Efficient Clustering for Multicast Key Distribution in MANETs

Mohamed Salah Bouassida[1], Isabelle Chrisment[1], and Olivier Festor[2]

[1] LORIA-UHP
[2] LORIA-INRIA
name.surname@loria.fr
MADYNES, Campus scientifique
B.P. 239, 54506 Vandœuvre-lès-Nancy Cedex - France
tel: +33-3-83-59-30-49 - fax: +33-3-83-41-30-79

Abstract. Securing multicast communications in ad hoc networks must consider several challenging factors such as high nodes mobility, limited bandwidth and constrained energy. Moreover, the establishment of a key management protocol within ad hoc environments meets the "1 affects n" problem, which is more critical in such type of networks, due to the high dynamicity of groups.
In this paper, we present an efficient clustering scheme for multicast key distribution in Mobile ad hoc networks. This scheme divides the multicast group into clusters, according to the localization of the group members and their mobility. Simulations indicate a valuable reduction in the average latency of keys distribution and a promising reduction in energy consumption. Analysis also shows that our scheme is scalable.

Key words: ad hoc networks, multicast security, group key management, energy-efficient, mobility-aware

1 Introduction and Motivations

The Multicast transmission is an efficient communication mechanism for group oriented applications, such as video conferencing, interactive multi-party games and software distribution. One main advantage of multicast communication is to save networks resources, mainly by reducing the consumption of bandwidth and source and routers resources.

In parallel to the development of the multicast services within the Internet, the last decade saw the exponential deployment of ad hoc networks, thanks to the emergence of new wireless technologies and standards (e.g. 802.11, Hiperlan [6], ...). A mobile ad hoc network is an autonomous and dynamic collection of devices, that are connected without any fixed infrastructure, that can be highly mobile. This mobility implies that network topology changes rapidly and unpredictably and the connectivity among the hosts varies with time. Generally, mobile ad hoc networks operate on low power devices, having limited energy and bandwidth, low CPU process capability and small memory sizes.

The combination of an ad hoc environment with multicast services to be operated, induces new challenges towards the security infrastructure needed to enable acceptance and wide deployment of multicast communication.

To prevent attacks and eavesdropping, several services need to be provided such as authentication, data integrity and data confidentiality. The most suitable solution to provide these services is the establishment of a key management protocol. This protocol is responsible for the generation and distribution of the traffic encryption key (TEK) to all group members, to encrypt multicast data by the source and decrypt it by the receivers using this TEK, which ensures data confidentiality ; so that only authenticated members are able to receive the multicast flow sent by the group source.

Multicast key distribution has to take into account a challenging element which is the "1 affects n" phenomenon [8] (after a Join or a Leave procedure, the TEK is renewed and redistributed to all group members to keep forward and backward secrecy). To face this problem and to attenuate its impact on the protocol performances, several approaches propose a multicast group clustering [2, 3, 1, 8]. The clustering consists in dividing the multicast group into sub-groups. Each sub-group is managed by a local controller (LC), responsible for local key management within its cluster. Thus, after Join or Leave procedures, only members within the concerned cluster are affected by the rekeying process.

The "1 affects n" phenomenon is more critical in ad hoc networks due to the high nodes dynamicity and additional factors such as mobility awareness, energy and bandwidth consumption.

In this paper, we present a clustering scheme for multicast key distribution in mobile ad hoc networks, consisting in forming strongly correlated clusters, based on localization and mobility information of the group nodes. Our approach, delivers fast, energy-efficient and mobility-aware key distribution in a multicast service within MANETs, and aims to:

- exploit the advantage of broadcast communications in a wireless environment with omni-directional diffusion ;
- optimize the data transmission time. A correlated cluster minimizes the number of relays responsible for forwarding the messages emitted by the source of the group and consequently minimizes the data-transmission time ;
- optimize the bandwidth consumption. This advantage is induced by the minimization of the multi-hop retransmissions number within the cluster ;
- optimize energy consumption. The reduction of energy consumption in an ad hoc network is a true challenge because of the limitation of the batteries resources. A strongly correlated cluster should minimize energy consumption required for the transmission of a message to all members of the cluster. Indeed, this energy is equal to the sum of energy units used by each relay to retransmit the data to its downstream members. The minimization of these relays will decrease the total energy consumption.

To present our approach, the remainder of our paper is structured as follows. Section 2 presents the related work, concerning multicast key management approaches and energy-efficient multicast tree building. In section 3, we describe our clustering scheme for multicast key distribution. In section 4 and 5, we show the interest of clustering for energy reduction and we assess by simulation the efficiency of our approach. And finally, section 6 concludes this paper.

2 Related work

2.1 Multicast Key Management Approaches

Several key management protocols for securing multicast communications have been elaborated over the last few years. We classified them into three main approaches (Figure 1): without clustering approach, with static clustering approach and with dynamic clustering approach.

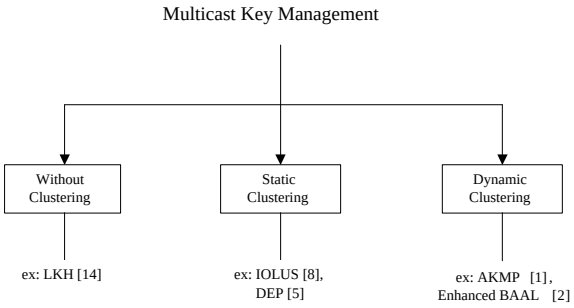

Fig. 1. Classification of multicast key management approaches

Without clustering, all group members share a unique symmetric key (TEK), used within a centralized architecture where a single server is responsible for TEK generation and distribution. LKH (Logical Key Hierarchy) [14] belongs to this approach. In LKH, group members are organized as leaves of a tree with logical internal nodes. The root of the tree is the group key. The cost of a compromise recovery operation

in LKH is proportional to the depth of the compromised member in the LKH tree. LKH scheme proposes maintaining a balanced tree, which gives a uniform cost of O(log n) for rekeying process in group of n members. The original LKH does not consider nodes mobility and localization in an ad hoc environment.
The proposal in [7] enhances the distribution scheme of LKH within ad hoc networks by optimizing energy consumption with the use of geographical localization information related to members. The basic idea of this approach is that members who are geographically close to each other, can potentially be reached by a broadcast, or can use the same path to receive data. By representing group members in a two-dimensions space, [7] uses the K-means clustering algorithm to form groups with strong correlation, and deduces the multicast key distribution tree. But, the mobility of nodes is not considered.
The design of multicast key management trees that match the cellular networks topologies is proposed in [11]. It reduces the communication cost by 55% to 80% compared to key management trees that are independent of the network topology. However, this scheme is built on the infrastructure of the cellular networks and is hardly applicable in ad hoc networks.

In the static clustering approach, the multicast group is initially divided into several subgroups. Each of them shares a local session key managed by a local controller. Protocols proposed within this approach like IOLUS [8] and DEP [5] are more scalable than centralized protocols. In IOLUS, each cluster holds its local TEK ; thus, the multicast flow should be decrypted and re-encrypted whenever it passes from a cluster to another. However, DEP achieves a better compromise between scalability and CPU processing by encrypting data with only one TEK which is shared by all clusters. Thus, local controllers have to decrypt and re-encrypt only the TEK and not all the multicast flow.

The dynamic clustering approach aims to solve the "1 affects n" problem without generating an overhead due to decryption/re-encryption process, this approach merges the two previous approaches. The basic idea is to start a multicast session with centralized key management (first approach), and to divide the group dynamically in order to delegate key management to the local controllers (second approach). Evaluation functions are proposed to decide when group clustering is achieved. AKMP [1], SAKM [4] belong to this approach and are dedicated to wired networks. whereas Enhanced BAAL [2] proposes a dynamic clustering scheme for multicast key distribution in ad hoc networks.

2.2 Energy-efficiency Approaches for Multicast Trees Establishment

The energy efficiency is a challenging issue for multicast key distribution tree in an ad hoc environment. Some interesting research works were published on this subject.

The broadcast incremental power algorithm BIP [12] determines the minimum-power broadcast tree, rooted at the source node, and that reaches every node in the wireless network, by computing the best point of attachment, according to the minimum required energy to reach them from the source. This algorithm takes advantage of the broadcast within wireless environment, but does not consider node mobility. [12] deduces from BIP, an algorithm called Multicast Incremental Power (MIP), which computes the best multicast tree, always by optimizing the energy required to reach every group member from the source, but by pruning nodes which are not members of the group.

BLU (Broadcast Least-Unicast-Cost) [12] superposes the best unicast paths to each destination individually, according to the minimum cost metric. Neither the node mobility nor the advantage of the broadcast within ad hoc environment are considered. An approach dedicated to multicast communications, MLU (Multicast Least-Unicast-Cost) is almost identical to BLU with the difference that unicast paths are established only for the group members. Multicast tree will be built by superposing these theoretically optimal paths.

The third algorithm presented by [12] is BLiMST (Broadcast Link-Based MST), establishes a minimum-energy diffusion tree, using MST technique (Minimum-cost Spanning Tree). The wireless broadcast advantage is also ignored in this algorithm. An algorithm called MLiMST deduces a multicast tree from the one built by BLiMST, by pruning nodes which are not group members.

[12] affirms that MIP establishes the best multicast tree, within an ad hoc environment, having the minimum number of relays responsible for multicast flow forwarding, and the minimum required energy to reach all the group members.

3 Energy-efficient and Mobility-aware Clustering Scheme for Multicast Key Distribution

We present in this section a new dynamic clustering scheme for multicast key distribution dedicated to ad hoc networks. This scheme wants to optimize energy consumption and latency for key delivery. Being mobility aware, our approach needs the geographical location information of all the group members in the construction of the key distribution tree. Thus, we assume that within an ad hoc network, a *Global Positioning System* (GPS) is available.

Our framework network is defined with the following parameters:

1. Nodes are represented by points in a two-dimensions euclidian space, where distance between two nodes $i(x_i, y_i)$ and $j(x_j, y_j)$ is classicaly computed as follows:

$$d(i,j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} = \sqrt{\| i - j \|^2}$$

2. The required power for sending a message from a node i to node j is proportional to the distance $d(i,j)$, and it is computed as following [10]:

$$P = (d_{ij})^{\gamma}$$

   γ is the propagation loss factor in the network that typically holds a value between 2 and 4, depending on the characteristics of the communication medium. We note that the application domain and context of our approach is heterogeneous ad hoc networks, which implies that each node in the network has its specific maximal range, depending on its physical characteristics.
3. Given a cluster with c local members, the distance matrix between theses members is D:

$$D = \begin{pmatrix} 0 & d_{12} & \cdots & \cdots & d_{1c} \\ 0 & \ddots & & & \vdots \\ \vdots & & d_{ii} = 0 & & \vdots \\ \vdots & & & \ddots & d_{c-1,c} \\ 0 & \cdots & \cdots & \cdots & 0 \end{pmatrix}$$

During multicast group initialization, every group member is attached to the group source, called global controller (GC). This entity is responsible for the TEK generation and its distribution to all group receivers. In addition, the GC verifies periodically whether the group is highly correlated, and consequently whether the key distribution process is optimal. The evaluation of the cluster cohesion is determined with a cluster cohesion parameter that we have defined as the centralization index of the cluster around the LC node (node 1):

$$Cohesion = \frac{members\_in\_the\_LC\_range}{cluster\_members\_number}$$

This parameter measures the proximity of the cluster members compared to their controller. The bigger the number of members reachable by the controller in one hop is, the closer the factor of cohesion reaches 1. With this cohesion parameter, we can verify if a cluster is strongly correlated or not.

We define $Min\_Cohesion$ as the minimum threshold that a cohesion factor of a cluster should not exceed. Otherwise, the controller must take the decision to split the cluster, and the election of new local controllers LCs according to their localization, must be initiated.

The split process optimizes the number of new LCs while reaching every member of the cluster. This process is done by the OMCT algorithm (Optimized Multicast Cluster Tree), whose principle is as follows (cf Algorithm 1):

- STEP 1: List_nodes is the list of the c cluster members. List_LCs is initially empty and will contain the local controllers of the new clusters which are built. Node 1 corresponds to the initial local controller of the cluster.

Algorithm 1 OMCT: Optimized Multicast Cluster Tree

List_LCs = ϕ (STEP1)
List_nodes = {1, 2, 3, ..., c}
i = 1;
j = 2;

while (List_nodes $\neq \phi$) **do**
  Sort_List_nodes(i) ; // Sort the list of members according to their distances compared to the node i (STEP2)
  while $d_{(i,List\_nodes(j)} \leq Max\_Distance$) **do**
    // Max_Distance corresponds to the range of the node i
    $List\_nodes = List\_nodes/\{List\_nodes(j)\}$; // Remove List_nodes(j) of the members list
    j++;
  end while
  //The node List_nodes(j+1) cannot be reached directly by the node i (STEP3)

  s=1;
  for (l = 2 to j) **do**
    if ($d_{(s,List\_nodes(j+1)} > d_{(l,List\_nodes(j+1))}$) **then**
      s=l;
    end if
  end for
  LC=s;
  // Elected LC will be closest to the node List_nodes(j+1) (STEP4)
  i=LC;
  $List\_LCs = List\_LCs \cup \{LC\}$; // Add LC to the local controllers list
end while(STEP5)

- STEP 2: Sort the list of members in the ascending order, according to their distances compared to the running LC, initially node 1.
- STEP 3: Traverse the list of members until the node $j+1$ which cannot be joined directly by the LC. All of the j members which can be reached directly by current LC are withdrawn from the List_nodes.
- STEP 4: The nearest node to the node $j+1$, belonging to the range of the running LC, will be elected as current LC. It is thus withdrawn of the List_nodes and added to the List_LCs.
- STEP 5: Iterate steps 2, 3 and 4 until having processed all the cluster members. Thus, all the members are reachable by the List_LCs and can start receiving multicast flow sent by the source.

Once the first clusters are created within the multicast group, the new LCs become responsible for the local key management and distribution to their local members, and also for the maintenance of the strongly correlated clusters property.

Thus, recursively, and at every moment of a multicast group session, the group is composed of strongly correlated clusters, ensuring that their respective cohesion is always higher than the definite minimal threshold $Min\_Cohesion$.

Figure 2 illustrates an execution example of this algorithm. The new LCs are nodes 1, 4 and 8.

The OMCT algorithm 1 achieves a group clustering, according to the localization and the mobility of members group. However, the number of members of each new cluster is not taken into account. Thus, in the example of the Figure 3, node 2 forms a cluster with only one member (node 3). Whereas node 4 can reach node 3, without forming a new cluster. An improvement of the OMCT algorithm is then essential, considering the number of members of each new cluster. This number is moving between two thresholds (cf Algorithm 2). The principals enhancements of the algorithm are as follows:

- STEP 3': Traverse the nodes list, sorted in ascending order according to their distances compared to the current LC. Nodes in the range of the current LC are withdrawn from the List_nodes and added to the list of the local members of this LC, while respecting the constraint of maximum threshold of local numbers (MAX_THRESHOLD). At the end of this step, the node $j+1$ cannot be reached directly by the current LC, or the cluster managed by this LC is saturated.

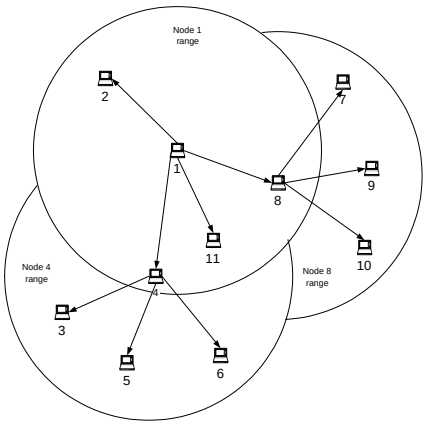

Fig. 2. Illustration example of OMCT

- STEP 6': Traverse the formed clusters. If a cluster has a local members number lower than the minimal threshold MIN_THRESHOLD, then traverse the local members of this cluster and try to move them to others clusters, as far as possible.

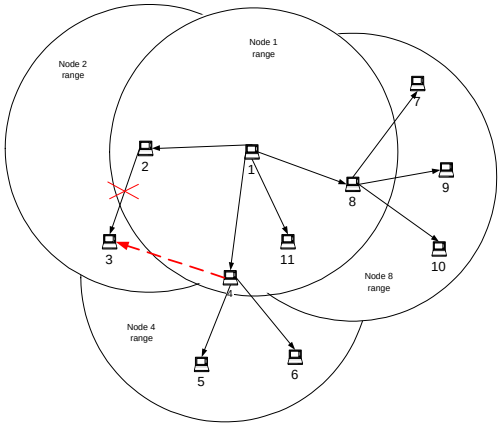

Fig. 3. Illustration example of OMCT V2

Mobility Management Periodically, each LC computes the cohesion parameter of its cluster. According to the value of this parameter, it decides or not to execute the OMCT algorithm to clusterize its sub-group.

When a member moves within the network, it can be no more reachable by its LC. All LCs send periodically "LC_Queries" messages containing their identities and their coordinates.
Thus, a member moving in the network, and receiving LC_Queries from LCs, chooses to join the nearest LC, according to its localization. Figure 4 illustrates this mobility scenario.

Then, for its re-authentication and access control to the new LC, this member uses the concept of the re-authentication ticket. This concept requires that local controllers form a restricted multicast group and share a session key called KEK_{LCs}. The re-authentication ticket contains a password encrypted with the KEK_{LCs}, and is sent securely to the member, during its first authentication in the multicast group. After receiving the ticket, the new LC will decrypt it, check the password and in case of success, the member will be

Algorithm 2 OMCT: Optimized Multicast Cluster Tree V2

List_LCs = ϕ (STEP1)
List_nodes = {1, 2, 3, ..., c}
i = 1;
j = 2;
nb_LCs = 0;

// Initialization of all the local members lists
for ($m = 1$ to c) **do**
  $List\_local\_members\_\{m\} = \phi$
end for
while (List_nodes $\neq \phi$) **do**

  Sort_List_nodes(i) ; // Sort the list of members according to their distances compared to the node i (STEP2)

  while $d_{(i,List\_nodes(j)} \leq Max\_Distance$) **do**
    if ($nb\_List\_local\_members\_\{i\} < MAX\_THRESHOLD$) **then**
      $List\_nodes = List\_nodes/\{List\_nodes(j)\}$; // Remove List_nodes(j) of the members list
      $Liste\_membres\_locaux\_\{i\} = Liste\_local\_members\_\{i\} \cup \{List\_nodes(j)\}$;
      j++;
    else
      Break;
    end if
  end while
  //The node List_nodes(j+1) cannot be reached directly by the node i or the cluster is already saturated (STEP3')

  s=1;
  for ($l = 2$ to List_nodes(j)) **do**

    if ($d_{(s,List\_nodes(j+1)} > d_{(l,List\_nodes(j+1))}$) **then**
      s=l;
    end if
  end for
  LC=s;
  // Elected LC will be closest to the node j+1 (STEP4)
  i=LC;
  $List\_LCs = List\_LCs \cup \{LC\}$; // Add LC to the local controllers list
  nb_LCs ++;
end while(STEP5)

// Algorithm improvement holding account of the members number of new clusters (STEP6')
for ($t = 1$ to nb_LCs) **do**
  if ($nb\_List\_local\_members\_\{t\} < MIN\_THRESHOLD$) **then**
    for ($m = 1$ to $nb\_List\_local\_members\_\{t\}$) **do**
      for ($z = t$ to nb_LCs) **do**
        if ($d_{List\_LCs\{z\},m} \leq Max\_Distance$ & $nb\_List\_local\_members\_\{z\} < MAX\_THRESHOLD$) **then**

          $List\_local\_members\_\{t\} = List\_local\_members\_\{t\}/\{m\}$;
          // Remove m of the local members list of t
          $List\_local\_members\_\{z\} = List\_local\_members\_\{z\} \cup \{m\}$;
          // Add m of the local members list of z
          Break;

        end if
      end for
    end for
  end if
end for

linked to the new cluster. The advantage of the ticket is to allow the LCs to decide about re-authentication and access control of the new members without needing to ask the global controller for it. We note that the ticket must be updated by LCs, periodically, after each member revocation within the multicast group, and when a LC leaves the group. This fact is necessary and mandatory in the case of revocation of a group member which should not be able to re-join the group, since any cluster within the group. A member who left the group for a duration longer than the ticket update period, cannot be re-authenticated using its ticket.

When a LC moves in the network, leaves the group or disappears due to any resources problems, it must previously send a notification message to all its local members asking them for moving to others clusters having nearest LCs. Then, these LCs will run, if necessary, the OMCT algorithm.

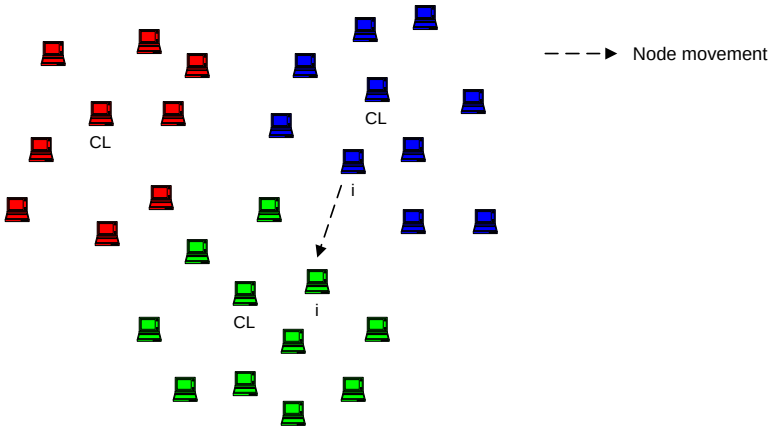

Fig. 4. Nodes mobility

4 Analysis and Discussions

This section presents the analysis realized to validate our cohesion parameter with OMCT algorithm, within ad hoc networks. The topologies are generated randomly and the nodes localization is randomly distributed on all of the network surface. The OMCT algorithm is executed with 10 different topologies. A node is randomly selected to be the multicast group source. Others nodes are selected to be members of the group and the remaining nodes are simple participants.

We choose two variable factors which are group members number (10-20-50) and network surface (50*50 -100*100 - 150*150). The number of nodes within the network is fixed at 100. Propagation loss factor is chosen as $\gamma = 2$, without loss of generality. We assume a maximum range of 50 m. We note that radio frequency of 2,4 Ghz in 802.11b offers a rather important range which can vary from 30 to 100 m, even 400 m following the used hardware.

Our discussion is based on the following metrics:

- the number of new LCs corresponding to the number of new clusters.
- the required energy to send a message in multicast by the source to all of group members. Only the energy required for message sending is computed, the energy used at message reception and that used at signals processing are neglected.
- we use also a metric presented by [13], called *yardstick metric*:

$$yardstick = \left(\frac{m}{p}\right)\left(\frac{m}{n}\right) = \left(\frac{m^2}{n*p}\right)$$

n is the destinations number (group members number), m is the group members number reached by the multicast flow sent by the source, and p is the sum of energies necessary for sending a multicast message. (m/n) represents the attainability factor, and (m/p) represents the energy-consumption factor per reached members number. More optimal is the clustering approach, greater is the yardstick metric.

For the performed analysis, the thresholds of members number within a cluster are not considered. We note also that the local controllers are selected only if they are group members, because they hold the traffic encryption key and can reach the multicast flow. Moreover, all the local controllers are ad hoc nodes and not dedicated routers as wired networks, and it is understandable that a node which is not a group member, does not ensure the role of local controller for this group.

	Surface	Cohesion	Clusters Number	Required Energy	Yardstick Metric
n = 10	50 * 50	0.99	1.1	1571.73	6.36
	100 * 100	0.5	2.5	3716.45	2.13
	150 * 150	0.35	2.8	4628.53	0.97
n = 20	50 * 50	0.985	1.2	1908.82	10.60
	100 * 100	0.445	3.2	5420.88	3.24
	150 * 150	0.32	3.9	6286.40	1.90
n = 50	50 * 50	0.982	1.2	2195.48	23.03
	100 * 100	0.522	3.9	6749.87	7.47
	150 * 150	0.376	4.9	9229.56	4.34

Fig. 5. Clusters Number, Required Energy and Yardstick Metric vs Cohesion

The table in figure 5 shows the number of new clusters compared to the cohesion of the initial group. Smaller the cohesion is, more we need to create new clusters within our multicast group. The creation of new clusters implies the creation of new LCs used as relays to their local members, thus requiring more transmission energy. The table 5 shows also the relationship between the yardstick metric and the group cohesion factor. When the cohesion of the group increases, the local controller can reach more group members while minimizing the required energy. That means the reachability factor (m/n) and the energy-consumption factor (m/p) increase, and consequently the yardstick metric.

We note that the number of created clusters is only influenced by the cohesion factor and not by the number of group members. This implies that our OMCT clustering scheme is scalable. We can also validate our cohesion factor, because it is proportional to the required energy, the number of new clusters and the yardstick metric. From this cohesion factor, we will deduce the minimum threshold, below which a multicast group must run the OMCT clustering algorithm.

To compare our OMCT clustering algorithm with the MIP approach [12], we use the same example presented in [12]. With the same topology, OMCT produces a cluster with cohesion equal to 1, using only one local controller which is the node source 10 (cf Figure 6) ; this solution is optimal in terms of energy consumption and key distribution latency. Whereas the MIP algorithm provides a key distribution scheme requiring more relays and more consumption of energy.

The proposal of [7] presented in section 2.1, uses a refined K-means algorithm, which enhances the Logical Key Hierarchy (LKH) in energy efficiency. K-means procedure provides 15% to 37% savings from the worst possible assignment of the LKH approach. However, as said in [7], there are cases where the application of K-means induces higher energy consumption than a randomly generated tree. This fact revealing the non optimality of the approach in some cases, comes from the fact that K-means algorithm fails to exploit the broadcast advantage in a wireless environment. Infact, in the specific example of two nodes which are at the same distance from a LC but in an opposite direction, K-means fails to clusterize them together, whereas, with the OMCT algorithm, the two nodes will belong to the same cluster, thus benefits from the broadcast advantage, saving energy and decreasing average key distribution latency in the group.

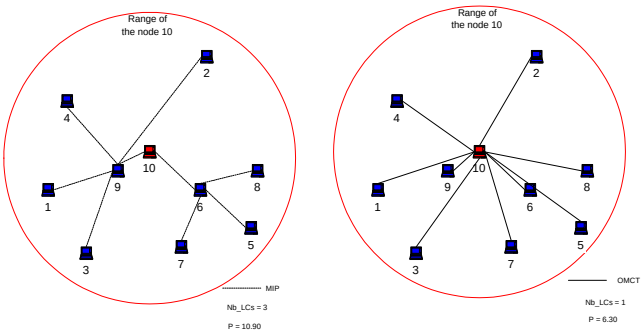

Fig. 6. OMCT vs MIP

5 Simulations and Results

The objective of our simulations is to compare the OMCT-clustering approach with the non-clustering and the localization-independent clustering approaches, according to the following factors:

- Latency: average delay for keys delivery from a sender to a receiver,
- Energy: sum of energy units required for messages transmission throughout the simulation duration.

The non-clustering approach consists in having a centralized multicast group, with one controller responsible for TEK distribution periodically to all group members. The forward and backward secrecy are ensured within this approach triggering a TEK redistribution process after each Join or Leave procedure.

The localization-independent clustering approach that we use in our simulations belongs to dynamic clustering protocols family. The clustering is achieved dynamically, by applying a simple evaluation function whose parameters are the number of the local group members and the local dynamicity frequency (number of Joins and Leaves per unit of time). Each group member computes its evaluation function. If the number of the local group members or the local dynamicity frequency exceeds a defined threshold, a member creates with its local members a cluster and becomes a local controller. Within this approach, the TEK is distributed periodically by the local controllers to all their group members. In addition, each LC distributes a local cluster key to its local members, after each Join or Leave procedure. This clustering approach is presented by [2].

We run several simulations under Linux Mandrake 10.0, using the network simulator NS2 version ns-allinone-2.26. The simulation environment is composed of:

- area: 500*500 meters.
- number of nodes 50 - 100.
- simulation duration: 1000s.
- physical/Mac layer: IEEE 802.11 at 2Mbps, 250 meters transmission range.
- mobility model: random waypoint model with no pause time, and mode movement speed 0m/s, 1m/s and 10m/s.
- multicast routing protocol is MAODV [9]. [15] presents MAODV implementation under NS2.26.
- the member arrival follows a Poisson process (λ arrival per time unit) and the membership duration is an exponential distribution with average μ.
- the group source is node 0.
- only distribution keys traffic exists in the simulation.

Figure 7 shows the average latency for keys delivery, using the non-clustering approach, the localization-independent clustering approach and the OMCT-clustering approach. Our OMCT-clustering approach brings a profit in latency computed as 28% to 31% compared to the non-clustering approach, and 15% to 27% compared to the localization-independent clustering approach. Indeed, the non-clustering approach suffers from the "1 affects n phenomenon", the unique LC is responsible for the TEK distribution to every group member,

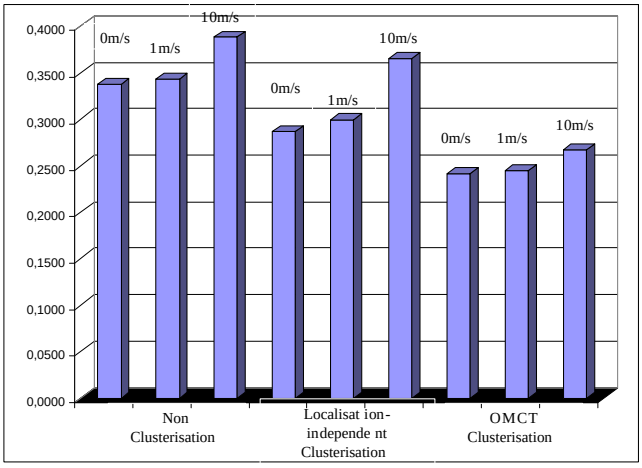

Fig. 7. Latency

after each Join or Leave procedure. Group members are also randomly localized in the ad hoc network. Thus, the delay for keys delivery increases.
The latency computed for the localization-independent clustering and the OMCT-clustering approach is smaller than latency within non-clustering approach, due to their attenuation of the "1 affects n" phenomenon. However, the most optimal latency is computed within our OMCT-clustering approach because it ensures strongly correlated clusters, each cluster being managed by a LC close to its local members which optimize valuably the keys delivery delay.

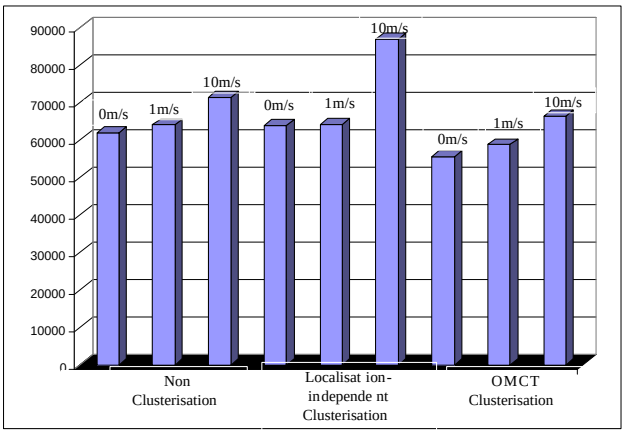

Fig. 8. Energy

Figure 8 indicates that OMCT brings also a profit concerning the required energy for keys distribution within the multicast group, computed as 7% to 10% compared to the non-clustering approach, and 8% to 23% compared to the localization-independent clustering approach.
The energy-efficiency can be explained as the latency optimization. Indeed, while forming a strongly correlated clusters within the OMCT-clustering approach, having a cohesion factor close to 1, the local controllers have to consume less energy to reach every member in the cluster.

However, the mobility factor affects the performances of our OMCT approach. This is due to the time necessary for the construction of the clusters. However, even in a high mobility environment, OMCT remains the most optimal concerning energy efficiency and average of latency.

6 Conclusion and Future Works

In this paper, we presented an efficient clustering scheme for key distribution dedicated to ad hoc networks. Our approach based on OMCT algorithm, is mobility-aware and energy-efficient. We divide the multicast group dynamically, into clusters, according to the localization and mobility of the group members. Each cluster is managed by a local controller responsible for the local key management.

Periodically, each LC verifies whether its cluster is highly correlated and consequently whether the local key distribution is optimal within its subgroup, by computing an evaluation parameter called cluster cohesion. This parameter measures the proximity of the cluster members compared to their controller, so that formed clusters are strongly correlated, thus optimizing the average latency of key distribution and the sum of energy units required for multicast messages transmission.

The OMCT algorithm will be integrated within a key management protocol approach in ad hoc networks, called BALADE [3]. This protocol delivers a fast and efficient key distribution in a sequential multi-sources multicast service, that we developed within our research team.

As future works, we will analyse the performance of our algorithms in terms of time complexity, while integrating the probability theory in these analyses. we plan also to enhance our scheme by integrating the key distribution reliability which is an important issue in ad hoc networks due to the high packets-loss rate in such environment.

References

1. H. Bettahar, A. Bouabdallah, and Y. Challal. An adaptive key management protocol for secure multicast. In *11th International Conference on Computer Communications and Networks ICCCN*, Florida USA, October 2002.
2. M.S. Bouassida, I. Chrisment, and O. Festor. An Enhanced Hybrid Key Management Protocol for Secure Multicast in Ad Hoc Networks. In N. Mitrou, K. Kontovasilis, G.N. Rouskas, I. Iliadis, and L. Merakos, editors, *Networking 2004, Third International IFIP TC6 Networking Conference*, volume 3042 of *Lecture Notes in Computer Science LNCS*, pages 725–742, Athens, Greece, May 9-14 2004. Springer.
3. M.S. Bouassida, A. Lahmadi, I. Chrisment, and O. Festor. Diffusion multicast sécurisée dans un environnement ad-hoc (1 vers n séquentiel). Rapport de recherche, INRIA, Sep 2004.
4. Y. Challal, H. Bettahar, and A. Bouabdallah. SAKM: A Scalable and Adaptive Key Management Approach for Multicast Communications. *ACM SIGCOMM Computer Communications Review*, 34(2), April 2004.
5. L. Dondeti, S. Mukherjee, and A. Samal. Secure one-to-many group communi-cation sing dual encryption. In *Comput. Com-mun.23, 17 (November)*, 1999.
6. B. Jones and D. Skellern. Hiperlan channel assignment strategies. In *Electronics Letters*, 1997.
7. L. Lazos and R. Poovendram. Energy-Aware Secure Multicast Communication in Ad Hoc Networks Using Geographical Location Information. In *IEEE International Conference on Acoustics Speech and Signal Processing*, 2003.
8. S. Mittra. Iolus: A framework for scalable secure multicasting. In *SIGCOMM*, pages 277–288, 1997.
9. E. Royer and C. Perkins. Multicast Ad hoc On-Demand Distance Vector (MAODV) routing, IETF Internet Draft: draft-ietf-manet-maodv-00.txt, 2000.
10. J. Sadowsky and V. Kafedziski. On the Correlation and Scattering Functions of the WSSUS Channel for Mobile Communications. *IEEE Transactions On Vehicular Technology*, 47(1), February 1998.
11. Y. Sun, W. Trappe, and K. Liu. A Scalable Multicast Key Management Scheme for Heterogeneous Wireless Networks. *IEEE/ACM Transactions on Networking*, 12(4):653–666, August 2004.
12. Jeffrey E. Wieselthier, Gam D. Nguyen, and Anthony Ephremides. On the Construction of Energy-Efficient Broadcast and Multicast Trees in Wireless Networks. In *INFOCOM 2000*, Tel-Aviv, Israel, March 26–30 2000.
13. Jeffrey E. Wieselthier, Gam D. Nguyen, and Anthony Ephremides. Algorithms for energy-efficient multicasting in static ad hoc wireless networks. *Mobile Networks and Applications*, 6(3):251–263, 2001.
14. Chung K Wong, Mohamed G. Gouda, and Simon S. Lam. Secure group communications using key graphs. Technical report, University of Texas at Austin, 1997.
15. Y. Zhu and T. Kunz. Maodv implementation for ns-2.26 - systems and computing engineering, carleton university, technical report sce-04-01, January 2004.