Journal of Theoretical and Applied Information Technology
20th February 2015. Vol.72 No.2
© 2005 - 2015 JATIT & LLS. All rights reserved·

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A CLUSTERING ALGORITHM BASED ON NODES TRUST

[1]MOHAMED DYABI ,[2]ABDELMAJID HAJAMI ,[3]HAKIM ALLALI

[1] Phd stdent, LAVETE Laboratory Hassan 1st University. Faculty of Sciences and Technologies Settat Morocco.

[2] Assis. Prof., LAVETE Laboratory Hassan 1st University. Faculty of Sciences and Technologies Settat Morocco.

[3]Prof., LAVETE Laboratory Hassan 1st University. Faculty of Sciences and Technologies Settat Morocco.

E-mail: [1] mohamedyabi@gmail.com , [2] abdelmajidhajami@gmail.com , [2] hakim-allali@hotmail.fr

ABSTRACT

A mobile ad hoc network (MANET) is a wireless network without the support of any fixed infrastructure. Security is one of the main challenges in ad hoc network due to dynamic topology and mobility of nodes. Clustering is one of the main techniques that are used to increase the scalability of MANETs, but without any security considerations clustering is prone to various security attacks. 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. This work proposes a trust based clustering algorithm which forms a cluster around the trustworthy nodes. The Criteria used to select the CH is the trust value of a node. Our algorithm gives major improvements regarding the number and the trust value of elected cluster heads. Analysis and simulation results are used to show the performance of our algorithm, compared with other clustering algorithms in literature.

Keywords: *Ad Hoc , Trust, Clustering, Olsr*

1. INTRODUCTION

A Mobile ad hoc network or MANET is a collection of resource limited mobile nodes which does not rely on any fixed or centralized infrastructure. These nodes dynamically form a temporary network and communicate with each other through bandwidth limited and multi hop wirelesses links [1]. From architectural point of view, MANETs can be classified into two types on networks: flat and hierarchical. All nodes have the same roles and responsibilities in the network of flat MANETs. They do not scale well but as the number of network nodes increases, the overheads of routing and other operations grow dramatically. Therefore, only small number of nodes and devices can be managed as flat MANETs. In order to support large number of devices, ad hoc networks should be organized hierarchically.

Clustering organize the ad hoc networks hierarchically and create clusters of ad hoc nodes which are geographically adjacent. Each cluster is managed by a clusterhead(CH) and other nodes may act as cluster gateway or cluster member

In the literature, several clustering approaches were proposed. They generally differ on the cluster head selection criteria.

The cluster head role is resource consuming since it's always switched on and is responsible for key generation, key distribution, and key maintenance. If a node has this role, it would burn it resource quickly, and after it died, all its members would be headless.

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.

This paper is organized as follows: in Part II, we will present an overview of the OLSR standard protocol, Part III will present the clustering solution. Part IV describes the details of our proposed algorithm. In Part V, we will show the results obtained from the simulations that we perform.

Part VI describes the details of the trust threshold. Finally, part VII, concludes the paper and draws directions for future work.

2. THE OLSR PROTOCOL

The optimized link state routing (OLSR) protocol [1] is a proactive routing protocol that employs an efficient link state packet forwarding mechanism called multipoint relaying.

Optimizations are done in two ways: by reducing the size of the control packets and also by reducing the number of links that are used for forwarding the link state packets. The reduction in the size of link state packets is made by declaring only a subset of the links in the link state updates. The subset neighbors that are designated for link state updates are assigned the responsibility of packet forwarding are called multipoint relays.

The optimization by the use of multipoint relaying facilitates periodic link state updates. The link state update mechanism does not generate any other control packet when a link breaks or when a link is newly added. The link state update optimization achieves higher efficiency when operating in highly dense networks. The set consisting of nodes that are multipoint relays is referred to as MPRset. Each given node in the network elects an MPRset that processes and forwards every link state packet that this node originates. Each node maintains a subset of neighbors called MPR selectors, which is nothing than the set of neighbors that have selected the node as a multipoint relay. A node forwards packets that are received from nodes belonging to its MPRSelector set. The members of both MPRset and MPRSelectors keep changing over time. The members of the MPRset of a node are selected in such a manner that every node in the node's two hop neighborhood has a bidirectional link with the node.The selection of nodes that constitute the MPRset significantly affects the performance of OLSR. In order to decide on the membership of the nodes in the MPRset, a node periodically sends Hello messages that contain the list of neighbors with which the node has a bidirectional link. The nodes that receive this Hello packet update their own twohop topology table. The selection of multipoint relays is also indicated in the Hello packet. A data structure called neighbor table is used to store the list of neighbors, the two-hop neighbors, and the status of neighbor nodes. The neighbor nodes can be in one of the three possible link status states, that is, unidirectional, bidirectional, and multipoint relay.

3. THE CLUSTERING SOLUTION

Clustering is the most popular method developed to provide resource management over mobile ad hoc networks .This technique based on partitioning the network in smaller and manageable groups, each group called cluster [2].

Clustering offers, several benefits when it used with MANETs listed as follows:

- Provides hierarchical architecture.
- Performs key management
- Helps to perform more efficient resource allocation
- Enhances routing process and mobility.

In [3] Yu et al., studied and analyzed various clustering algorithms which are proposed for MANETs. In spite of numerous papers which studied the clustering in [3-11], security is one of the main items that is ignored in these surveys. Considering the vulnerability of MANETs to numerous passive and active security attacks, the clustering schemes can be classified as trust-based or cryptographic-based. Each of these categories protects the clustering process against special type of attackers. Pure cryptographic–based clustering techniques increase the security of clustering operation against outsider and insider attackers, they are unable to detect the compromised nodes and insider attackers. To defend against insider malicious nodes, trust and reputation management systems should be used. Several general purpose reputation management systems have been proposed for MANETs in the literature but they have high overheads which decrease their effectiveness in the resource limited ad hoc networks. In this context, trust-based clustering algorithms integrate the trust management systems with the clustering algorithms and are aimed to reduce the overheads of reputation management. These schemes manage the trust related information for each node and prevent the election of a malicious or compromised node as the CHs or other cluster components.

4. TRUST BASED CLUSTERING

The network can be considered as a set of areas (or clusters). Each cluster is formed around a representative called Cluster Head. Cluster Heads are selected according to a well defined criteria. A cluster is designated by an identifier that relates to its representative (i.e. its cluster head). Each node in the network carries the cluster identifier to which it belongs

1.1 Selection criteria of the cluster heads

This section briefly describes the trust-based clustering schemes which are presented in

the literature. In [12] Elhdhili et al., propose CASAN to elect trustworthy, stable and high-energy CHs. 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 [13] Xu et al., present a trust evaluation based clustering which CHs jointly 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 by Park et al., in [14]. In this scheme each node evaluates the trust value of neighbor nodes and recommends one of neighbors that has the highest trust value as its trust guarantor. Then recommender node becomes a member of CH node which is one-hop away.

VCA or Voting-Based Clustering Algorithm is another trust-based clustering scheme which is presented by Peng et al., in [15]. 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 [16] Kadri et al., propose a secured weight-based clustering algorithm called SCA which includes a trust value defining how much any node is trusted by its neighborhood

and used the certificate as node's identifier. SCA elects CH according to its weight computed by combining stability, battery and etc. It uses voting mechanism to elect the most trusted node.

In [17] Ferdous et al., propose CH selection algorithm based on an efficient trust model. It aims to elect trustworthy stable CHs that can provide secure communication via cooperative nodes.

In [18] Wang et al., present a novel self-clustering maximum flow algorithm 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.

Our proposal presents a simple, light and quiet solution. First, our proposal does not add any new control message and the network is not overloaded or slowed at all. No changes are made to standard control messages. Our solution works transparently with the OLSR standard protocol. Clusters are formed around the trustworthy nodes.

1.2 Node trust computation : trusti

In ad hoc networks, the node process routing control messages and data messages.

To calculate the trust metric of a node, our algorithm use several types of messages, including:

Hello message, TC message and data messages routed through a node.

To determine the weight associated with each type of message we use the Rank Order Centroïde method (ROC) [19]

4.2.1. Multi-criteria analysis method:

Multi-Criteria Decision Analysis, or MCDA, is a valuable tool that can be applied to many complex decisions.

It can solve complex problems that Include qualitative and/or quantitative aspects in a decision-making process.

4.2.2. Why use multi-criteria analysis in trust assessment :

The trust value of a node is calculated based on a number of criteria that the list is not exhaustive. So far we have identified three: Hello message, TC message and data messages routed through a node.

The global trust value of the node is obtained by adding the partial criteria affected by relative weights.

In decision analysis, this operation is called synthesis or additive aggregation.

Regarding the assessment of the relative weights of the criteria, there are several Multi-criteria Decision Analysis methods. We selected Rank Order Centroïde (ROC) [20] for its simplicity and its proven efficiency.

4.2.3. Calculation of weight by the classification rank order centroid:

Step 1: Sort criteria in descending order of importance:

Routed message > TC message > Hello message

Step 2: fill the matrix

	Routed msg	TC msg	HELLO msg	Control
R1	1,00	0,00	0,00	1,00
R2	0,50	0,50	0,00	1,00
R3	0,333	0,333	0,333	1,00
AVG	0,61	0,28	0,11	1,00
				1,00

The column control ensures that all weights are normalized (sum of weights = 1)

After this work, the formula becomes:

RTRST = 0.61 * ROUTEDmsg + 0.28 * TCmsg + 0.11 * HELLOmsg

1.3 OLSR clustering algorithm

In a clustered OLSR network, each node can be in one of three states:

- State 0: not decided. When a node has just arrived, or it has just left its cluster and has no neighbors in its neighborhood, its status is not decided yet. There is no cluster head or cluster member. It must wait for the receipt of HELLO messages.
- State 1: Cluster head. The node was exchanged HELLO messages, and it has the highest trust value. It creates a cluster in which it was appointed head of the cluster.
- State 2: member. The node has exchanged HELLO messages; it has a low trust value compared to its symmetric neighbors, and is part of the cluster members.

Each node evaluates the trust of the other nodes with which it communicates.

If a node operates according to the OLSR specifications, i.e , it sends control messages and forwards data messages periodically, the trust value of the node is incremented according to the weight associated to each type of message , otherwise the trust value will decrement. This information is carried in Hello message.

After receiving the hello messages, each node have a vision about the trust of other nodes by computing the trust average of each one.

Upon receiving a HELLO message, it compares the neighbor's trust with its own trust to decide whether to become a cluster head or join the neighbor's cluster.

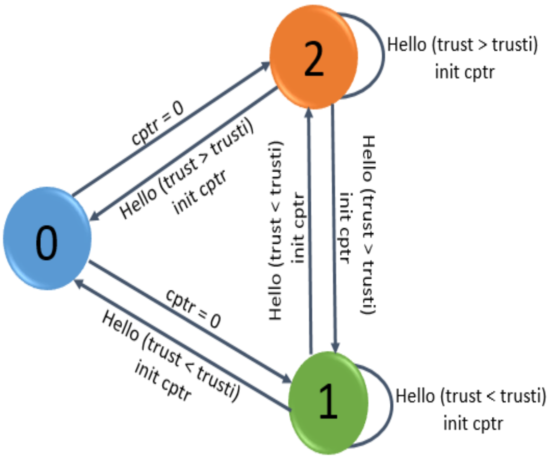

Figure 1. Clustering algorithm

• Initially, each node begins with a status 0 (not decided). Upon receiving a HELLO message, the node compares its own trust (trusti) with the trust of the message it received (trust).

• If (trust < trusti), the node goes to state 1 (cluster head) because its trust value is greater than trust of the received message.

Once in state 1, node i triggers a counter Cptr. If after passing this timeout, the node i has received no HELLO message, that means it has no neighbors in its radio range, so it decides to move to state 0 (not decided state).

• If (trust > trust i), the node goes to state 2 (member) because its trust value is lower than that of the received message.

Once in state 2, node i triggers a counter Cptr. If after passing this timeout, the node i has received no HELLO message, that means it has no neighbors in its radio range, so it decides to move to state 0 (not decided state).

• If the node i is in state 1 (respectively in state2), and it receives a HELLO message with (trust < trusti) (respectively (trust > trusti)), it remains in state 1 (respectively remains in state 2) because its state has not changed.

• If the node i is in state 1 (respectively in state2), and it receives a HELLO message with (trust > trusti) (respectively (trust < trusti)), it moves to state 2 (respectively move to state 1) because its condition has to change.

4.1 System stability

We note that the system may become unstable after receiving several Hello messages. A node may change either its state or its cluster whenever the trust value of the received message is greater than its own trust value. This may cause some instability in the clustering approach.

To prevent this phenomenon, we chose to keep the node to decide its status (i.e. head or member) for a longer time than the period of a HELLO message.

For simulations, we have taken a period equal to three times the emission range of Hello messages. This time, which we call clustering interval, represents the interval at which each node restarts the process of trust calculation [21].

5. SIMULATION RESULTS

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 [22] as a network simulator with the following parameters:

table1 NS2 PARAMETERS

Parameter	Value
Simulation area	1000 x 1000
Radio range	250 m
Number of nodes From	10 to 100 by step of 10
Velocity of nodes	From 0 m/s to 50 m/s by step of 5
Simulation time	300 s

We performed simulations with, and without clustering interval and we have recorded the average number of clusters built (which we note NC) and the average time during which a cluster is maintained

5.1 Trust value of Cluster Head based on the number of nodes

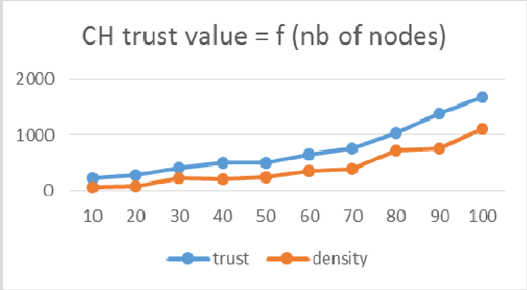

Figure 2. Average Trust value of CH = f (nb of nodes), V = 10 m/s

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 value of the clusterhead in our proposal are much more important than in the algorithm based on density.

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 between 76,076 and 1100,7.

5.2 Number of clusters formed based on the number of nodes in the network

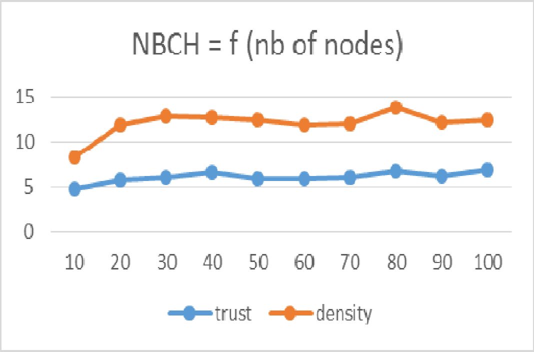

Figure 3. Average Number of Cluster = f (nbr nodes), V = 10m/s

Figure 3 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.

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

5.3 Number of clusters formed based on the number of nodes in the network

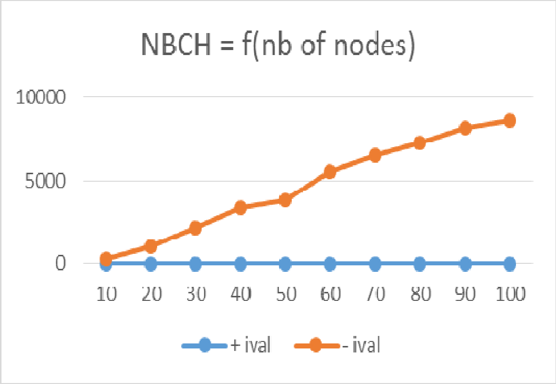

Figure 4. Average Number of Cluster = f (nbr nodes), V = 10m/s

Figure 4 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.

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.

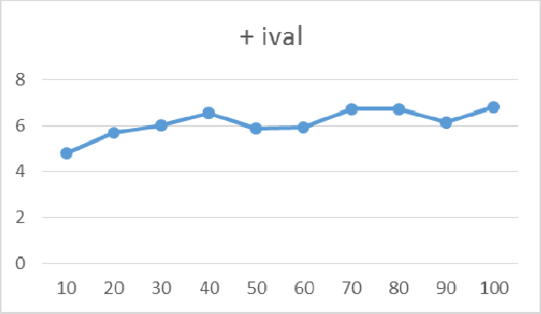

Figure 5. Number of Clusters = f (nb of nodes) V= 10m/s

This figure shows the same information in figure 4 but at different scale.

5.4 Trust value of Cluster Head based on the number of nodes

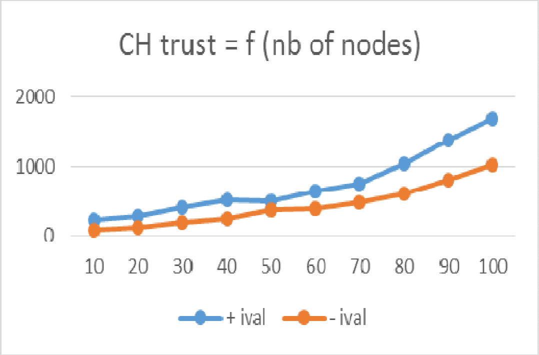

Figure 6. Average Trust value of CH = f (nb of nodes), V = 10 m/s

Figure 6 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.
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.

5.5 Average cluster duration based on the number of nodes in the network

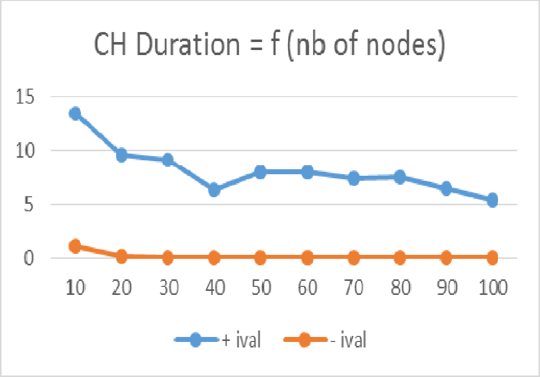

Figure 7. Average Cluster duration = f(nbr nodes) , V = 10m/s

Figure 7 shows the behavior of the average time during which a cluster is built based on the number of nodes in the network. 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

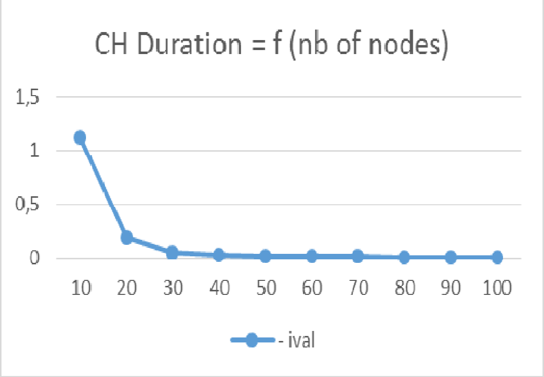

Figure 8. Average Cluster duration = f (nbr nodes), V = 10m/s

This figure shows the same information in figure 8 but at different scale.

6. THRESHOLD OF TRUST

We Notice that the system becomes a little bit stable after the application of the interval of clustering.

However, even if we apply the interval of clustering, a node can change its status or its cluster if the trust value of the received message is bigger than its own trust value while it always have the adequate trust to play the role of the clusterhead.

To resolve this problem we suggest applying a threshold of trust. Therefore, even if the node receives a bigger trust than its own trust, it is going to keep its status until it reaches the threshold of trust.

6.1. Number of clusters formed based on the number of nodes in the network

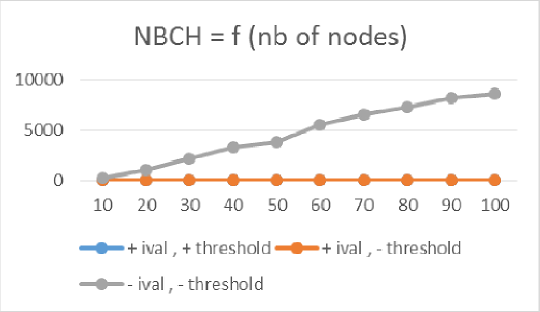

Figure 9. Average Number of Clusters = f(nbr nodes), V= 10m/s

We notice a great improvement with the use of the threshold of trust. The number of clusters varies:

- between 246,25 and 8588,5 : when the clustering interval is not used
- between 4,8 and 6,8 : when the clustering interval is used
- between 2,6 and 4,65 : when the trust threshold is used with the clustering interval

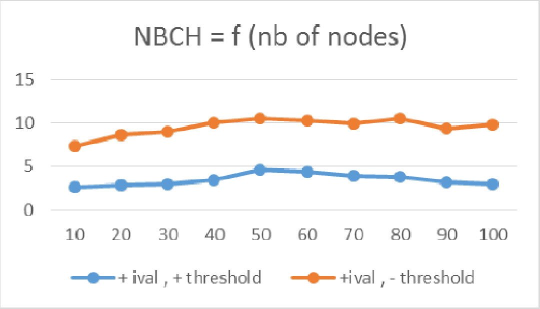

Figure 10. Average Number of Clusters = f(nbr nodes), V = 10m/s

This figure shows the same information in figure 9 but at a different scale

6.2. Trust value of Cluster Head based on the number of nodes

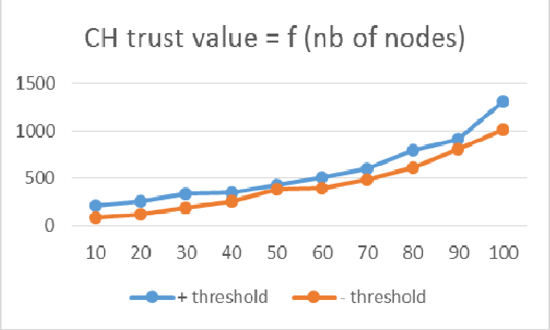

Figure 11. Average Trust value of CH = f (nb of nodes), V = 10 m/s

Figure 11 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.

We notice a great improvement with the use of the threshold of trust.

The trust value varies between 88,9 and 1112,5 in the case where the trust value is not used, when it varies between 214,86 and 1308.9 with the the trust value is use for a network with 100 nodes.

6.3. Average cluster duration based on the number of nodes in the network

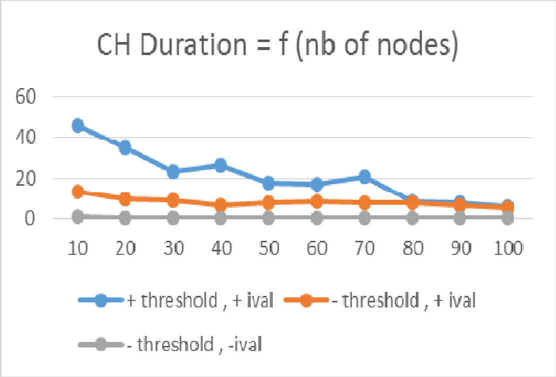

Figure 12. Average Cluster duration = f(nbr nodes) , V = 10m/s

Figure 12 shows the behavior of the average time during which a cluster is built based on the number of nodes in the network. We notice a significant improvement given by the threshold of trust . The average turns between :

- 0.07 and 1,11 ms when the interval clustering is not used.
- 5,39 and 13.37 ms when the interval clustering is used.
-
- 6,1 and 45,76 ms when the threshold of trust is used with the clustering interval

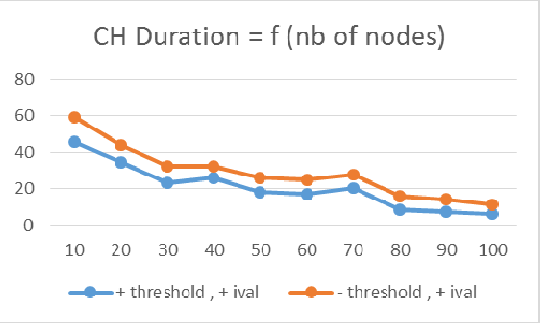

Figure 13. Average Cluster duration = f(nbr nodes) , V = 10m/s

This figure shows the same information in figure 12 but at different scale

7. CONCLUSION AND PERSPECTIVES

Clustering is an important research topic for (MANETs) because clustering makes it possible to guarantee basic levels of system performance.

A large variety of approaches for ad hoc clustering has been presented.

In this work, we introduce an algorithm for efficient clustering of mobile ad-hoc networks.

Its contributions, compared to existing solutions, are summarized in the following: it does not add any new control message and the network is not overloaded or slowed at all, No changes are made to standard control messages. It works transparently with the OLSR standard protocol. Clusters are formed around the most trustworthy node; in other words, the node that has the highest trust value is elected as the cluster head. 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 clusterhead.

According to the results of simulations that we made, we notice a great improvement and better system stability with the adopted solution.

As perspective to this work, we plan to use the clustering solution to manage cryptographic key in MANETs.

REFRENCES:

[1] Y. Zhang, J. Mee Ng and C. Ping Low, “A distributed group mobility adaptive clustering algorithm for mobile ad hoc networks”, Journal of Computer Communications, vol. 32, (2009), pp. 189-202.

[2] S. Sarkar, T. G. Basavaraju, and C. Puttamadappa, Ad Hoc Mobile Wireless Networks: Principles, Protocols and Applications New York: Auerbach Publications, 2007

[3] J. Y. Yu and P. H. J. Chong, “A Survey of Clustering Schemes for Mobile Ad Hoc Networks”, IEEE Communications Surveys & Tutorials, (2005).

[4] D. Wei and H. A. Chan, “Clustering Ad Hoc Networks: Schemes and Classifications”, 3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks, (2006), pp. 920-926.

[5] S. Chinara and S. K. Rath, “A Survey on One-Hop Clustering Algorithms in Mobile Ad Hoc Networks”, Journal of Network and Systems Management archive, vol. 17, no. 1-2, (2009) June, pp. 183-207.

[6] K. Erciyes, O. Dagdeviren, D. Cokuslu and D. Ozsoyellery, “Graph theoretic clustering algorithms in mobile ad hoc networks and wireless sensor networks”, Applied and Computational Mathematics, vol. 6, no. 2, (2007), pp. 162-180.

[7] R. C. Hincapie, B. A. Correa and L. Ospina, “Survey on Clustering Techniques for Mobile Ad Hoc Networks”, (2006).

[8] G. Kumar, K. K. Tripathi and N. Tyag, “Research Survey of Load Balancing Clusters in Wireless Ad hoc Network”, International Journal of Electronics Engineering, (2011), pp. 305-307.

[9] P. Rai and S. Singh, “A Survey of Clustering Techniques”, International Journal of Computer Applications, vol. 7, no. 12, (2010) October.

[10] I. G. Shayeb, A. R. H. Hussein and A. B. Nasoura, “A Survey of Clustering Schemes for Mobile Ad-Hoc Network (MANET)”, American Journal of Scientific Research, (2011), pp. 135-151.

[11] R. Agarwal and M. Motwani, “Survey of clustering algorithms for MANET”, International Journal on Computer Science and Engineering, vol. 1, no. 2, (2009), pp. 98-104.

[12] M. E. Elhdhili, L. B. Azzouz and F. Kamoun, “CASAN: Clustering algorithm for security in ad hoc networks”, Computer Communications, vol. 31, (2008), pp. 2972-2980.

[13] C. Park, Y. Lee, H. Yoon, S. Jin and D. Chio, “Cluster based Trust Evaluation in Ad Hoc Networks”, pp. 503-507.

[14] L. Xu, X. Wang and J. Shen, “Strategy and Simulation of Trust Cluster Based Key Management Protocol for Ad hoc Networks”, Proceedings of 4th International Conference on Computer Science & Education, (2009), pp. 269-274.

[15] C. Park, Y. Lee, H. Yoon, S. Jin and D. Chio, “Cluster based Trust Evaluation in Ad Hoc Networks”, pp. 503-507.

[16] S. Peng, W. Jia and G. Wang, “Voting-Based Clustering Algorithm with Subjective Trust and Stability in Mobile Ad-Hoc Networks”, IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, (2008), pp. 3-9.

[17] L. Wang and F. Gao, “A Secure Clustering Scheme Protocol for MANET”, International Conference on Multimedia Information Networking and Security (MINES), (2010), pp. 785-789.

[18] P. Chatterjee, “Trust Based Clustering And secure routing Scheme for Mobile Ad Hoc Networks”, International Journal of Computer Networks & Communications, vol. 1, no. 2, (2009) July, pp. 84-97.

[19] Barron, F.H. 1992. Selecting a best multiattribute alternative with partial information about attribute weights

[20] Roy, B. (2005). An overview of MCDA techniques today: paradigms and challenges. In: Figueira, J., Greco, S. and Ehrgott, M. (eds) Multiple criteria decision analysis: state of the art surveys.

[21] A Hajami, M. Elkoutbi. ” Density Based Clustering ”, ICOGCAC 09, 9-11 December 2009 Padur, Chennai, India

[22] Network Simulator NS2 http://www.isi.edu/nsnam/ns/.