A New Two Level Hierarchy Structuring for Node Partitioning in Ad Hoc Networks

Benaouda Nacéra
Department of Computer Science
Sétif Automatic Laboratory
Faculty of engineering Science
Sétif, Algeria
nacbenaouda@univ-setif.dz
nacbenaouda@gmail.dz

Guyennet Hervé
Laboratory of engineering Science
University of Franche Comté
Besancon, France
guyennet@lifc.univ-fcomte.fr

Hammad Ahmed
Laboratory of engineering Science
University of Franche Comté
Besancon, France
hammad@lifc.univ-fcomte.fr

Mostefai Mohammed
Department of automatic
Sétif Automatic Laboratory
Faculty of engineering science
Sétif, Algeria
mostefai@univ-setif.dz

ABSTRACT

A big challenge in designing a distributed application is the partitioning which consists of optimizing the system component inter-communication. In this paper we introduce a new approach of *Ad Hoc* network group formation based on two level hierarchy structure: many groups with one leader per group at the first level and a super-leader for the whole network at the second one. We worked on the minimization of the communication between nodes by the choice of decent metrics in group formation as well as in leader and super-leader elections. Moreover the designed algorithm is validated by simulation results and then verified by model checking. In addition, a re-organization strategy and the principle of communication protocols will be presented.

Keywords

Communication, Distributed System, Graph Partitioning, Group Formation, Partitioning

1. INTRODUCTION

wireless technology and Internet development made applications be increasingly distributed and used in different sectors(industrial, medical, commercial, etc. . .). But many problems have raised such as scalability, coverage area, fault tolerance and routing information.

In designing a distributed application, one must answer the following fundamental question: - How to organize and share or partition processing, in order to optimize, the execution time, as well as the use of resources. An *Ad Hoc* network, being nothing else than a distributed system, the issue of partitioning must take place in any design of application on this type of network. Our objective in this paper concerns the development of an approach structuring the network in groups, that aims to minimize communication between the network nodes, this being the main aim of partitioning. This paper is organized on two parts: the first part presents partitioning theory in distributed applications, group structuring in *Ad Hoc* networks, the second part concerns our contribution in this issue. The structure and the management of *Ad Hoc* networks is presented. Control parameters are introduced to measure the performance of the structure. Some simulation results show the efficiency of the proposed organization. A previous work section is followed then and, finally, formal verification of the entire network is proceeded using model checking.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
Copyright 200X ACM X-XXXXX-XX-X/XX/XX ...$10.00.

2. PARTITIONING IN DISTRIBUTED APPLICATIONS

Partitioning means that we have to define the best strategy to distribute calculations on different units. These units must then share the results or communicate and synchronize between them. When the distributed system comprising several different units, the overall execution time of the application is influenced by the number and duration of communications between these different units. More are communications held, more the execution time is increased. On the other hand, the issue of process partitioning may be put back to a mathematical problem in graph theory: partitioning graph. Indeed a distributed system can be seen as a graph composed of several nodes. These nodes represent the process units, and arcs between nodes are links or communications between these units. We explain below the principle of the graph partitioning.

2.1 Graph Partitioning

The graph partitioning (PG) consists of dividing all summits of a graph into several subsets, so that the number of these subsets peaks are equal or substantially equal and at the same time, minimizing cut edges valuations. This problem has been widely studied for over thirty years and has many applications namely: design of VLSI integrated circuit, in parallel computing [4], in finite elements optimization method [7], etc. We define bellow in a formal manner the Graph Partitioning.

2.2 Graph partitioning definition

Let $G = (U, Q)$ a graph where U is the set of summits u_i and Q is the set of edges (u_i, u_j). In some cases, the graph peaks may be associated with spatial coordinates determining their relative positions in the d-dimensional space R^d. A k-partitioning of this graph is an application $F : U \longrightarrow [1, k]$ of peaks to subsets (called partitions) U_1, U_2, , U_k ($F(u_i) = j$ if and only if the summit u_i is in the partition U_j), where $\bigcup_i u_i = U$ and $U_i \bigcap\ U_j = \phi$, $(i \neq\ j)$.

2.2.1 Partitioning techniques

There are two major methods of partitioning. The first concerns a local solution trying to iteratively, converging towards a better solution starting from an initial solution; we quote in local methods : Kernighan-Lin and Fiduccia-Mattheyses heuristics. The second is more realistic trying to take into account the underlying graph properties. RCB (Recursive Coordinate Bisection), RGB (Recursive Graph Bisection), and RSB (Recursive Spectral Bisection) are global methods.

2.2.2 Existing methods limits

The graph partitioning, is resolved today by the use of effective heuristics, which are not necessarily relevant to the deployment of parallel applications on heterogeneous platforms. Furthermore, experiments showed that some heuristics are much more effective than others depending on the nature of the concerned graph, including those which do not use the metric based on the cut edges. Indeed, often spectral approaches are more effective because they include all the problem parameters. The study of the deployment of parallel applications on heterogeneous platforms requires to study the nature of networks that underly them : Indeed, methods of cutting edges, use rigorous mathematical models, which consider abstract graph. And then they do not benefit the particularity of graphs representing heterogeneous and dynamic networks. This feature facilitate even partitioning.

3. ORGANIZATION IN GROUPS OF DISTRIBUTED SYSTEMS

Several models exist for distributed systems (example: hierarchy, groups, ...). Organizing in groups a system consists of putting together some objects, materials or machinery in groups cooperating and communicating.
The group concept allows to define a group of entities as a single virtual entity: it assigns the same name to each member of a particular group, and communicate with them using the same address[9]. In general, in a distributed system, communication between nodes, objects or processes transfer, and decisions are difficult problems that cannot be resolved for all the network nodes. To solve these problems, one can use the formation of entities subsets called groups [2], clusters[3][14], partitions [16] or territories [15]. These groups are composed of members. In each group, one member plays a particular role, and is called leader[11][2], manager[6] interconnection point[17], local coordinator [3]. This one is responsible for communication between various members or levels, receiving information and referring to other members, and overseeing the internal organization of the group. The notion of group is extended today by the definition of a two level hierarchy structure and n level hierarchy structure[13]. A two level hierarchy structure requires, in addition to groups formation, and the choice of a coordinator for each group, the election of an overall coordinator that is called global coordinator [3] or Super-leader[2], playing the role of interconnection point of all the network groups. In practice, each application gives rise to new concepts concerning properties associated with groups.

4. GROUPS FORMATION IN AD HOC NETWORKS

The group formation process corresponds to a virtual cutting of the network netW, into groups geographically close $(G_1, G_2, ..., G_k)$. These groups are not necessarily separated such as: $netW = \bigcup_{i=1}^{k} G_i$. A particular node in each group is elected to represent the group. This node is called "group Leader": GL". The election of GL is usually based on a specific metric, the group is then built of GL and all nodes that are attached to it. The principle of the algorithm training groups, is to find a set of interconnected groups, which cover the entire population of nodes. This algorithm is evaluated in terms of groups stability, compared to nodes movements, and depending on the produced groups number. One group formation objective is the information on the network topology maintenance, and reduction of the overhead generated by the route discovery.
Many group formation algorithms have been conceived, widely studied and classified. Generally, the term "clustering" is used in a great number of publications and theses that speak about clustering on the basis of: mobility[11][12][14], signal power [11], node weight[11][12], density[14], distance between nodes[3], lowest identifier[12]. Our group formation approach is the result of the study of some of these algorithms.

5. PROBLEMATIC: GROUPS FORMATION AND PARTITIONING IN AD-HOC NETWORK

An *Ad Hoc* network is a particular distributed system. In addition to the communication requirements imposed by applications, is added the communication imposed by routing. Because a communication that takes place in a wired network into a jump using a cable is done in an *Ad Hoc* network into several jumps when nodes are remote, in that case, *Ad Hoc* nodes constitute relays between the transmitter and receiver. This induces more interaction between nodes, therefore, more time is consumed in the execution of the application. In terms of partitioning, we must react in an *Ad Hoc* network at two levels: the first one is the application level which can be solved as in any other distributed system. The second level, is located in routing which is closely dependent on the organization that is in our case the group formation.

It is in the intended meaning to establish our approach of group formation: We tried in our protocol to minimize communications at the routing. We have established an organization in groups taking as metrics, geographical location, and making cluster head (local and overall) in order to minimize communication between nodes. We explain below our approach.

6. OUR APPROACH

6.1 Used parameters

1. Coverage field
   Every node $n_i(x_i, y_i)$ of the network, creates around itself, a coverage field that spreads on the disk delimited by the circle whose equation is, :
   $$(x - x_i)^2 + (y - y_i)^2 = D_i^2$$
   , where D_i is the range of n_i. A node $n_j(x_j, y_j)$ is within reach of $n_i(x_i, y_i)$ if it verifies the following inequality:
   $$(x_j - x_i)^2 + (y_j - y_i)^2 = D_i^2$$

2. Euclidian space Distance
   Let's consider two nodes, n_i and n_j , located, in euclidian space, by the respective coordinates x_i, y_i and x_j, y_j. The distance between n_i and n_j is given by the relation:
   $$D(n_i, n_j) = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}$$

6.2 Algorithm description

(A two level Hierarchy Structure for *Ad Hoc* Networks (HSL-2-AN)) The group formation algorithm that we conceive aims to create a two level hierarchy structure, that's: Several groups with a local coordinator or a leader for each of them, and a global coordinator or super-leader for the management of the whole network. This structure is motivated by the fact that we aim to define communication protocols that simulate the case of a cellular network. A communication in the group comes through the group leader. A communication between different group members comes through the super-leader. The best situation is the one where, in each group, the group leader, is reached in one jump by its members, and the super-leader reached in one jump by every leader. Our algorithm includes three stages, and takes as a basis on the following assumptions: the network nodes constitute a related graph, every node has a unique identifier number, communications are bi-directional and FIFO, all communication channels are reliable (no lost or duplicated message), messages arrive to destination at the end of a finite time, every node has some properties(energy quantity, signal power, mobility, resources quantity), the topology is dynamic and nodes movement is random, a localization mean, as a GPS, is used in the network.
In this context, our algorithm takes place according to the following stages:

```
Algorithm HSL − 2 − AN;
  Begin
    Groups_Formation;
    Leader_Election;
    Super − leader_Election;
  EndAlgorithm
```

Where, groups_formation, Leader_election and super-leader election are our algorithm stages. We describe, below each stage.

1. Groups Formation

```
Algorithm Groups_Formation;
  Begin
    k = 1; #k is the number of groups#
    For each node n_i in the nodes list Do
      Addition of n_i to G_k;
      Suppression of n_i from nodes list;
      For each node n_j in the nodes list (j) Do
        if sqrt((x_j − x_i)^2 − (y_j − y_i)^2) < D_i
          Then
            Addition of the node n_j to G_k
            Suppression of the node n_j from
            the nodes list
        Endif
      Endfor
      k = k + 1;
    Endfor
  EndAlgorithm
```

   This stage permits to put the nodes that are in the same region in the same group. The second stage consists of the choice of a leader for each group according to the algorithm that we present below.

2. Leader election The leader is the node of which the average value of distances between it and the other group nodes is minimal. In order to make the group leader election, every group member executes the following algorithm:

```
Algorithm Leader_election;
  Begin
    For each node n_i ∈ G_k Do
      K = 0;
      Somme_distance = 0;
      For each node n_j ∈ G_k(j ≠ i) Do;
        somme_distance = somme_disatnce+
        sqrt((x_j − x_i)^2 − (y_j − y_i)^2);
        k = k + 1;
      EndFor;
      Moy_distance[i] = somme_distance[i]/K;
    EndFor;
    Min(moy_distance[i]); ♯let's suppose that
          Moy_distance[m] is the minimum; ♯
    Election of n_m as the group leader;
    If it exists two (ormore) minimums
      Then election of the one that has the
          lowest identifier
    Endif;      ♯ The elected leader broadcasts
    in the network its identifier; ♯
  Endalgorithm;
```

   At the end of the second stage, formed groups are separated; but group members are not necessarily in the range of their leader.

3. The super-leader election
   The super-leader is the node that has the maximum of leaders in its coverage field. It is elected according to the following algorithm.

```
Algorithm super − leader_election;
  Begin
  For each node ni (≠ leader) Do
```

$\underline{For}$ *each node* n_j *in the list of leaders Do*
$\underline{if}(x_j - x_i)^2 - (y_j - y_i)^2 \leq D_i^2$
$\underline{Then}$#*increment of the number of leaders that are in the node range*
$Nb\_leader[i] = nb\_leader[i] + 1;$
$\underline{Endif}$;
$\underline{EndFor}$;
$\underline{EndFor}$;
$Max(nb\_leader[i]);$ # *Let's suppose* $nb\_leader[m]$ *is the maximum*#
Election of n_m *to be super − leader*;
$\underline{If}$ *it exists two (or more) maximums*
$\underline{Then}$ *Election of the one that has the lowest identifier*; ♯ *only one super − leader is elected in the network*♯
$\underline{EndIf}$;
♯ *The super − leader broadcasts its identifier number . in the network*.♯
$\underline{Endalgorithm}$.

6.3 The network structure at the end of HSL-2-AN

In order to illustrate the network structure at the end of HSL-2-AN, let's resume notations used in [13].
[13] works on wireless Sensor Network (WSN) and uses an H level tree structure, where H is the tree depth.
HSL-2-AN is a two level hierarchy structure, and may be considered as a particular case of [13] structure, *where* $H = 2$. Let's see what gives HSL-2-AN, for ten nodes composed network, in 500×500 topology and a $range = 165$. The obtained results are:
Number of groups=3 and Super-leader node=9
$G1 = \{0, 5, 7, 9\}$, $G2 = \{1, 8\}$, $G3 = \{2, 3\}$, $G4 = \{4, 6\}$, leaders are, respectively, nodes: 5, 8, 3 *and* 6.
Inspired by [13] we may Put
G_k^h the group number k at level h.
$Leader(g)$ returns the group g leader.
$leader^{-1}(P)$ returns the group which has P as leader, and NULL if P is a leaf node.
For instance: $G_2^3 = 4$ and $Leader^{-1}(4) = G_2^3$.
For our example shown in (Fig1)we may write: G_1^1 represents the group number 1 at level 1.
$G_1^1 = \{9, 5, 8, 3, 6\}$
$Leader(G_1^1) = 9$
$G_1^2 = \{0, 5, 7, 9\}$ and $Leader(G_1^2) = 5$
$G_2^2 = \{1, 8\}$ and $Leader(G_2^2) = 8$
$G_3^2 = \{2, 3\}$ and $Leader(G_3^2) = 3$
$G_4^2 = \{4, 6\}$ and $Leader(G_4^2) = 6$
$0, 7, 1, 2, 4$ are leaf nodes, therefore the application of $Leader^{-1}$ to each of them returns NULL: for instance $Leader^{-1}(7) = NULL$, Fig1 illustrates this vision that permits to, easily, extend HSL-2-AN for too large networks.

7. THE NETWORK MANAGEMENT

Such a structure requires management operations in order to maintain it. Besides a re-organization operation (the group formation algorithm) is periodically triggered by the super-leader, the main protocols that we foresee are: A member addition/deletion, mobility, communication and tolerance to the breakdowns. We present below communication protocols principle.

- Communication

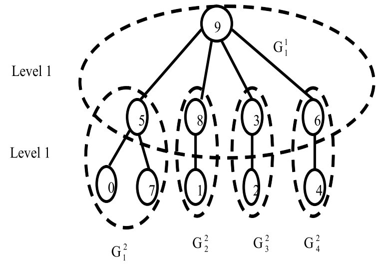

Figure 1: The network structure at the end of HSL-2-AN

The defined group structure has been conceived in order to make the communication protocols optimal. These protocols are conceived as follows:

1. Communication between the same group members comes inevitably through the group leader. it is optimal when it is performed in two jumps: One jump for $Member1 \longleftrightarrow Leader$ and one jump for $Leader \longleftrightarrow Member2$.
2. The communication between different group members comes inevitably through the super-leader. It is optimal when it is performed in four jumps: Two jumps for $member1 \longleftrightarrow leader1 \longleftrightarrow Super-leader$ and Two jumps for $Super-leader \longleftrightarrow leader2 \longleftrightarrow member2$

8. SYSTEM PERFORMANCE CONTROL

We use three factors to measure our structure performance: the group cohesion factor $Group\_cohesion_k$ as the percentage of nodes in coverage field of their leader in the group number k (G_k), the network cohesion factor Network_cohesion as the percentage of leaders in the coverage field of the super-leader, and Taux_group_cohesion the percentage of groups in cohesion. According to these definitions, the group_cohesion factor has a different value in each group, but each of network_cohesion and Taux_group_cohesion factors is calculated for the whole network. In order to define these parameters, let's suppose A_k, B, C, Nb_k and NBG such that :
A_k=number of members that are in the range of the leader in G_k
Nb_k=The number of nodes in G_k
B=number of leaders that are in the range of the super-leader
C=number of groups that are in cohesion
NBG=The number of groups(or leaders)in the network.
We may write:

$$Group\_cohesion_k = \frac{A_k}{Nb_k}.100$$

$$Network\_cohesion = \frac{B}{NBG}.100$$

$$Taux\_group\_cohesion = \frac{C}{NBG}.100$$

Let's suppose that SGC, SNC and STGC are respectively Group_cohesion, Network_cohesion and Taux_Group_Cohesion thresholds. At any time, these parameters can be calculated. We retain three different cases for these parameters as meaningful cases for the network, that are:

1. In cohesion, if,
   Taux_group_cohesion $\geq$ STGC and Network_cohesion $\geq$ SNC.
   This situation corresponds to the fact : the majority of the group members are in their leader's range, and the majority of leaders are in the super-leader's range.

2. In strong cohesion, if,
   - Taux_group_cohesion =100 (with group_cohesion=100 in all groups) and Network_cohesion > SNC.
     This means that all groups members are in the range of their leaders , and the majority of leaders are in the super-leader's range.
   - Taux_group_cohesion=100
     (with SGC $\leq$ group_cohesion < 100 in all of groups) and Network_cohesion=100.
     This implies that in each group, the majority of members are in the range of their leader; and in the network all leaders are in the range of the super-leader.

3. In absolute cohesion, if
   Network_cohesion =100 and Group_cohesion =100 for each of the network groups.
   All leaders are in the super-leader's range, and members of each group are in their leader's range.

   Initially, the Group_cohesion value in every group is equal to its initial value $init\_group\_cohesion_i$, and Network_cohesion is equal to its initial value Init_Network _cohesion, with the change of the network topology caused by the nodes mobility these two values can change. Periodically, every group leader calculates group_cohesion value and sends it to the super-leader. This last calculates then, Network_cohesion and taux _group_cohesion, and decides, according to these factors values to broadcast the re-organization message or not.

9. SIMULATION RESULTS

In order to have simulation results of our algorithm, we use the version 2.31 of the network simulator NS2. Our objective is to study our performance parameters network_cohesion, group_cohesion and taux_group_cohesion.

- Simulation principle
  First of all, the most important question to which we must respond is: why did we choose the study of performance parameters in simulation? the response is that we focus in this paper on partitioning which is expressed by these parameters, our algorithm is not only a structuring in groups approach, but it is in addition, an *Ad Hoc* network node partitioning approach, this matter has been explained and justified in section5. At this initial stage of our work, the algorithm takes place as follows: in its three steps, you can remark that we use only nodes coordinates X_i and Y_i; in practice these coordinates are obtained by the use of a GPS. In simulation we obtain them from the trace file. We used the new format for trace file, defined in NS2, this format gives a great number of information about the node, among others, nodes coordinates. After making a specific scenario run, we accede to the trace file, extract nodes coordinates and apply HSL-2-AN to theses coordinates. A procedure calc_affich_parameters has been integrated to HSL-2-AN to calculate the number of the formed groups, the number of nodes in each group, leaders' identifiers, the super-leader identifier and performance parameters values. It is only a simulation that shows us what would be these parameters at a given instant. For more information, we used the TCL programming language to explore the trace file.

Table 1: Group_cohesion = F(NGroup)

Group Number	Group Cohesion
1	77
2	93
3	90
4	80
5	88
6	92
7	93
8	0
9	75
10	71
11	66
12	71
13	0

- Obtained results
  In order to study the evolution of our cohesion parameters we worked with 100 nodes. To facilitate, we supposed also, that all the network nodes have the same range. In addition, we consider that a group is in cohesion if 80% of its members are in the range of their leader (SGC = 80). We consider also, that SNC = 80 and STGC=80.

Table1 discussion : the value of D for each node is 100. Table1 shows group cohesion value in each group. It is equal to zero for the groups number 8 and 13 because each of them has only one member which is the leader of the group. In that case Network_Cohesion=30 and Taux_Group_Cohesion=46. According to the threshold values that we consider for cohesion parameters, the network isn't in cohesion and must be re-organized (Fig2).

Table2 discussion : Table2 values and the corresponding graph (Fig3) show that the number of groups decreases with D. Network_Cohesion and Taux_Group_Cohesion increase with D; and until D=250 these two parameters stop at 100. The value 66 of Taux_Group_Cohesion when D is equal to 350 is due to the fact that the number of groups is reduced to 3 and one of these groups is one member composed, it's the reason why, Taux_Group_Cohesion decreased. We can observe the situation of the network When D=250 (Table3, Fig4).

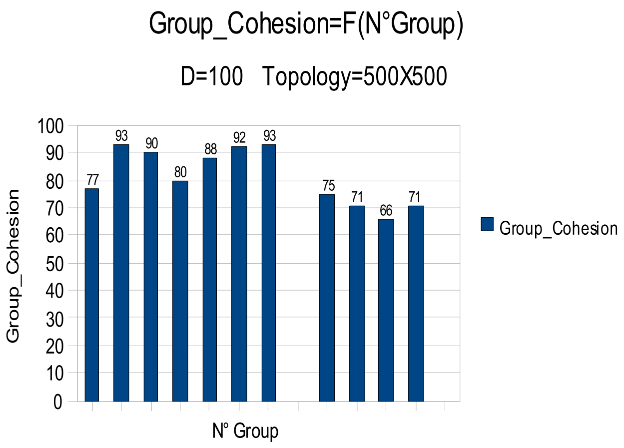

Figure 2: Group_cohesion Values

Table 2: Evolution of cohesion parameters according to D

D	Number of Groups	Network cohesion	Taux group cohesion
10	97	2	0
20	77	3	0
30	61	6	0
40	44	9	2
50	36	11	11
100	13	30	46
150	7	42	85
200	6	83	66
250	4	100	100
300	4	100	100
350	3	100	66

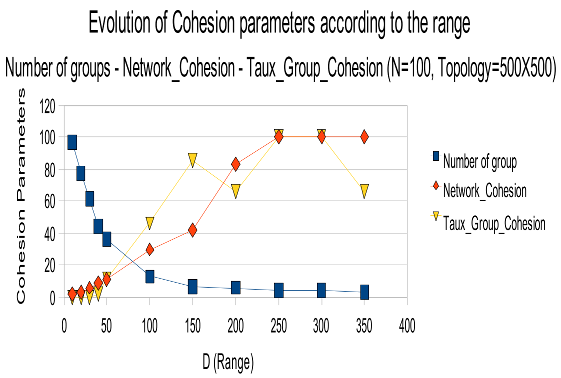

Figure 3: Evolution of Cohesion parameters according to the range

Table 3: Group_cohesion = F(NGroup) N=100, D=250

Group Number	Group Cohesion
1	97
2	95
3	95
4	91

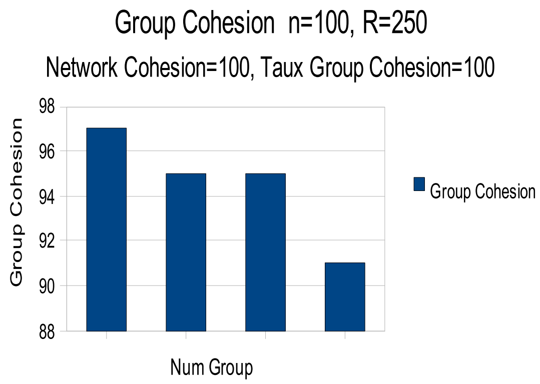

Figure 4: Group_Cohesion values for N=100,D=250

When the number of nodes=100 and D=250, we have: the number of the formed groups is four (4), Network_Cohesion = Taux_Group_Cohesion = 100 (see Table2). Table3 and Fig4 show the value of Group_Cohesion in each group That's: 97, 95, 95, 91. According to the threshold values that we supposed, the network is in strong Cohesion or almost in absolute cohesion.

10. FROM PREVIOUS WORK TO HSL-2-AN

HSL-2-AN is a result of the study of a great number of *Ad Hoc* networks structuring in groups; among others, we can cite [11],[3].

[11] tries to establish a one level hierarchy structure: many groups with one leader per group. On the basis of the node mobility, signal power and a diffusion algorithm to form groups and elect leaders. The principle of the proposed algorithm is strong, but the use of the one level hierarchy structure is insufficient: it does as if we reduce the number of nodes because after this structuring, a group of nodes becomes punctual and is considered as a single node. Maybe, when passing to a greater number of nodes this structuring would not have a big significance.

[3] designs an algorithm to establish a two level hierarchy structure oriented to the group key distribution. In this work, authors use the distance between nodes as criteria to form groups, and speak about connectivity, cohesion but no condition is imposed to make communication optimal, notably in the super-leader election. Authors present then simulation results based on the consumption of time and energy. The particularity of our approach is that the basis of the thought in designing HSL-2-AN is to optimize communication: in forming groups, in electing leaders and super-

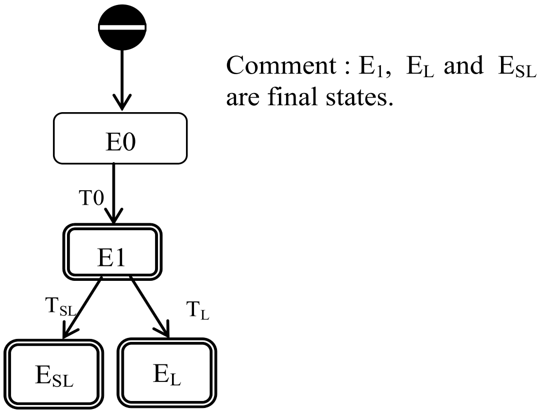

Figure 5: A node UML State transition diagram

leader. The network performance is measured on the basis of this fact. Indeed, we defined the case of optimal communication and established three specific parameters to measure the performance of the network on the basis of these parameters: communication takes place using the lowest number of jumps when the network is in absolute cohesion.

11. HSL-2-AN MODEL CHECKING

11.1 Model checking principle

Our approach is built on the fact that the *Ad Hoc* network is composed of several nodes but HSL-2-AN runs on each node, regardless other nodes; because each node uses only, the network nodes coordinates that can be obtained by a GPS. Moreover, according to the assumptions that we mentioned in paragraph6.2, messages that are sent from leader and super-leader, in the algorithm, arrive to destination at the end of a finite time, and no exception case is taken in consideration. For theses reasons, it is good enough to make the formal verification for one node and to deduct for the entire network.

1. UML State transition diagram[10] (Fig5).
   - States.
     E_0: without status (the node has none status)
     E_1: G_k group member
     E_L: G_K group leader
     E_{SL}: The network super-leader
   - Transitions.
     T_0:group formation algorithm (the first stage of $HSL-2-AN$)
     T_L:Leader election algorithm (The second stage of $HSL-2-AN$)
     T_{SL}:super-leader election algorithm (The third stage of $HSL-2-AN$)
2. The system Kripke structure (Fig6).
   - Kripke structure mathematical formulation.
     $SK\_node = \langle Q, Q_0,\ E,\ T,\ \lambda\rangle \quad Where,$
     $Q = \{E_0,\ E_1,\ E_L,\ E_{SL}\}$
     $Q_0 = \{E_0\}$
     $E = \{T_0,\ T_L,\ T_{SL}\}$
     $T \subseteq\ Q\times\ E\times\ Q = \{(E_0, T_0, E_1), (E_1, T_L, E_L), (E_1, T_{SL}, E_{SL})\}$
     $\lambda = \{E_0 \rightarrow\ Without\ Status,\ E_1 \rightarrow\ Member\ of\ G_k,\ E_L \rightarrow\ Leader\ of\ G_k,\ E_{SL} \rightarrow\ Superleader\ of\ the\ network\}$
3. Our system qualitative properties
   In order to verify our system, we defined two properties that the system must verify: a safety property and a vivacity property. The chosen properties concern a fundamental aspect of our system and if checked, they are enough to ensure that HSL-2-AN runs correctly. we define below the two properties.
   - Safety property definition
     "Being in the state without status, a node doesn't remain indefinitely in this state"
   - vivacity property definition
     "Being in the state without status, if the node will not be the network super-leader, nor a group leader, hence it will be a group member."
     This property expresses the fact that the node must have a status at the end of HSL-2-AN.

   To write these properties in linear temporal logic(LTL), let's suppose $P,\ P_M,\ P_L\ and\ P_{SL}$ atomic propositions, respectively defined as "to be without status", "to be a group member", "to be a group leader", "to be a super-leader"; and ∅1, ∅2 respectively, the defined safety and vivacity properties. In LTL, we may write:
   $\emptyset\ \equiv\ P\ \Rightarrow\ \diamond(\neg P)$
   $\emptyset\ 1 \equiv\ P \Rightarrow (\diamond(\neg\ P_{SL} \wedge\ \neg\ P_L) \Rightarrow\ \diamond\ P_M)$
4. Interpretation and verification deduction.
   The Kripke structure represents all behaviors that are accepted by the system. On the basis of this automaton, safety property is checked because the passage from the state "without status" to the state "member" is reliable, this transition is carried out at the first stage of HSL-2-AN execution, that concerns groups formation on the basis of nodes coordinates.
   Vivacity property is, also checked, because it's clear on the automaton, that the node has three issues from the initial state: either to be a member of a group and stop there, or to be a group leader as result to HSL-2-AN second stage and stop there, or to be the network super-leader as result to the third stage of HSL-2-AN. Therefore, we conclude that the two properties expressed by ∅1 and ∅2 are verified and the node does not remain, indefinitely in the state "without status", and it will have a status that may be: a member of a group or a group leader or the network super-leader. This fact expresses that HSL-2-AN runs properly. Given that any node checks the defined safety and vivacity properties, and given that the network nodes run HSL-2-AN independently of each other, the two properties are assumed in the whole network.

12. CONCLUSION

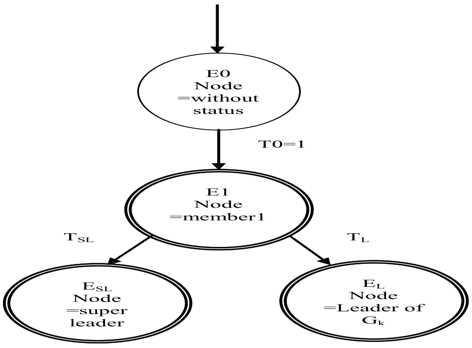

Figure 6: SK_node:A node Kripke structure

We presented in this paper HSL-2-AN which is a new two level hierarchy structuring algorithm for *Ad Hoc* networks. we aimed in designing it to optimize nodes intercommunication. Hence, our approach may be considered, at a time an approach to structuring in groups and an approach of partitioning applications distributed on *Ad Hoc* networks. This matter has been well explained and justified.
We established performance parameters that allow identification at all times the network. These parameters are calculated periodically, and may be used by the network manager to require particular geographical positions or a range of mobility of nodes to maintain a desired cohesion in the network. In addition to our previous proposition, we tried in this paper to validate our approach by some simulation results, and to verify our algorithm by model checking. As a perspective, we intend to use our algorithm for the management of WSN[13][12] which are particular *Ad Hoc* networks, and used in several applications in various fields: domotics[12], military, and especially for surveillance in industrial maintenance[1].

13. REFERENCES

[1] N.Benaouda, H.Guyennet.and A.Benaouda, *Simulation d'un scénario de e-maintenance préventive utilisant les services web .NET.*, in IEEE SETIT'2005, electronic edition, abstract page 235, Tunis, 2005.

[2] N.Benaouda, H.Guyennet, A.Hammad, A/H Benaouda, M. Mostefai, *Modélisation par les groupes d'un réseau sans fil Ad Hoc*, Cip'2007, electronic edition, abstract page 108, Sétif, Algeria, 2007.

[3] M.S.Bouassida, I.Chrismen, O.Festor, *Group key management*, International Journal Of Network Security, IJNS, 2006.

[4] B.Chamberlain, *Graph partitioning algorithms for distributing workloads of parallel computations*,Technical Report UW-CSE-98-10-03, University of Washington, 1998.

[5] E.M.Clarke, O.Grumberg and D.A PELED, *M*odel checking, MIT Press, Cambridge, Massachusetts, 2001.

[6] D.J. Evans W.U.N. Butt:*Load Balancing with Network Partitioning Using Host Groups, Parallel Computing*,20, 325-345, 1994.

[7] P.O.Fjallstrom, *Algorithms for graph partitioning: A survey*, Linkoping Electronic Articles in Computer and Information Science, 3(10), 1998.

[8] M.R.Garey and D.S Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman Co., 1979.

[9] H.Guyennet, Jean-Christophe Lapayre: *The Group Approach in Cooperative Work and in Load Balancing* , Journal of Parallel and Distributed Computing Practices, Vol 2, Nř3, p285-297, 2001.

[10] A.Hammad and H.Mountassir, *Heuristics to verify LTL properties of hierarchical systems*, in VECoS'2008, 2nd Int. Workshop on Verification and Evaluation of Computer and Communication Systems, eWiC - electronic Workshops in Computing, Leeds, UK, July 2008, British Computer Society.

[11] N.Kettaf, A. Abouaissa, P. Lorenz, H. Guyennet, *A Self Organizing Algorithm for Ad hoc Networks*, 10th IFIP Int. Conference on Personal Wireless Communication, PWC05, pp115-124, Colmar, Aug 2005.

[12] M.Lehsaini, H. Guyennet, and M. Feham, *CES : Cluster-based Energy-efficient Scheme for Mobile Wireless Sensor Networks*, chapter in the book Wireless Sensor and Actor Networks II (Boston Springer), vol.264, pp.13-24, May 2008.

[13] W.Massri, *QoS requirements mapping in TDMA-based Wireless Sensor Networks*, Ph.D. in Computers from the University Paul Sabatier (Toulouse 3), 2009, Toulouse, France.

[14] N.Mitton, *Auto-organisation des réseaux sans fil Multi-sauts a grande échelle*, Ph.D. in Computers and networks, 2006, Insa-Lyon, France.

[15] P.G. Raverdy:*Gestion de Ressources et Répartition de charge dans les systèmes hétérogènes a Grande échelle*, PHD Thesis, Paris VI, 1996.

[16] E.Rosti et al. : *Robust Partitioning Policies of Multiprocessor Systems*, Performance Evaluation North Holland, 19(2-3), 141-165, 1994.

[17] J.G. Vaughan: *A Hierarchical Protocol for Decentralizing Information Dissemination in Distributed Systems*, The Computer Journal, Vol. 38, 1, 57-70, 1995.