tunisia
(Email: mehermansouri@yahoo.fr)

*(Received Nov. 13, 2017; Revised and Accepted May 5, 2018; First Online Feb. 16, 2019)*

## Abstract

An Mobile Ad-hoc network (MANET) is formed when group of mobile wireless nodes collaborate between them to communicate through wireless links in the absence of the fixed infrastructure and any centralized control. This paper focuses on the design a self-stabilizing clustering algorithm in MANETs. The Topology that we propose is a partitioning based on trust between members of a group. It forms a structure able to adapt dynamically to changes in the topology. Some cryptographic-based schemes have been proposed to secure the clustering process, but they are unable to handle the internal attacks. To defend against insider malicious nodes, trust and reputation management systems should be used .Our solution is based on our efficient trust model and distributed algorithm to clustering network.We present our clustering approach based on trust for applications in the field of security.

*Keywords: Clustering Algorithm; Maintenance of the Topology; Mobile Ad Hoc; Self-stabilizing; Trust Relationships*

## 1 Introduction

A mobile ad hoc network is a collection Mobile entities interconnected by a wireless network forming temporary independently of any infrastructure or centralized administration. The nodes in Mobile ad hoc networks join and leave the networks dynamically. At some point of time there is a possibility of enormous increase in the size of the network. Handling nodes in big network may put a burden on network management schemes and may introduce delays in the net-work.

Dividing big networks in small groups called clusters may prove to be a good solution for handling them in a better and efficient manner. As MANET (Mobile Ad hoc networks) is self organized, the challenge of achieving security is critical. Evolving and managing trust relationships among the nodes in the network are important to carry efficient transmissions Clustering organize the ad hoc networks hierarchically and create clusters of ad hoc nodes which are geographically adjacent. Each cluster is managed by a cluster head (CH) and other nodes may act as cluster gateway or cluster member.

In this article, we present a clustering approach for efficient, scalable and secure clustering of MANETs. Our proposal consists on forming clusters around the trustworthy nodes; in other words, the node that has highest trust value is elected as the cluster head. A threshold of trustworthy is used to perform system stability.

## 2 Related Work

In the literature, there are many proposals to construct clusters in mobile ad hoc networks.The first algorithms of Lowest-ID clustering algorithm (LID) proposed by Baker and Ephremides [5]. Clustering High-Connectivity (HCC) of are based on a particular criterion the selection of cluster-heads, which is the identifier of a node. This algorithm to form clusters in a single jump, where each member is its direct neighbor cluster-head. In the construction phase clusters, nodes communicate with their neighbors to have a local knowledge and thus fix the cluster-head.

This phase is repeated periodically for any topology change. The algorithm cited in [15]is a modified version of the Lowest-ID algorithm .The authors propose a clustering algorithm to reduce the work-Clustering fic control. A node broadcasts a single message containing his clustering decision. According to his local knowledge of the topology, each node decides to become a head-cluster or not. This decision is communicated to the neighborhood, forcing the neighbors of the new cluster-head who are not yet affiliated a cluster choose it as a cluster-head.

In [6], authors propose energy efficient secure trust based clustering algorithm for mobile wireless sensor network. Their solution creates one hop members to minimize the overhead and take into account the trust level of a node, mobility, remaining energy and its distance to neighbors.

In [18], authors present a preference-based protocol for trust and head selection for cluster-based MANET perform the tasks of a certification authority and proactive secret sharing scheme is used to distribute the private network key to the CHs. In this solution, each cluster is first formed based on the trust values of the neighbor nodes. To create cluster, an ad hoc node evaluates its neighbor nodes' of neighbor nodes; each node chooses one node that has the highest value as its trust guarantor. Then, the chosen node becomes the CH and the chooser becomes a member of the cluster, a node of the second highest trust value is chosen, in this way, a cluster is formed by the CH which has the highest trust value among the cluster members.

The other trust-based clustering scheme is designed in [1], Authors propose trust based secure on demand routing protocol (TSDRP) for Manet's. In this scheme each node evaluates the trust value of neighbor nodes and recommends one of neighbors that have the highest trust value as its trust guarantor. Then a node becomes a member of CH node which is one-hop away.

In [16], authors propose a self-stabilizing clustering algorithm in mobile ad hoc networks Clustering Algorithm is another trust-based clustering scheme. It evaluates the stability of node through computing the neighbor change ratio and the residual battery power of mobile nodes. To elect CHs by using the voting mechanism, each node votes other nodes only if the node is the most trustful one among its neighbor nodes and the node's stability is better than itself.

In [10], authors propose an efficient secure group communication in MANET using fuzzy trust based clustering and hierarchical distributed group key management which includes a trust value defining how much any node is trusted by its neighborhood and used the certificate as node's identifier. It uses voting mechanism to elect the most trusted node.

In [17],authors give a honey bee algorithm–based efficient cluster formation and optimization scheme in mobile ad hoc networks. It aims to elect trust worthy stable CHs that can provide secure communication via cooperative nodes. The authors in [17], authors propose performance analysis of TSDRP and AODV routing protocol under black hole attacks in Manets by varying network size.

The authors in [12], authors present a multi-metric-based algorithm for cluster head selection in multi-hop ad hoc network to improve the search performance and scalability of MANETs with trust mechanism. In this solution, the trust relationship is formed by evaluating the level of trust using Bayesian statistic analysis and clusters can be formed and maintained with only partial knowledge which makes it suitable for distributed autonomous MANETs.

In [11], authors give a model of mobility aware clustering scheme with bayesian-evidence trust management for public key infrastructure in ad hoc networks.

The authors in [14], propose an efficient trust-based scheme for secure and quality of service routing in MANETs. A composite trust model for secure routing in mobile ad-hoc networks is proposed in [13], and Trust threshold based public key management in mobile ad hoc networks proposed in [3]. Also, a preference-based protocol for trust and head selection for cluster-based MANET is proposed in [18].

## 3 Global Architecture and Criteria of Clustering

This section introduces our topology. We assume first that all nodes periodically broadcast a hello message to their neighbors in a single jump for the information of the nodes around them. Our topology is organized clustered. Each cluster consists of a cluster-head, a core and a periphery.

- The cluster-head is the node that identifies the cluster. He is responsible for the communication between clusters. The cluster-head is the root of a under tree built during the clustering process and covers all members the cluster.

- The core is the center of cluster. The cluster-head is one of the members of the core of its cluster. All core members are neighbors to the cluster-head.

- The periphery is composed of cluster members that are not in the core. Figure 1 illustrâtes the main features and elements of our topology.

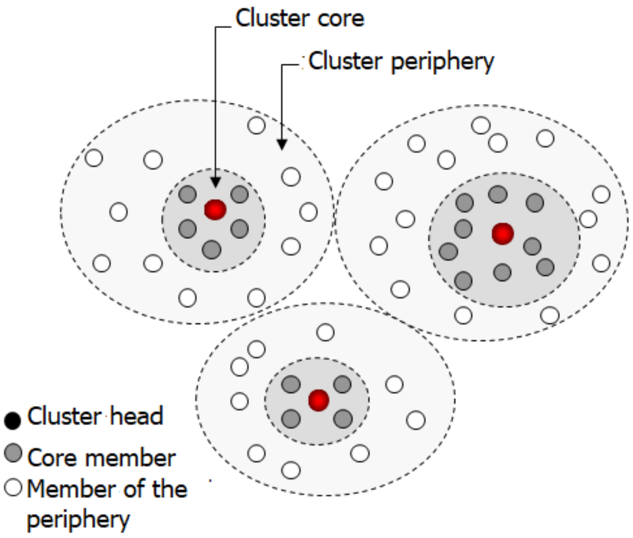

Figure 1: General structure of our topology

We are interested in the clustering area of security. Our algorithm uses trust as partitioning criteria.Trust is fundamental to maintain a certain level of security. This is a important aspect in the design of network.

However, in a dynamic and mobile environment without trusted authority centralized, it is not easy to assess confidence. Many existing solutions propose to calculate the confidence in MANET based on the information that a node can collect other nodes passively [8].

As these interactions are frequent in the cooperative behavior of the nodes of a MANET, it will not difficult to quickly establish a first estimate of the level of trust between direct neighbors. If ordinary interactions are not sufficient to evaluate a ratio trust/distrust, the nodes can generate additional traffic to evaluate how much they trust their neighbors. Thus, to manage its trust, each node $i$ maintains $tv$ value of trust $(i, j)$ for any neighbor $j$ to node $i$ (and possibly former neighbors and nodes not adjacent where the node i receives recommendations). This value reflects the degree of trust or distrust node $i$ has on its neighbor $j$. Several trust functions were proposed in the literature. We use quantification confidence proposed in [9] and we extend to reflect Account trusted recommendations developed in [20].

Thus, our confidence values are real numbers between -1 and +1. A number negative represents the degree of distrust. -1 indicates a total distrust. A number positive represents the degree of confidence. 1 represents absolute confidence. When a new or unknown node j between in the neighborhood of node $i$, the node $i$ initializes $tv(i, j)$ to a first value init_trust ($tv(i, j)$ = init_trust).

This initial value is useful for two nodes that have not yet reported together. For example, if they are completely unknown $init\_trust = -1$ else $init\_trust = 1$.

Note that two neighboring $i$ and $j$ may have different interpretations of their exchanges.Thus, $tv(i, j)$ may be different from $tv(j, i)$. We use the following function to calculate the value of the confidence node $i$ to node $j$:

$$tv(i, j) = \tanh\left(\sum_{k=1}^{n_1} u_k w_k\right) \tag{1}$$

Where $n_1$ is the number of interactions between the two nodes. $w_k$ is the weight of the interaction number $k$. $u_k$ is 1 if the $k$ is positive interaction and -1 if it is negative. The function tanh is used to project the sum of different interactions in the interval [-1, 1].

Several examples of interactions can be used to calculate confidence values . All these interactions is based on the routing information. Here we develop the all interactions we use in our clustering algorithm.

**Passive Knowledge:** A node can obtain important information a neighbor in road construction, for example. In fact, if a node starts in "promiscuous" after the transmission of all packets to hear the retransmission by the destination node, it can get the following information about this neighbor [2]:

- It acts as a black hole if the packet is not forwarded.
- There is a change of attack if the content has changed.
- It makes an attack if a manufacturing self-produced packet is transmitted.
- It makes an identity theft attack if the IP addresses were falsified.
- It induces delays by delaying the retransmission of the packet.

**Accuracy/packet error:** When a node receives a correct packet, can increase the value of trust which he attributes to the one that sent the packet, and other nodes in the path from the source (if the protocol routing provides information on the nodes that make up a route from a source to a destination). Similarly, if the received packet is wrong, the receiver can reduce the value of the confidence he attributes to the sender of the package and the value of the confidence he attributes to intermediate nodes of the road.

**Altruistic Behavior:** If an intermediate node on a route to a destination given, receives a packet for which the next hop is not available, it can remove the package and notify the sender. Even So, if there is a route to the final destination can use this route from its cache, send the packet on the new road and notify the sender the broken link. If the new road is to be correct, it reveals that the sender Error in altruistic behavior. Therefore, this information can be used to increase trust between the two nodes. Compliance / non-compliance with the rules of clustering: a node that does not meet the clustering rules is obviously a malicious node. His neighbors may detect this problem by observing how it sets its clustering variables, described later in this chapter in its hello messages.

**Inconsistent Trust:** A node that distributes false reports trust or lies about his relationship of trust is malicious. This behavior may be detected by comparing the ratio of trust receipt and monitoring communications of its neighbors.

Communication with malicious nodes: when a node regularly exchanges messages with a malicious node, it is considered a malicious node. These direct assessments of trust can be strengthened by reports distributed trust. The confidence reports allow nodes to share the information in confidence and disseminate in the network. A simple approach to distribute the relationship of trust is for each node to broadcast only trusting relationships with its immediate neighbors. A report confidence initiated by node $k$ lists the values of trust that has the other nodes, namely $tv(k, j)$. When node i receives reports of confidence of a certain node $j$, it uses them to improve their confidence value $tv(i, j)$ as follows:

Where $n2$ is the number of nodes that have sent confidence reports node $j$ to node $i$. If the received one report of an un-known node, report trust is not considered. In addition, the use of $tv(i, k)$ as a weight for a relationship of trust initiated by a node $k$ favors direct considerations confidence. Some malicious nodes can lie about trust. However, these false reports can be detected by neighbors.Monitoring Mutual nodes avoids inconsistencies trusted reports received.

To use the confidence values calculated by the nodes as a criterion for clustering,we define two confidence thresholds $S_{min}$ and $S_{max}$ where $S_{min} \leq S_{max}, S_{max} \in [0, 1]$ and $S_{min} \in [-1, 0]$.

- Full Trust (TT): A relationship between two nodes i and j is a relationship full trust $((Relation(i, j) = TT) if tv(i, j) \in [S_{max}, 1] and tv(j, i) \in [S_{max}, 1])$.
- Partial Trust (PT): A relationship between two nodes $i$ and $j$ is a relationship partial trust (Relation $(i, j)$ = PT) if and only if:
  - $Tv(i, j) \in [S_{max}, 1]$;
  - $Tv(j, i) \in [S_{max}, 1]$;
  - $Tv(i, j) \in [S_{min}, S_{max}]$.
- Suspicion (DT): A relationship between two nodes i and j is a relationship distrust $((Relation(i, j) = DT)$ if and only if $tv(i, j) \in [-1, S_{min}]$ or $tv(j, i) \in [-1, S_{min}])$.

Note that the three relationships are symmetrical. For example, if a node $i$ has a total trust with a node $j$ then the node $j$ has a relationship total confidence with node $i$. In the following, we develop the different steps of our algorithm clustering.

## 4 Clustering Algorithm Based on Trust

This algorithm uses a distributed heuristic and tries to minimize the explicit information of the formation of clusters. Our algorithm is composed of three sub algorithms: 1, 2 and 3. In the follows, the authors show the rules forming algorithm.

**Algorithm 1 :RULE0: The excluded members.**

```
Begin
if $\vee j \in Ni.Relation(i, j) = DT$ then
    $CH_i = null \wedge Parent_i = null.$
end if
```

### 4.1 Description of Rules of the Algorithm

In this section, we present the rules of our clustering algorithm.We start by characterizing a legitimate state. A legitimate state is stable clustering training cluster-heads, core members, members of the periphery and members excluded. That algorithm consists of five detailed rules , each node determines the role as hello messages it receives from its neighbors.

**Algorithm 2 :RULE1 & RULE2: Selection and updating the cluster-head members**

```
RULE1
Begin
if $\vee j \in Ni.Relation(i, j) = DT$ then
    $CH_i = null \wedge Parent_i = null.$
end if
RULE2
if $\vee j \in N.i[Relation(i, j) = TT \wedge Compare(TT - edge_i, TT - edge_j)] \wedge \vee j \in N.i[CH_j \neq j \wedge (CH_j = null \wedge Relation(i, j) = TT) \wedge Compare(TT - edge_i, TT - edge_j)])$ then
    $CH_i = i \wedge Parent_i = i \wedge Hop - CH_i = 0$
end if
```

**Algorithm 3 :RULE3 & RULE4: Selection and update the other cluster nodes.**

```
RULE3
if $\vee k \in N.i \exists j \in N.iCompare(Relation(i, j), Relation(i, k)) \wedge CH_i \neq i$ then
    $CH_i = CH_j \wedge Parent_i = j \wedge HopCH_i = HopCH_{j+1}$
end if
RULE4
if $CH_i \neq CHParent_i \vee HopCH_i \neq HopCHParent_{i+1}$ then
    $CH_i = null \wedge Parent_i = null \wedge HopCH_i = null$
end if
```

In our algorithm, the rules (R0), (R2) and (R4) have priority over other rules (R1) and (R3). Indeed, an asset that has a top incorrect value of its variable initializes its state null. Then, it executes the rule corresponding to become a cluster-head, a core member, a member of the periphery or excluded member. The clustering algorithm added to hello messages the following fields:

- $TT - edge_i$: Number of TT relationships a node $i$ have with its neighbors.
- $CH_i$: The cluster cluster-head is attached node $i$. $CH_i$ is equal to null if node $i$ does not yet belong to a cluster.
- $Core_i$: The kernel member belongs node $i$. If $i$ is a clusterhead or ring member then $Core_i$ is set to i. If $i$ is not yet attached to a kernel then $Core_i$ is set to null.
- $Hop - CH_i$: Number of hops from node i to clusterhead.
- $Tv(i, j)$: For each neighbor j, this field represents the value of the trust from node $i$ to node $j$.

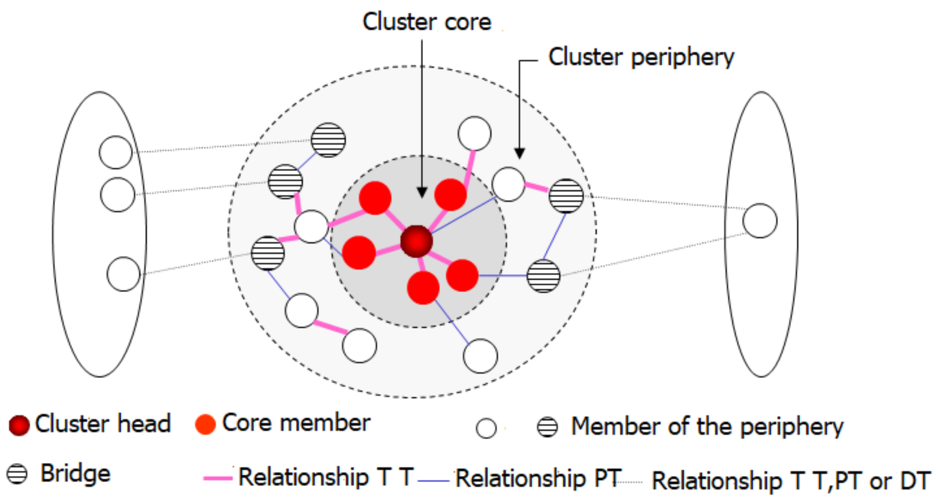

Figure 2: The resulting clustering

- *Parent$_i$*: This field expresses the father of node i in the cor-responding subtree. Clustering is performed in three fully dynamic and distributed phases:

  **Phase 1: Election of cluster-head.** Each node which has the largest number TT-edge among its neighbors who do not yet belong to a cluster declares as a cluster-head. In case of a tie the node with the most neighbors is elected. In case of equal values TT-edge and the equal number of neighbors, the node that has the greater identity is privileged. Once a node becomes a cluster-head, he puts his identity in the scope of its CH hello messages and changes the value of Hop field - CH to zero.

  **Phase 2: Core training.** All the neighbors who have relation-ships TT with a cluster-head form the cluster core. Their hello messages contain the identity of the cluster-head in the CH field and the value 1 in the Hop-CH field. A node can have relationships with several TT cluster-heads. In this case, the node chooses the cluster-head that has the largest TT-edge.

  **Phase 3: Formation of the periphery of the core.** After the phase 1 and Phase 2, the nodes surrounding the core joining the cluster according to the two steps following (TT privileged relationship is a relationship that PT is privileged to DT relationship.

  **Step 1.** Members of the periphery having TT relations. After incorporation cores, if any of the nodes that have not yet ad-hered to a cluster TT and have relations with at least one ring, they join the cluster which they have the greatest confidence value.

  **Step 2.** Members of the periphery having PT relationships. The latter step is to add the nodes that share relationships with the PT least one node in the cluster. A node in this category favors cluster with which it has the lowest distance (number of hops) to the cluster-head. The neighbor with whom he has a relationship PT and the lowest distance clusterhead became his father in the sub-tree rooted. The cluster-head is the root of the subtree. This subtree simplifies communication between clusters. A node in the periphery with at least one neighbor belonging in another cluster is called a gateway. When a node joins a cluster, it updates the CH and CH-Hop fields of hello message.

Clustering obtained is shown in Figure 2.

To succeed clustering, despite the presence of malicious nodes, the honest nodes cooperate closely. They do not communicate the message clustering malicious nodes and ignore all messages from clustering these nodes. Thus, clustering messages and data dissemination spend only by TT relation-ships or relationships PT. However, even if all malicious nodes were detected, clustering can be disrupted.

The condition on the number of malicious nodes and their dispersion in the network is necessary. In fact, if the network is not sufficiently dense and malicious nodes are scattered so that they prevent honest nodes to participate in clustering, the protocol will fail to achieve a complete clustering. As a result, isolated nodes and clusters can appear disconnected.

## 4.2 Clustering Example

We explain our clustering algorithm by applying it to the set of nodes described in Figure 3. In this example, we assume that the nodes have already calculated their confidence values from their direct neighbors. Each node has a unique identifier and is denominated by the trust he attributes to his neighbors. The confidence thresholds are set at $S_{min} = 0$ and $S_{max} = 0.2$.

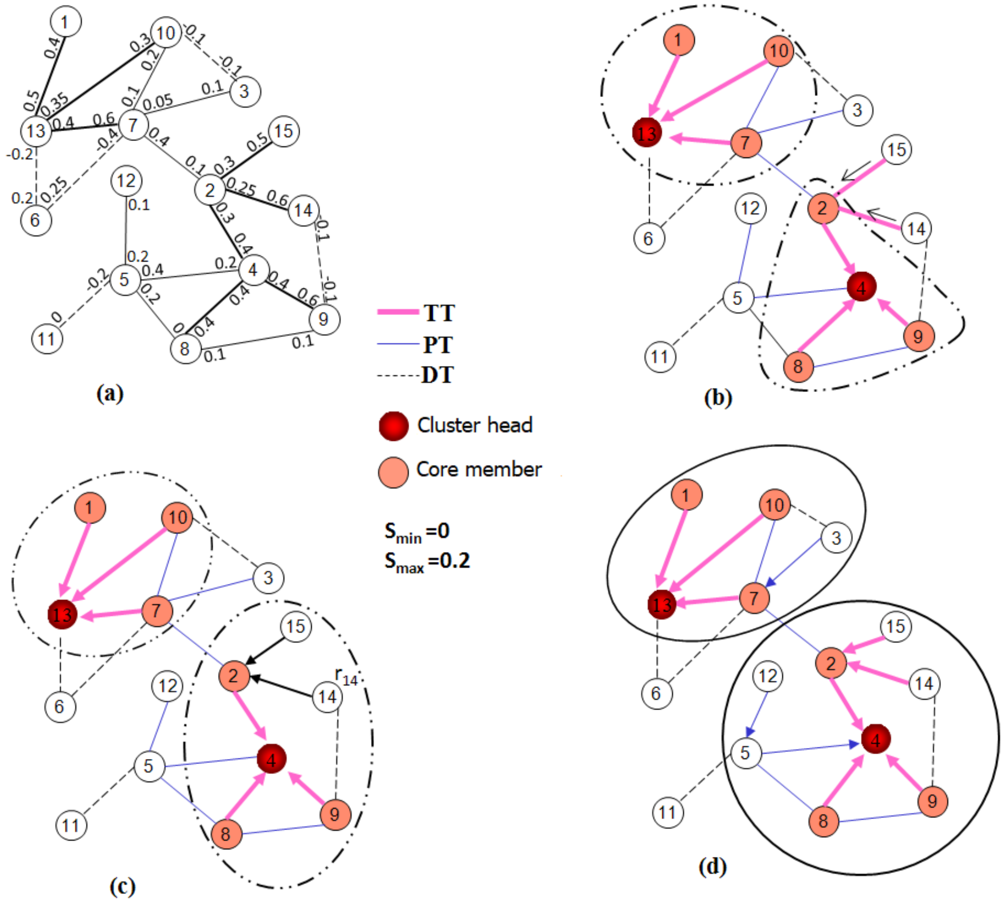

Figure 3: Example of clustering

Figure 3(a) illustrates the different relationships TT, PT and DT in the network according to the values of $S_{min}$ and $S_{max}$. Figure3(b) show nodes 4 and 13 are clusterheads according to the number of TT relationships they have with their neighbors.

Recall that this information is broadcast in hello messages. Each of the nodes 4 and 13 puts its identity in the CH field and updates the Hop-CH field to 0. Then, nodes 1, 7 and 10 (respectively 2, 8 and 9) which share a relation TT with cluster-head 13 (respectively 4) form the core of it (see Figure 3 (c)) Nodes 1, 7 and 10 (respectively 2, 8 and 9) change their fields CH and Hop - CH from their hello messages to respectively 13 and 1 (respectively 4 and 1). The two nodes 14 and 15 have TT relations with node 2. Thus, they join cluster 4. They are attached to kernel member 2 (see Figure 3(c)).

They update their CH and Hop-CH fields accordingly. In the last step, the nodes 12 and 5 (respectively 3) that share a PT relationship with a node belonging to cluster 4 (respectively 13), join this cluster and update the CH and Hop - CH fields of their hello messages. Oriented edges in the example illustrate the subtrees built during the clustering process. When a node joins a cluster, it chooses a father in the shortest way to the cluster-head. Nodes 6 and 11 do not share any TT or PT relationships with other nodes of the network. So, the clustering algorithm does not take into account these nodes in the clusters obtained.

## 4.3 Convergence and Accuracy of the Algorithm

We show in this section that our algorithm converges to a state legitimate. For example, from any initial state, the algorithm con-verges to a stable state composed of cluster-heads, core members, member of the periphery and excluded members. We will assume for this that the nodes are not malicious.

**Lemma 1.** *The node running the rule (R0) does not change its state only time and remains stable there after.*

*Proof.* A node that only sharing DT relations with its neighbors will belong to any cluster, and runs the rule (R0). In addition, any of its neighbors will consider it, and it will be ignored. Therefore, it will no longer change state. □

**Lemma 2.** *The node running the rule (R1)stabilizes after more* $(\triangle - 1)^2$ *movements.*

*Proof.* The node that has the maximum number of TT compared Relations with all its neighbors with which it shares a TT relationship will become a cluster-head. It does not change state thereafter because the decision is local and has best value (the maximum number of TT relations). This comparison depends only the number of relationships that TT is previously updated by the message hello. Against by, in some cases, the rule (R1), the node is declared as a cluster-head because it has the maximum number of relationships TT compared with all neighbors with whom it shares TT relationship and who are not yet to a cluster. His decision depends only on its neighbors which are stabilized at the after $(\Delta - 1)$ move. Hence, a cluster-head stabilizes more after $(\triangle - 1)^2$ movements. □

**Lemma 3.** *The system enters a legitimate state in* $O$ $(n(\triangle - 1)^2)$.

*Proof.* The other vertices which are not cluster-heads always choose the parent with whom they share the strongest relationship. This Parent exists and is unique because the Com-pare function (x, y) selects a single vertex. The best relationship is the relationship with a cluster-head, which is the root of subtree. According to the previous lemma, a cluster-head converges to a stable state more after $(\triangle - 1)^2$ movements. The algorithm converges and a legitimate state $O(n(\triangle - 1)^2)$. □

# 5 Maintenance of the Topology

The clustering algorithm is self-stabilizing. It runs continuously and readjusts clusters based on trust relationships between nodes. Relationships confidence evolves over time depending on the interactions between the nodes. The Mobility can also change the situation of clusters. In fact, when a node acquires new neighbors or loses some of them, because of mobility, several changes can appear in the situation of the node inside Cluster:

1) The number of TT (TT-edge) relationships of the node can change. For example, if the node is a cluster-head, it can no longer be. If the node is a member from the periphery, it can become a member of the kernel or a cluster-head.

2) The PT or TT relationships that link a node to a cluster can break and the node no longer belongs to the cluster.

3) A node that has only DT relationships with its neighbors can acquire PT or TT relationships and thus joins a cluster.

4) Etc. Phase 1, Phase 2 and Phase 3 of the clustering algorithm are re-executed as often as necessary to form new clusters or up-date existing clusters.

## 5.1 Election of a New Cluster-head

Several situations may involve the election of a new cluster-head. We focus on two of these situations: the failure of the current cluster-head and the change the TT-edge value of the current cluster-head so that it is no longer the node with the highest TT- edge value among its neighbors.

1) Cluster-head failure: If a cluster-head goes down, its wires (that's to say other members of the same kernel) no longer receive his hello messages periodicals. In this case, the kernel members re-initialize the CH field from their messages hello to null. Upon receipt of these modifications of the message hello, members of the underlying periphery are also putting the CH field of their hello messages to null. This launches the clustering phases of these nodes and rearranges the cluster.

2) Modification of the TT- edge value: A cluster head including one of its neighbors has a higher value TT-edge re-initializes the CH field of its messages hello to null. When receiving updates to the hello messages, the neighbor's cluster-head that are members of the cluster propagate this re-initialization to all members of the cluster. This leads to a failure situation of the cluster-head.

## 5.2 Breaking Trust Relationships

When a node is authenticated as a member of the group and has at least a TT or PT relationship with a cluster member, he remains a member of the cluster. When the TT or PT relation-ships that link the node to the cluster are broken (because of mobility or change of trust value), the node must to be excluded from the cluster: it is considered a malicious node. At this level, exclusion does not have a practical impact. In fact, if the node is malicious, it does not will not respect the clustering rules. The node then joins another cluster if it has TT or PT relationships with other nodes in the network. This modification may generate other changes in the constitution of clusters. Note here that all stages of clustering are based on trust and strict respect for Clustering rules for members. Malicious nodes may not respect these rules. In this case, using the confidence values, it is possible to detect the malicious nodes. As described earlier in this section, a node can detect that a neighbor is not complying with clustering rules by controlling values of its clustering variables (contained in its hello messages).

## 5.3 Failure of a Core Member or a Member of the Periphery

When a kernel member or a member of the periphery fails, the other periphery members that depend on it are isolated from the cluster. These re-execute Phase 3 clustering to select a new father in the cluster or to join another cluster.

## 5.4 Management of New Nodes

When a new member j becomes neighbor of a member i, i assigns the value 1his trust relationship with $j(tv(i,j) =$

1). This means that a new member he is granted a TT relationship upon his arrival and, therefore, has access to the different clustering steps. However, if i and j fail to authenticate, they attribute to each other a relationship of mistrust.

# 6 Evaluation

We simulated our algorithm to evaluate its performance and compare it with other clustering algorithms performance. In This section provides an overview of our simulation model and results we have obtained. We simulated our approach within the platform NS2 network simulation. Our simulation MANET models a maximum of 100 nodes moving randomly in an area of 1000x1000 m2 under model Random waypoint Model [4]. Each node is equipped with a radio transceiver capable of transmitting up to 250 m.

We use as 802.11 protocols MAC layer in our experiments. We assessed the stability of our system clustering by studying the variation in the number of clusters it generates. To see the behavior of this approach and to measure the effect that will cause the implementation of our algorithm in an OLSR network, we performed several simulations with variable number of nodes and different nodes velocity. We used NS2 [7] as a network simulator.

We performed simulations with, and without clustering inter-val and we have recorded the average number of clusters built (which we note NC) and the average time during which a cluster is Maintained.

## 6.1 Trust Value of Cluster Head Based on the Number of Nodes

To approve the efficiency of our algorithm, we compared it with another algorithm in the literature, which is the algorithm of clustering based on node density. We notice that the trust values of the clusterhead in our proposal are much more important than in the algorithm based on density.

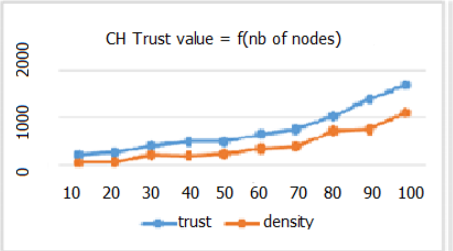

Figure 4: Average trust value of CH = f ( nb of nodes ), V = 10 m/s

In our algorithm the trust of the CH varies between 224,07 and 1673,9 while in the algorithm of clustering based on density it varies between76,076 and 1100, 7 (see Figure 4).

## 6.2 Number of Clusters Formed Based on the Number of Nodes in the Network

Figure 5 shows the evolution of the number of clusters in relation to the number of nodes in the network for a maximum speed of 10 m /s.

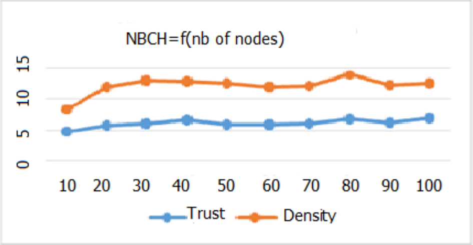

Figure 5: Average number of cluster = f (nbr nodes), V = 10m/s

We notice that the numbers of clusters in our proposal are less than in the algorithm based on density, which shows the stability of our proposal.

## 6.3 Number of Clusters Formed Based on the Number of Nodes in the Network

Figure 6 shows the evolution of the number of clusters in relation to the number of nodes in the network for a maximum speed of 10 m /s.

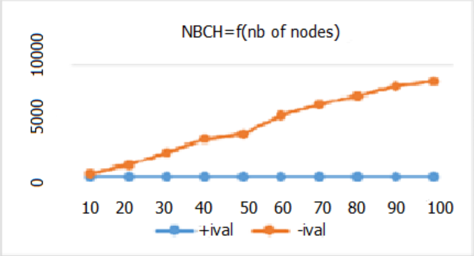

Figure 6: Average number of cluster = f (nbr nodes), V = 10m/s

We notice a great improvement with the use of the clustering interval. The number of clusters varies between 246 and 8588 in the case where the clustering interval is not used, when this number varies between 4.8 and 6.8 with the use of clustering interval for a network with 100 nodes.

## 6.4 Trust Value of Cluster Head Based on the Number of Nodes

Figure 7 shows the evolution of trust value of clusters in relation to the number of nodes in the network for a maximum speed of 10 m /s.

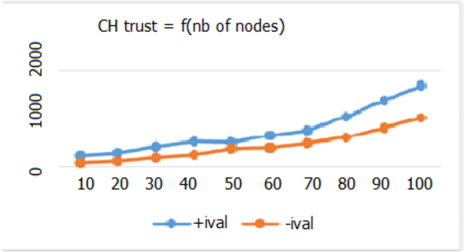

Figure 7: Average trust value of CH = f ( nb of nodes ), V = 10 m/s

We notice a great improvement with the use of the clustering interval. The trust value varies between 88,9 and 1112,5 in the case where the clustering interval is not used, when it varies between 224,07 and 1673.9 with the use of clustering interval for a network with 100 nodes.

### 6.5 Average Cluster Duration Based on the Number of Nodes in the Network

Figure 8 shows the behavior of the average time during which a cluster is built based on the number of nodes in the network.

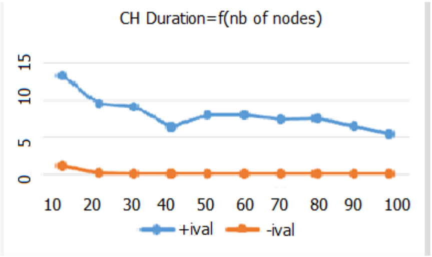

Figure 8: Average cluster duration = f(nbr nodes) , V = 10m/s

We notice a significant improvement brought by the clustering interval. The average duration of clusters varies between 0.007 ms and 1.116 ms in the case where the clustering interval is not used, when this number varies between 5,39 ms and 13.37 ms with the use of clustering interval for a network with 100 nodes.

## 7 Comparison and Analysis

We compared our clustering algorithm with two existing schemes SGCP [19] (Secure Group Communication Protocol)and LID [5](Lowest IDentifier). LID is one of the most known protocols clustering. LID is usually used as a reference protocol evaluation of clustering algorithms performance. We simulated the three protocols in the three scenarios described above mobility: mobility low, medium, high mobility and mobility. We considered different connectivity rate and measured the number of clusters realized by each three protocols.

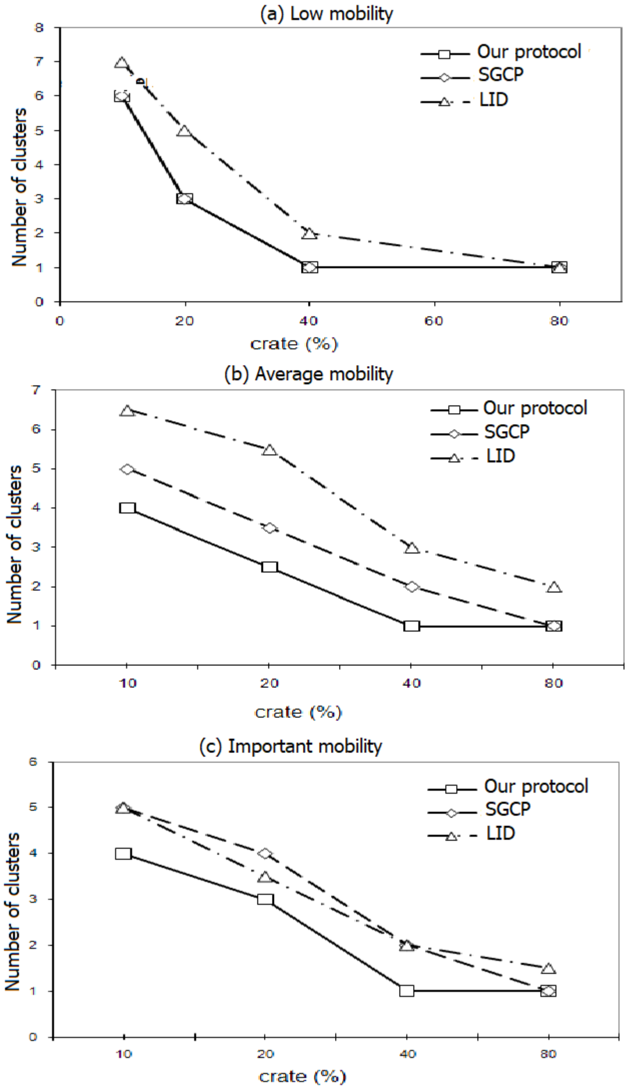

Figure 9: Comparison of the number of clusters of our protocol with LID, SGCP

Figure 9 shows that our algorithm gives good results. It has the same number of clusters as LID in a low mobility scenario(see Figure 9(a) ) and has the lowest number of clusters in medium and high mobility scenarios.(see Figure 9(b),(c) We also studied the variation of the number of clusters of three protocols over time. For this, we also performed the three protocols under the same conditions of mobility and connectivity for 50s.

Figure 10 represents our results. It's clearly shows that the variation in the number of clusters is not important and is stable with LID and our protocol. It is not the case for SGCP where the number of clusters increases dramatically after 20s simulation. This shows that compared to SGCP and LID, our protocol Clustering is stable: it generates a reasonable number of clusters in all mobility scenarios.

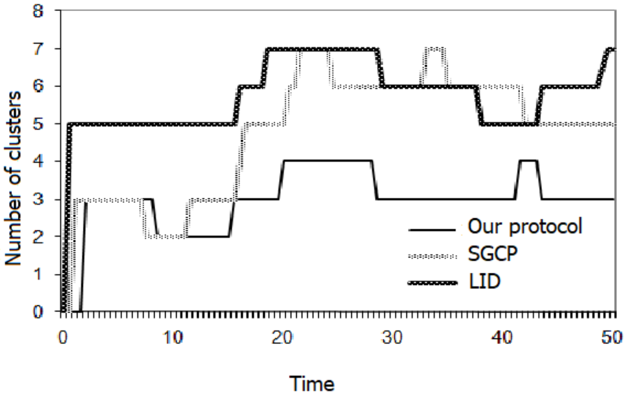

Figure 10: Évolution of the number of clusters

## 8 Conclusions

Our clustering algorithm is used to manage the network dynamics, it is based on building a topology to minimize the mobility of the network, optimize the scalability, facilitate and secure protocols communication. Our proposal is based on the definition of a model Dynamic and distributed trust for ad hoc mobile networks. Our approach is to divide the network into clusters organized into subtrees linked and supervised by cluster-heads. By Therefore, the addition or removal of a member only affects the cluster to which it belongs. The security of our protocol is enhanced by its clustering criterion that constantly monitors the relationship of trust between nodes and expels malicious nodes in the broadcast session.

The clustering algorithm is self-stabilizing. It runs continuously and read just clusters based on trust relationships between nodes. Relationships confidence evolves over time depending on the interactions between the nodes. The Mobility can also change the situation of clusters.

To succeed clustering, despite the presence of malicious nodes, the honest nodes cooperate closely. They do not communicate the message clustering malicious nodes and ignore all messages from clustering these nodes. Thus, clustering messages and data dissemination spend only by TT relation-ships or relationships PT. However, even if all malicious nodes were detected, clustering can be disrupted. The condition on the number of malicious nodes and their dispersion in the network is necessary. As perspective to this work to make our algorithm more stable, we added the concept of the threshold of trust, which represents the trust value at which each node can act as cluster head.

Our algorithm gives good results. It has the same number of clusters as LID in a low mobility scenario and has the lowest number of clusters in medium and high mobility scenarios. This shows that compared to SGCP and LID, our protocol Clustering is stable: it generates a reasonable number of clusters in all mobility scenarios. According to the results of simulations that we made, we notice a great improvement and better system stability with the adopted solution.Also, we plan to use the clustering solution to manage cryptographic key in MANETs.

## References

[1] A. Akshai, G. Savita, C. Nirbhay, and J. Keyurbhai, "Trust based secure on demand routing protocol (tsdrp) for manets," in *Fourth International Conference on Advanced Computing & Communication Technologies (ACCT'14)*, pp. 432–438, IEEE, 2014.

[2] P. Asad Amir and M. Chris, "Establishing trust in pure ad-hoc networks," in *Proceedings of the 27th Australasian Conference on Computer Science*, vol. 26, pp. 47–54, 2004.

[3] J. H. Cho, I. R. Chen, and K. S. Chan, "Trust threshold based public key management in mobile ad hoc networks," *Ad Hoc Networks*, vol. 44, pp. 58–75, 2016.

[4] B. Christian, H. Hannes, and C. Xavier, "Stochastic properties of the random waypoint mobility model," *Wireless Networks*, vol. 10, no. 5, pp. 555–567, 2004.

[5] B. Dennis and E. Anthony, "The architectural organization of a mobile radio network via a distributed algorithm," *IEEE Transactions on communications*, vol. 29, no. 11, pp. 1694–1701, 1981.

[6] R. Eid, S. Muhammad, N. Syed Hussnain Abbas, B. Khan, and U. Kamran, "Energy efficient secure trust based clustering algorithm for mobile wireless sensor network," *Journal of Computer Networks and Communications*, vol. 2017, 2017.

[7] K. Fall, K. Varadhan, *The NS Manual*, 2003.

[8] T. George and B. John, "On trust models and trust evaluation metrics for ad hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 2, pp. 318–328, 2006.

[9] T. George and B. John, "A testbed for comparing trust computation algorithms," in *Proceedings of the 25th Army Science Conference (ASC'06)*, 2006.

[10] K. Gomathi, B. Parvathavarthini, and C. Saravanakumar, "An efficient secure group communication in manet using fuzzy trust based clustering and hierarchical distributed group key management," *Wireless Personal Communications*, vol. 94, no. 4, pp. 2149–2162, 2017.

[11] V. S. Janani and M. S. K. Manikandan, "Mobility aware clustering scheme with bayesian-evidence trust management for public key infrastructure in ad hoc networks," *Wireless Personal Communications*, vol. 99, no. 1, pp. 371–401, 2018.

[12] P. Jay, K. Rakesh, K. Sarvesh, and J. P. Saini, "A multi-metric-based algorithm for cluster head selection in multi-hop ad hoc network," in *Next-Generation Networks*, pp. 513–524, Springer, 2018.

[13] R. H. Jhaveri, N. M. Patel, and D. C. Jinwala, "A composite trust model for secure routing in mobile ad-hoc networks," in *Ad Hoc Networks*, InTech, 2017.

[14] H. Jingsha, Z. Z. Ali, M. M. Qasim, H. M. Iftikhar, M. S. Pathan, Z. Nafei, "An efficient trust-based

scheme for secure and quality of service routing in manets," *Future Internet*, vol. 10, no. 2, p. 16, 2018.

[15] X. Li, S. Jill, and Y. Shaokai, "Evaluating trust in mobile ad hoc networks," in *The Workshop of International Conference on Computational Intelligence and Security*, Citeseer, 2005.

[16] A. Mansouri and M. S. Bouhlel, "Self-stabilizing clustering algorithm in mobile ad hoc networks," in *SAI Intelligent Systems Conference (IntelliSys'15)*, pp. 978–983, IEEE, 2015.

[17] C. Nirbhay, A. Akshai, G. Savita, and J. Keyurbhai, "Performance analysis of tsdrp and aodv routing protocol under black hole attacks in manets by varying network size," in *Fifth International Conference on Advanced Computing & Communication Technologies (ACCT'15)*, pp. 320–324, IEEE, 2015.

[18] M. Rajkumar and S. Subramanian, "A preference-based protocol for trust and head selection for cluster-based manet," *Wireless Personal Communications*, vol. 86, no. 3, pp. 1611–1627, 2016.

[19] Y. M. Tzeng, C. C. Yang, and D. R. Lin, "A secure group communication protocol for ad hoc wireless networks," *Advances in Wireless Ad Hoc and Sensor Networks, Signals and Communication Technology Series*, Springer, pp. 102–130, 2007.

[20] L. Zhaoyu, J. Anthony, and T. Robert, "A dynamic trust model for mobile ad hoc networks," in *10th IEEE International Workshop on Future Trends of Distributed Computing Systems (FT-DCS'04)*, pp. 80–85, 2004.

# Biography

**Ali Mansouri** is an assistant professor at Jendouba University since 2011; Head of Computer Department at Higher Institute of Applied Languages and Computer Sciences in Beja since 2013. Researcher at the SETIT Research Laboratory: Research Unit: Science and Technology of Image and Telecommunications (from the University of Sfax), member of PRISMa Laboratory Nautibus Building (ex 710) Claude Bernard University Lyon 1 Research field: Dynamic algorithms for ad hoc network communication, Coloration of graphs. Computer Science Master: INSA DE LYON: The National Institute of Applied Sciences of Lyon in France.Master's degree in Computer Science Management Faculty of Jendouba Tunisia.

**Mohamed Salim Bouhlel** is a full professor at Sfax University, Tunisia. Head of the Research Lab SETIT since 2003. President and founder of the Tunisian association on HMI since 2013. Editor in Chief of the international Journals "HMI", "MLHC and a dozen of special issues. Chairman of many international conferences research interests: Image processing, Telecom and HMI in which he has obtained more than 20 patents so far. More than 500 articles were published in IJ, IC & books.