Improving Data Accessibility in Mobile Ad Hoc Networks

S. Moussaoui moussaoui_samira@yahoo.fr
M. Guerroumi guerroumi@gmail.com
N.Badache Nbadache@wissal.dz
USTHB, Faculty of Electronic and Computing
Computing Department, Algiers

Abstract:
Improving data availability in mobile ad hoc networks is very important issue since the ultimate goal of using these networks, is to provide information access to mobile nodes. Make data available in such environment where bandwidth, battery power and memory size are limited, is very difficult issue

In this paper we propose tow data replication methods for improving the availability of data. These methods executed under two main phases. In the first phase we create the primary replicas, and in the second phase we redistribute these replicas dynamically. The first method creates the primary replicas for each new data created in the network or received from fixed network. The second method creates the primary replicas only for the important data. The simulation experiments show that the data availability is improved and the traffic is reduced.

Key words: Replication, Ad hoc mobile network, data availability, data access frequency, primary replicas Process.

1-Introduction:

Mobile Ad hoc Networks are formed only by mobile hosts which connected only by wireless links with no supporting fixed infrastructure or central administration.
The mobile hosts in this environment moves freely which favourers the frequently disconnections. The data replication technique allows to improve the data availability, it consists to create many copies of the same resource on the different mobile hosts.

Taking into considerations the inherent characteristics of this environment, we propose in this paper, two data replication methods. These methods present two main phases. In the first phase, the first method creates the primary replicas of each new data arrived or created in mobile ad hoc network using a hop counter. In the second method the primary replicas are created only for the important data. In the second phase we redistribute the replicas in order to cure the dynamic changes of topology and to better reply to the evolution of users' needs.

The remainder of the paper is organized as follows. In section 2, we introduce some related works and their inconvenient. In section 3, we describe our environment and its hypotheses. In section 4, we describe our proposed methods. In section 5, we show the results of simulation experiments. Finally, in section 6, we summarize this paper.

2- Related works: Several strategies of replication have been proposed for the fixed environments [22], [19], [18], [20], [21] where the access failures are not frequently produced and where the material resources are more important. These strategies improve the accessibility and reduce the load around one or several servers.

Other solutions of replication are proposed for the mobile environment [8], [9], [10], [11]. These latter usually suppose the existence of fixed server. This hypothesis is constraining for autonomy of the Ad Hoc Mobile Network.

T. Hara proposed in [1] three methods to assign the replicas to mobile nodes in an ad hoc mobile network in for improving the availability data. These methods are founded on the hypothesis that all the nodes of the network know the access probabilities to each data in the and these probabilities do not change.
These methods consider acceded data only in reading. In the method SAF, several replicas of the same elementary data can be found on neighbouring nodes. Thus, there are other data which are not assigned due to space fault. The method DAFN tries to remedy this redundancy. The algorithm of replication considers the set of frequencies of neighbours in order to avoid redundancies. The third method DCG shares replicas in larger groups of mobile hosts and limits the redundancy within the group. In this method, each node must periodically broadcast its access' characteristics, which gets a large the traffic. In spite of all these improvements, the problem of redundant replicas between the neighbouring nodes is still posed.

In [4], T. Hara extends the three methods proposed in [1] supposing the periodically updated of data items. These methods replicate the data items on the mobile nodes using the data access frequencies, the time remains until each data is updated, and the topology of network.

In [5], [6], T. Hara and S. Kumar proposed an extension of these three methods by supposing the randomly updates of data items.

T. Hara and others proposed a new method (DCG-S1) [2], derived from the one proposed in [1]. In this method, the authors used the links stability's notion, for constructing stable groups. This replication technique allows to have a better availability of data. But the traffic produced stays high.

Yan and Cao [3] proposed two replication data methods, the data are maintained by fixed servers and mobile nodes can create replicas for these servers. The first method "CacheData" allows the intermediary nodes between the access caller and the server to create replicas for serving the future demands. The second method "CachPath" allows the mobile nodes to store Paths towards the data and to use them for reorienting the future demands.
In [7], we find a method of data replication owing to a detection of eminent partitioning of the network. In this case, the data of a partition are copied on the other one to

maintain the accessibility to data after the division of the networks. In This method the authors suppose that the nodes in the network are organized into mobility groups. However, this assumption is really rarely found.

In our proposed solution we attempt to keep the various advantages of the preceding solutions, and we response to drawbacks in which mainly the traffic induced which stays one of the major cases.

3- Environment model and hypotheses:

The considered environment is an ad hoc mobile network, where each mobile node can cooperate for constructing the common cache, by the sharing of its memory space with the other nodes. The communication links of nodes are maintained since these are found in the same of radio communication range. These links are bidirectional.

A mobiles hosts can create replicas and maintain them locally. It can generate new data (original data and share them with the other users. It can also create locally a data access paths ("PathData") which allow a quicker access for distant data.

In this environment, a mobile node can eventually receive data from a static network in case of connection. We consider in this first phase of our work, the data only acceded only in reading. We can found several applications where the update of data doesn't present a major interest. For example, the information concerning a region can be delivered in the form of a map indicating the different routes and establishments like hospitals, airports, restaurants, petrol stations, hotels,…etc.

In this environment, we suppose that:

- Each mobile node is designed by a unique identifier N_i, such as $1\leq i \leq NT$, where NT is the total number of the mobiles nodes constituting the ad hoc network.
- Each datum created by a mobile node is associated to a unique identifier D_{ij}, where “i” is the identifier of the node which created these data and “j” is the sequential number of data creation. The data received from static network keep their original identifier and they mustn't be codified in this way.
- Each mobile node has a limited memory space, for maintain locally the replicas, the original data and the data access paths.
- A mobile node periodically sends to the neighbouring nodes its information concerning the access characteristics of the local data.

For each mobile node, each acceded datum is characterised by two types of access frequency:

- An external access frequency which represents the access rate of the mobile node to external data,
- An internal access frequency which represents the access rate of the mobile node and all other mobiles nodes to internal data which is local to the node.

4- Replication Data Method:

We propose a distributed and decentralized algorithm of dynamic data replication on an environment composed of mobiles hosts connected by wireless linking forming an ad hoc Network.

The aim of this replication is to offer a large data availability under such environments where the memory size of mobile host is limited and the communication links is frequently disconnected. By providing to each node a probability of a higher access.

This method is enriched by a data access protocol which aims at improving the access performances. The parameters considered for the evaluation of the data access service quality (QoS) are the reply time and the traffic generated.

4.1- Hop-Based Data Replication:

In this method we present two behaviours according to the status of data. Newly created, or already exiting in the network, in this case we replicate them dynamically in order to offer a better access service. These behaviours correspond respectively to two algorithms:

• Primary replication algorithm:

Given a shared data which present a certain interest for the majority of the network, it replicates and distributes uniformly the replicas.

Using a hop counter, this phase is initialised when a new data is created by one of the mobiles nodes or when receiving the first replica of data from a fixed server to which, the ad hoc network could be connected. This distribution attempts to put the data to the disposal of each node, all by making a compromise between the used storage space and the access time when the data are not locally found on the node.

• Dynamic replication algorithm:

The mobile nodes change the position and certain communication links are disconnected while others appear. This algorithm improves the reply times by approximating the data (in number of hops) to the nodes which often manipulate it.

The evaluation of distances between the replicas is carried out in term of the number of hops.

4.1.1- Primary replication

This algorithm generates primary replicas for every new data on the network. It considers the existing nodes and the present topology of the mobile network.

Principle:

We use a hop counter initialized to zero by the node which created or received, at first, the data. This replica of data is called original replica.

A diffusion to all immediate neighbours of a message **PrimaryCreat(NodeAdd, D_k, HopCpt)** of primary replicas creation of the data D_k is then initiated by the node creator.

Each time the message arrives in the level of a node, the hop counter is:

- incremented to one,
- or reinitialized to zero, if a replica is created on the visited node.

The node which receives this type of message, it proceeds as follows:

- If the counter is equal to three hops and if a replica of data exists neither in the cache of the node nor in one of the neighbours, for example, the node N_9 of the figure 1. After the creation of a replica, the node reinitializes the hop counter to zero and sends the message of creation to the neighbouring nodes.

- If the counter is equal to three hops, and at the same time, exist a replica of this data in one of the neighbouring nodes. In this case, the node "receiver " initializes the hop counter to one and sends the messages of creation to the following nodes (the nodes which don't send messages **PrimaryCreat**) and which don't hold a replica of data.
  For example, in the figure 1, the node N_{10} receives at first the message **PrimaryCreat** of N_6 and increment the hop counter to 3. But N_9 is a neighbouring node of N_{10} disposing a replica of data, N_{10} doesn't create replica of the same data in order to limit the redundancies of replicas on neighbours. The message **PrimaryCreat** is then transmitted towards N_{12} and N_{11}.
- If the hop counter is equal to two hops, and it doesn't exist the following links (links besides those by which the message has been received) thus, the node is chosen, to hold the replica as for the case for the node N_{18}.

A node which receives for the second time the message of primary creation of data must ignore it.

When placing primary replicas, if there is a lack of memory space, the local data which have a low access frequency are deleted in order to set free the space.

After their creation, these primary replicas are considered like all the other replicas existing on the network. And their dynamic management is presented in the following paragraph.

The aim of this repartition is to offer to each node a larger probability of availability of data acceded in its cache or in its nearest neighbouring nodes.

If the data acceded doesn't locally exist, a high probability of finding it on a node neighbour is warranted. This reduces the reply time of the access requests as well as the bandwidth and the energy consumption.

The load of work is repartitioned on the network with a uniform way, which balances the expenses in energy and reduces the traffic around the same mobile node.

The memory space is used in an optimal way by avoiding the unjustified redundancy of replicas. The redundancy is measured by the number of replicas of the same data on the neighbouring nodes.

The messages **PrimaryCreat** are for the functions:

- To inform the nodes of the creation of a new data on the network.
- To estimate the distance which separates a node of a replica of the data.
- To distribute the data replicas on the network.

Algorithm:

1/ the functions and the parameters used in the algorithm are mainly:

- **PrimaryCreat (NodeAdd, D_k, HopCpt) :** Message sent for a primary data replication where a replica is found at the node level **NodeAdd**. **HopCpt** is the counter of the nodes covered since the last reinitialization of this counter.
- **PrimaryReplica(D_k)** :This function creates locally a data replica, puts the counter **HopCpt** to zero and diffuse the message **PrimaryCreat(NodeAdd, Dk, HopCpt)** to the following neighbouring nodes if exist. In this case the **NodeAdd** is the identity of the local node.

  A border node is a node which hasn't a following link like N_8 (figure 1).

  A node preceding a node N_i is a node "sender" of a message **PrimaryCreat** to N_i.

A node following a node N_i is a node "non-sender" of a message **PrimaryCreat** to N_i.

2 / Algorithm of primary replication of data D_k

```
Begin
 If NJ creates a new data Dk then
        HopCpt = 0;  NodeAdd= NJ ;
   send PrimaryCreat(NodeAdd, Dk, HopCpt )
     to neighbours;
 endif;

If reception of PrimaryCreat(NodeAdd, Dk, HopNb) then
  If 1st reception then
   HopCpt = HopCpt +1;
    If (HopCpt =3) then
     If (a replica of Dk does not exist on the neighbours) then
                       execute PrimaryReplica(Dk )
        else
      If (a replica of Dk exists on one of the neighbouring Ni)
       then
          HopCpt = 1;  NodeAdd= Ni ;
          Send PrimaryCreat(NodeAdd, Dk, HopCpt) to
           neighbours
       endif
     endif
    else
    If (HopCpt =2) and (this node has only one neighbouring)
                    then
                       Execute PrimaryReplica(Dk ) ;
                    else
           Send PrimaryCreat(NodeAdd, Dk, HopCpt) to
            the following nodes
     endif
   endif
  else  Ignore the received message;
  endif
 endif
END.
```

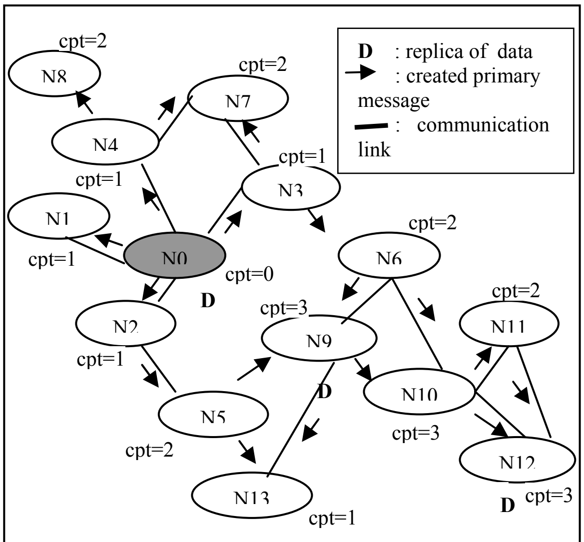

Figure1 : Example of primary replicas generation

4.1.2- Dynamic replication

Dynamic replication aims to redistribute the replicas for each data dynamically in order to maintain the data accessible for each node, to favour the best access time, and to reduce the consumption of bandwidth and energy. It also aims to limit the number of the replicas deployed for the best using of the capacity memory of mobiles hosts. Every time, two neighbouring nodes can hold replicas for the same data if it is match enquired by the two nodes.
This redistribution is carried out by realizing the new replicas of data on certain nodes and by suppressing others on other nodes according to the access frequencies of nodes.
A user can frequently use data where he doesn't hold a replica. Then the generated traffic becomes significant which increase the consumption of the bandwidth as well as an effort in energy from the part of the nodes included in the treatment and the advance of the requests.

In order to response to these preoccupations, we take into consideration the access rate to data. We define a variable $\mathbf{T_{ik}}$ which represents the access rate to data $\mathbf{D_k}$ by node $\mathbf{N_i}$, such as:

$$\mathbf{T_{ik} = NB/U,}$$

Where

U : Time unit

NB : The number of the access demands to $\mathbf{D_k}$ by $\mathbf{N_i}$.

In each time unit(U), the access rates are dynamically re-evaluated.
For a better management of cache, we define for each node $\mathbf{N_i}$ two types of rates:

- $\mathbf{TE_{ik}}$: The access rate of $\mathbf{N_i}$ to external and non local data $\mathbf{D_k}$.
- $\mathbf{TI_{ik}}$: The access rate of nodes to internal data $\mathbf{D_k}$ localized in $\mathbf{N_i}$.
- **S**: A threshold of an external access rate of node to data. Once this threshold is reached, the data are locally replicated in the node in order to avoid the performance degradation.

This replication can be found confronted the problem of the memory space lack. The solution is to set free the space by deleting the data replicas which have the lower external access rate and shouldn't be original data. This condition allows to ensure that the data should never be completely suppressed from the network, and at least, a replica will always subsist.

➢ <u>Replication to the demand</u>:

For a node which joins the network, we use the replication to the demand by respecting the hop conditions that we have used in the previous part. Two possible situations:

- For the data which will be created later, the node will be considered as the other nodes.
- For the data which have already achieved their primary replication, the node can profit by a replica to the demand. Therefore the processing can detect the need to realize a replication by respecting the condition of hops previously presented.

When a node comes to be connected, it wants to have access to non locally available data. It casts a message of the access demand to data. This demand will cover the nodes until their arrival on a node which holds a replica or the original data. The demand behaves besides the identity of the node "caller", the identifier of data, and the number of hops covered by this demand. The node holder of the replica consults this number. If the number of hop is greater or equal to three, then, instead of sending to it just a reply to its demand, it transmits to it a whole data replica.

The dynamic replication is a replication to the demand, it means, a creation of data replica can be done owing to an access demand of the user. The condition of replica generation will be explained in the following paragraph.

4.2- Importance-Based Data Replication :

In a scalable mobile ad hoc network, where the number of the nodes is more important, each node can be generated several original data. this generation of the primary replicas for each data by using the previous method, can increase the traffic and overload the mobile nodes, in particular, when the several nodes generate original data at the same time. In addition, with this unconditioned replicas generation, we can create a replicas of data which will not be very demanded by the other nodes. By what, the generation of the primary replicas for a data which does not represent an interest for the other nodes, increases the traffic and consumes the memory size of the mobile nodes and does not provide any profit to the mobiles users.
This method tries to ameliorate the previous. It proposes an improvement by using the following rule:

" *a node in an ad hoc mobile network executes the primary replica process for a new data which has created or has received from a fixed network, if it knows* previously *that this new data will be used by the mobiles nodes of the network. If not, it sends only a control message informing the other nodes of the existence of a new shared data* ".

In order to motive this idea, we give the following example: In a battle field, if a warrior detects a new reality(data), he should not wait until it is very demanded by the other warriors so that he replicates it. But, since he knows previously that this information which has collected, it will be interested by more than one warrior(mobile host), he replicates it using the primary replica process(previously described). Later, any soldier can reach this information and find it quickly.
This same scenario can be found in a collective work, where a significant researchers number forming an ad hoc network, cooperate with them to realize the same project. By doing his work, a researcher can arrive to found a new result, which the others need it to continue their work. Although this new data(result) is not required yet, but it must be previously replicated by using the primary replica process, because it will be the object of interest for the other researchers.

With this amelioration the primary replicas process is achieved as following:

```
If Ni creates or receives a new data Dk from a fixed
   server
   Then
        HopCpt = 0;  NodeAdd= NJ ;
      if Dk will be very requested
         then
           send PrimaryCreat(NodeAdd, Dk, HopCpt )
             to neighbours;
         else
         send Crtl_Msg(NodeAdd, Idf_Dk, HopCpt );
      Endif
Endif
```

Such as: **Crtl_Msg** is a control message and **Idf_D**$_k$ is an identifier of the new data **D**$_k$

4.3 Access to data

- **Data paths creation:**

The access demand to data by a node is processed as follows:

- The data are firstly researched locally at the node,
- If they exist at the node level, the access request is immediately satisfied,
- Otherwise, the request is cast to neighbours.
- A node which receives this request and which can not satisfy it, it will cast it to its neighbouring nodes.
- A node which receives this request and holds a data replica is going to reply by **Rep(**N_j**,** N_i**,** D_k**, HopCpt)** where N_j is the identity of the node holding data replica D_k inquired by N_i and **HopCpt** is the number of hops until to reach D_k. This broadcasts can charge or saturates the network and even take a higher execution time notably in a scalable ad hoc mobile network where the number of node is important. In order to avoid these drawbacks, and to reduce the energy and the bandwidth consumption, a node which doesn't hold a replica of data saves the shortest access path which detects, it means, the identity of the nearest node in number of hops which holds a data replica.

In the node N_j, an access path is structured as follows: (**IdNode**, **IdData**, **NbHop)** where, **IdNode** is the identity of the node which holds a replica of the data identified by **IdData** and **NbHop** is the number of hops which separate N_j of the identity node **IdNode.**

Periodically, the messages of the discovering neighbours are transmitted by the nodes. These messages have the mission of determining the immediate neighbouring nodes. They are also put into contribution for the building of data paths. Each time, a node discovers a new neighbour, it sends to it the list of the local data replicas. The node 'receiver' of this list updates its table of data access paths. Likewise, when a node updates its data table by generating a local replica, it broadcasts this update to its direct neighbours. These latter will proceed to the update of the access paths table of data.

The updating algorithm of the access paths is the following:

```
Begin
If (reception of data list of a node)   Then
    For each data non local save the path.
endif
If Ni don't possess a replica of Dk
 then
  if Ni receives the message PrimaryCreat (NodeAdd, Dk,
     HopCpt) of Nj then
        if HopCpt = 1  then
                 Nj possesses a replica of Dk
                 Save the access path: (Nj, Dk, 1).
        endif
        if HopCpt = 2 then
           Save the access path: (Nj, Dk, 2) if it doesn't exist a
            shorter path.
        endif
  endif
  If Ni receives a response Rep(Nj, Nk, Dk, HopCpt)
          to an access demand to Dk carried out by Nk
   then  /*HopCpt : number of hops between Ni and Nj, Nj
               the node holding the replica*/
     HopCpt= HopCpt+1;
     if Ni <> Nj then
          If  it doesn't exist a shorter path then
               Save the access path: (Nj, Dk, HopCpt)
               Send Rep(Nj, Nk, Dk, HopCpt)
          Else /*it exist a shorter path (Np, Dk, HopCpt1) */
               Send Rep(Np, Nk, Dk, HopCpt1) to Nk
          endif
     else
          If HopCpt =3 then
              Create a replica of Dk;
              Update the data paths tables "PathDatas";
              Send Rep(Ni, Nk, Dk, HopCpt=0) to Nk;
           endif
     endif
  endif
endif
fin.
```

The mobility of mobiles hosts makes that the data access paths can become invalid, and so they need a permanent update. We associate a time TTL (Time To Live) to each path. When this TTL is expired, the path is considered invalid and then it is deleted.

During the time of saving the path, if there is a lack of memory space, the oldest path is suppressed since will rapidly become invalid.

Example:

We consider the following example(figure 2) where N1 is the node 'initiator' of the primary data replication. It holds the original replica **D**. The nodes N11 and N12 are connected to the network after the primary phase of data creation. We suppose that N12 asked for the access to these data.

We execute this method in order to notice its effects on the dynamic creation of replicas and on updates of the access paths to data.

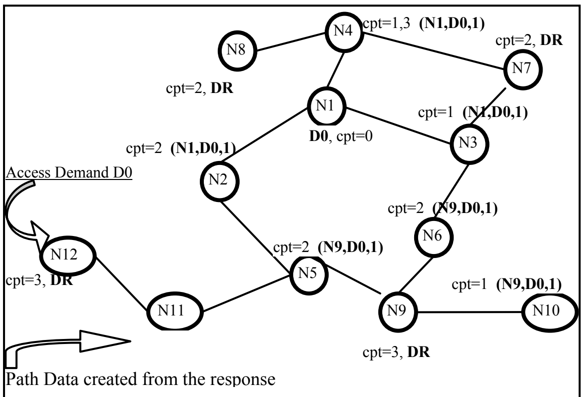

Figure2 : Application of the replication algorithm.

Algorithm:

```
If the node Nj calls for a data access then
  If Nj possesses a replica (original or replica) of data
      then
          The access is locally carried out
    Else
    If Nj possesses a valid path access towards
          this data then
              The demand is transmitted to the
          node indicated by the path.
      Else
        The demand is sent to the neighbouring
          nodes.
      Endif
  Endif
Endif
```

The receiver of an access request to a data behaves as a caller. The same previous processing is carried out to its level, except when the target data are locally existing, a reply is transmitted to the to the requester node.

5- Performances evaluation:

This section is reserved for the results of the performance evaluation of the proposed method. The evaluation has been carried out by using the simulator Glomosim [17].

The table 1, below, shows the parameters of our environment. The nodes move according to the model 'Random Way Point' which is a model widely used and which seems the closer real movement of the mobile nodes. In this model, the speed of displacement of the node is varied between a minimal value précised by the variable MOBILITY-WP-MIN-SPEED and a maximal value given by the variable MOBILITY-MAX-SPEED. The default dimension of the network is 1000x1000m^2. And, in order to test the scalability, the number of nodes can reach 250 on a site of 5000x5000m^2.

The nodes are randomly chosen in order to create new data and send access demands. The processes of creation of primary replica and the sending of the access demands follow the model of POISSON with an average interval of 60 seconds.
This interval is varied between 1 second and 60 seconds for testing the network load.

Parameter	Default value	Variation interval
Number of nodes	50	50 - 250
Number of data	50	
Size of data (KB)	1	1 – 4
Size of cache (KB)	20	5 – 50
Max Speed(M/S)	1	1 - 10
Time of break (S)	10	
Threshold of access frequency (HZ)	0.5	0.01 – 1.5
Threshold of the size of data(KB)	2	
Bandwidth(Mbps)	2	
Interval of data creation (S)	60	1 - 60
Interval of the access demand (S)	60	1 - 60

Table1: Configuration parameter

The access frequency is calculated by using the relation of the moving averages. [12], [13], [14], [15], [16] which is one of the formulae of well known technical analysis. The moving average is one of the oldest and widely used indicators.
It allows to calculate an average value on a given period. The access frequency is then calculated by the formula:

$$MAf_{ij} = \beta\, MAf_{ij}^{*} + (1-\beta)\, f_{ij}$$

Such as:

MA*f*ij: Moving average of the access frequency to the data j by the node i for the new period.
MA*f*ij*: Moving average of the access frequency to the data j by the node i for the old period.
***f*ij** : The access frequency of the node i to data j for the new period.

β: is a smoothing constant, in our simulation **β = 0.5,** with this value, we give more consideration to the new frequency value than to the historical background of the access of each node.

The **MA*f*ij** is calculated to each time unit(U). If the value of the external **MA*f*ij** exceeds a certain threshold S, the data concerned must be replicated.

The metrics of our simulation are defined as follows:

- *The rate of availability*:
  This metric allows to know the rate of data accessibility during the simulation time. Formally, this rate is defined by the following formula: TD = NDR/TDE
  Such as NDR and TDE are respectively, the number of successful demands and the total sent demands. The aim of all the protocols of replication is to increase the most possible TD.

- *The traffic* :

  The traffic is the number of messages transmitted by all the nodes during the simulation duration. Formally, the traffic is defined as follows:

$$T = \sum_{i=1}^{n} NmesTri$$

  Such as n is the number of nodes and NmesTri is the number of messages transmitted by the nodes i during the simulation.

To evaluate the proposed methods, we name the first method "Hop-Based replication Data" M1, and the second " Importance-Based replication Data " M2.

5.1 Accessibility, traffic and the replication threshold

A mobile node replicates the data if its access frequency to this latter exceeds a certain threshold. The access frequency is calculated by using the function of the moving average described previously. The aim of these measurements is to evaluate the influence of the value chosen as a threshold on the accessibility rate (figure 5).

The result shows that when the threshold increases, the data accessibility rate decreases then it stays stable after a certain value. When the threshold is low, all the access frequencies will exceed this threshold and the data concerned are locally replicated, which explain the high level of accessibility. When the threshold increases, the accessibility level decreases, because the selected threshold always stays greater than the most of access frequency of data, these latter can not be locally replicated which obliges the mobile node to transmit the access requests far which can not be satisfied.

This accessibility risks to be decreased when the load increases or the storage capacity decreases, what we are going to see later.

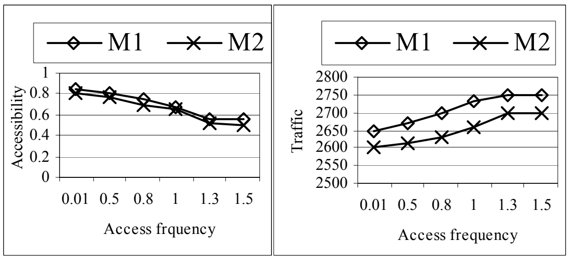

Figure 5 : Effect of the access frequency threshold on the accessibility and traffic

In this figure we observe that the traffic increases in a non-considerable way with the growth of the threshold. When the value of the threshold is small, almost the data required are replicated which reduces the number of the messages transmitted and reduces the traffic.

When the threshold increases, most of the data are not replicated and almost all the access demands are transmitted to other nodes, consequently, the result is a more important traffic. But the difference is not important since the replication traffic, produced in the first case, is compensated by the access traffic in the second case. The generated traffic depends on the importance of the numbers of distant data access . The traffic induced by the replication depends on the value of the fixed threshold.

5.2 Accessibility and storage capacity

In figure 6, we examine the effect of the memory size and the node movement speed. this figure shows that the rate

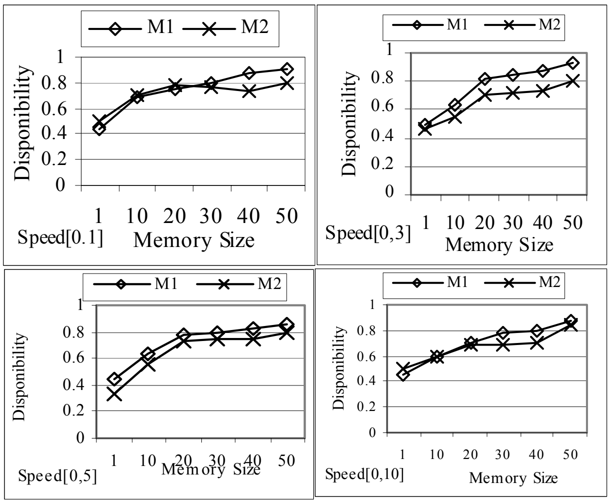

Figure 6: Accessibility and storage capacity.

of accessibility for the tow methods increases when the memory size increase, this is with the possibility of locally retorting the data requested.

In the case of our methods, we observe that if the memory of storage is broad the first method M1 gives the best rate of accessibility. This is due to the procedure of creation of replicas primary which makes it possible each new data to be retorted so that it is close to the user. When the storage capacity is small, the method m2 gives a more or less high level of accessibility compared to the premiere. This is due to the procedure of creation of replicas primary which is carried out, in this case only, for the much in demand data, by what the creation of the primary replicas for the data which are not very required requires a larger storage capacity and requires the suppression of the data locally stored which can be reached data.

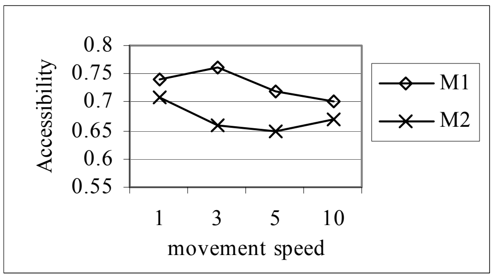

Figure 7 : Accessibility and speed of displacement.

To clearly see the influence of the variation rate of travel of the nodes on the accessibility of data, we gathered the four states of the figure6, in only one state presented by the figure7, where each point in this figure is the average of all the points represented by each state of the figure7.

In this figure we notice that both are not very influenced by the increase rate of travel of the nodes and the method M1 always gives the highest rate of accessibility in the event of memory size raised even when the rate of travel increases by what M1 does not limit the creation of the primary replicas for the data shared by their importance.

5.3 Traffic and storage capacity

The traffic generated by these methods is produced by the primary replicas process and by the access requests when the data requested is not locally available. Figure 8 shows that the traffic for the both methods remains relatively low.

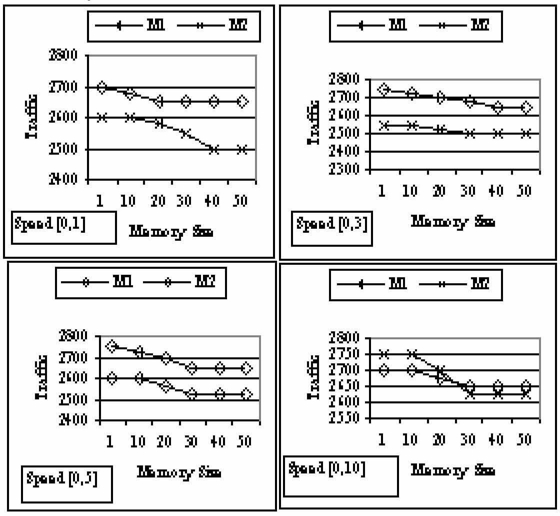

Figure 8 : Traffic and storage capacity

The increase of movement speed can support the network instability which reduces the accessibility rate and increases the traffic. In figure 8 we observe that the traffic generated by the methods M1et m2 remains always relatively very low and the two methods are not influenced by the increase of movement speed, the method m2 gives the smallest traffic. The method M1 executes primary replicas process for all new data created which increase the number of produced messages.

The method m2 creates the primary replicas only for the new data which will be very required by the other mobiles hosts which decreases the number of produced messages and decreases the traffic.

5.4 Accessibility and access frequency

We aim in this experiment to observe the influence of the increased number of access requests on the rate of data availability (figure 9). We see that the rates of data accessibility stays above the acceptable limit and the variation interval of these rates is not very influenced by the load.

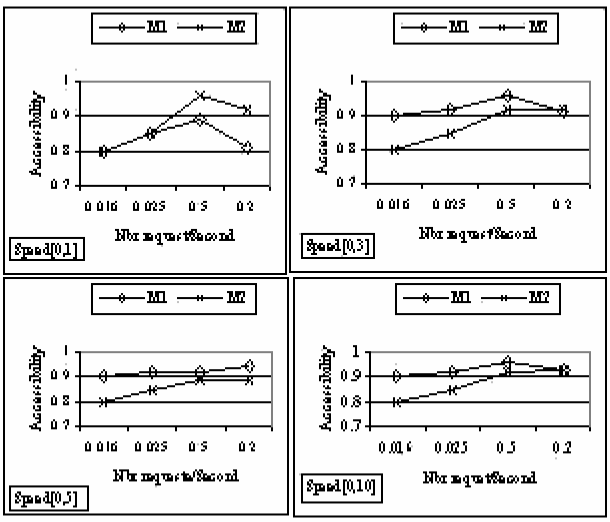

Figure 9 : Accessibility and access frequency

On the other hand, the speed variations influence the success of access too much. We observe that the data accessibility increases when the access frequency increases, because the data are locally replicated if their access frequency exceeds the fixed access frequency threshold.

The access demands are first sent to distance if the data concerned are not locally available. These requests can't all be satisfied. But after the creation of replicas for these data, since their access frequency exceed the threshold, all following access will be more quickly satisfied.

5.5 Traffic and access frequency

In figure 10, we notice that the traffic increases with the increase of the load expressed in frequency access. When the number of demands increases, the traffic increases too, notably in the case where the data required are not locally found and there is no path which leads towards these data.

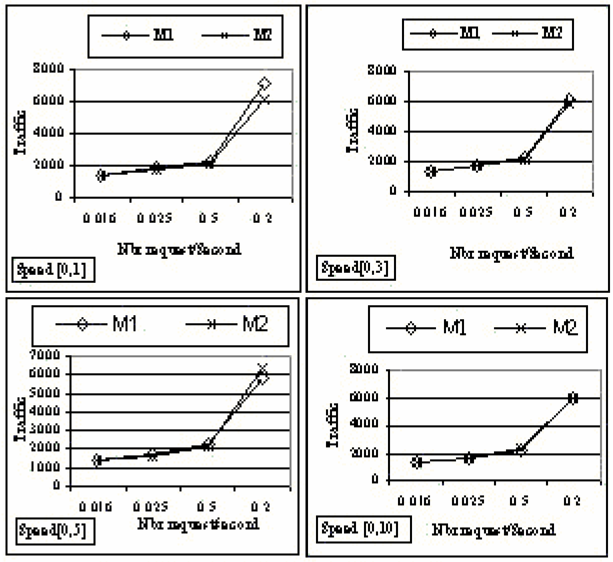

Figure 10 : Traffic and access frequency

We also observe that the traffic is more influenced by the access frequency than by the node movement speed, and the method m1 gives a more traffic than m2 because m1 executes the primary replicas process for each new data created in the network.

4.6 The reach of communication

The accessibility rate increases with the increase of the radio communication range. Which is explained by a higher number of connections and consequently a higher probability of finding data in a neighboring nodes when this data does not exist locally.

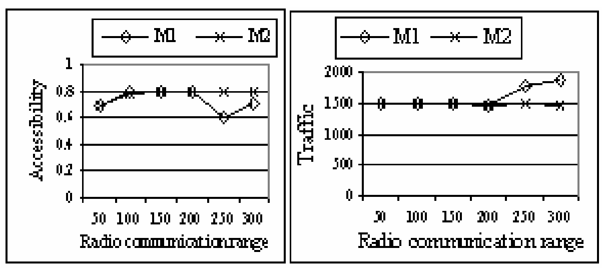

Figure 11 : Effect of radio communication range

In figure11, we observe that the method m2 gives a stable accessibility rate, because in this method the primary replicas occur only for the new data which will be very required, this improves the memory size and reduces the traffic. For the method M1, the accessibility decreases a little when the range of communication becomes larger, because in this case, the number of nodes which can be chosen by the primary replicas process decrease. This can decrease the accessibility and increase the traffic.

5.8 Scalability

In general, we observe that the accessibility in figure 12 increases with the increase of the nodes number because the network becomes denser which favors the connections.

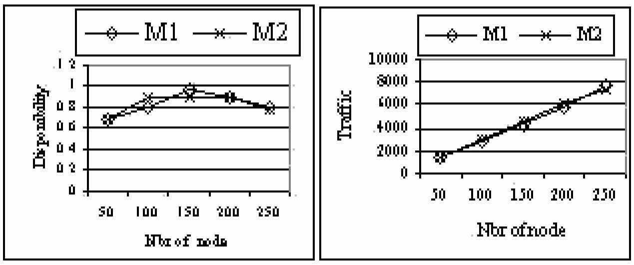

Figure 11 : Scalability

The method M1 gives a high level of accessibility which decreases a little when the number of nodes becomes larger, because in this method, any node which creates a new data, must execute the primary replicas process, which makes it possible to keep closer the data from the mobiles hosts. But when the number of node increases the number of new data increases too. This will consume a memory capacity of some nodes and will increase the traffic(nodes chosen by the primary replicas process). For this reason the accessibility decreases a little when the number of node increases. The second method m2 gives an accessibility level close to that given by m1 and the traffic provided by m2 remains small compared to the first method m1

6- Conclusion

In this paper, we have proposed two data replication methods. The simulation experiences of these methods, shows that the data accessibility rates are interesting in most of the cases and the traffic induced stays acceptable. The method M1 gives a high accessibility level, but when the number of new data creation increases, the accessibility rate decreases a little and the traffic increases, this is because M1 executes the primary replicas process for each new data created. This technique allows to keep any information closer to the mobiles hosts. But it consumes the memory size, therefore this method is very effective when the mobile devises have memory capacity relatively high or the number of new data creation is low.

The method M2 gives a high level of accessibility approximately like the method M1. The traffic produced by this method is lower than that caused by M1. The primary replicas process in this method, has executed only for important data. the memory capacity consumed by this method is lower than that consumed by the method M1. However the method M2 is very effective when the memory capacity of the mobile devises is not significant.

As part of our future work, we are planning to address replica creation taking into account the size and the replication time of data replicated

References

[1] T. Hara. "Effective replica allocation in ad hoc networks for improving data Accessibility". In Proc. of IEEE Infocom 2001, Anchorage, Alaska, U.S.A., volume 3, pages 1568-1576, April 2001.

[2] Takahiro Hara, Yin-Huei Loh, Shojiro Nishio. " Data Replication Methods Based on the Stability of Radio Links in Ad Hoc Networks", Proceedings of the 14th International Workshop on Database and Expert Systems Applications (DEXA'03) 1529-4188/03 2003 IEEE.

[3] Liangzhong Yin and Guohong Cao : "Supporting Cooperative Caching in Ad Hoc Networks", IEEE INFOCOM, March 2004.

[4] T. Hara "Replica Allocation in Ad Hoc Networks with Periodic Data Update".Proceedings of the Third International Conference on Mobile Data Management (MDM.02) 0-7695-1500-2/02 2002 IEEE .

[5] Takahiro Hara and Sanjay Kumar Madria. "Dynamic Data Replication Using Aperiodic Updates in Mobile Adhoc Networks", 9th international conference. (Eds.): DASFAA 2004, LNCS 2973, pp. 869–881, 2004. © Springer-Verlag Berlin Heidelberg 2004.

[6] Takahiro Hara. "Replica allocation methods in ad hoc networks with data update", ACM-Kluwer Journal on Mobile Networks and Applications, vol 8(4), pp 343-354, 2003.

[7] K. Wang and B. Li. "Efficient and guaranteed service coverage in partitionable mobile ad-hoc networks". In Proceedings of IEEE INFOCOM 2002.

[8] Malika Boulkenafed, Valerie Issarny, "AdHocFS: Sharing Files in WLANs",INRIA-Rocquencourt Domaine de Voluceau, Rocquencourt, BP 105, 78153 Le Chesnay C´edex, France 2002

[9] D. Barbara and T. Imielinski, "Sleepers and workholics: Caching strategies in mobile environments", Proc. ACM SIGMOD'94, pp.1-12, 1994.

[10] Y. Huang, P. Sistla, and O. Wolfson. "Data replication for mobile computer",Proc. ACM SIGMOD'94, pp.13-24, 1994.

[11] E. Pitoura and B. Bhargava, "Maintaining consistency of data in mobile distributed environments", Proc. IEEE ICDCS'95, pp.404-413, 1995.

[12] I. W. Burr, “Statistical Quality Control Methods”, Volume 16, New York: Marcel Dekker, Inc (1976).

[13] D. C. Montgomery, “Introduction to Statistical Quality Control”, Third Edition, New York: John Wiley & Sons, Inc. 1996.

[14] S. V.Crowder, “Average Run Lengths of Exponentially Weighted Moving Average Charts”, 1987b, Journal of Quality Technology, 19, 161 - 164.

[15] H. Kume, “Statistical Methods for Quality Improvement”, Tokyo: AOTS Chosakai, Ltd. 1985.

[16] Hunter, J. S. “The Exponentially Weighted Moving Average”, 1986, Journal of Quality Technology, 18, 203 -210.

[17] R. Bagrodia, X. Zeng, and M. Gerla, “GloMoSim - A Library for Parallel Simulation of Large-scale Wireless Networks”, computer Science Departement, University of California at Los Angles, 1999.

[18] T.Hara, K.Harumoto, M.Tsukamoto, S.Nishio, “Dynamic replica allocation using database migration in broadband networks”,Proc. IEEE ICDCS 2000, pp.376-384,2000.

[19] A.W. Fu and D.W.L Cheung, “A transaction replication scheme for a replicated database with node autonomy,” Proc. 20th VLDB Conference, pp.214-225, 1994.

[20] N. Krishnakumar and A.J. Bernstein. “High performance escrow algorithms for replicated databases”, Proc. 18th VLDB Conference, pp.175-186, 1992.

[21] O.Wolfson and S. Jajodia, “Distributed algorithms for dynamic replication of data”, Proc. ACM PODS’92, pp.149-163, 1992.

[22] M.J. Carey and M. Livny, “Distributed concurrency control performance: Astudy of algorithm, distribution, and replication”, Proc.14th VLDB Conference, pp.13-5,1988.