QUORUMS BASED MOBILES LOCATION SCHEME

Nadia Chetta
LSI- Department of Computer Science, Faculty of Electronic Genius & Computer Science, USTHB, El Alia BP n°32, Bab Ezzouar, Algiers, Algeria

Nadjib Badache
LSI- Department of Computer Science, Faculty of Electronic Genius & Computer Science, USTHB, El Alia BP n°32, Bab Ezzouar, Algiers, Algeria

Keywords: Mobile networks, Location, Quorums systems, Communication cost.

Abstract: Mobile location management is one of the most important problems in the mobile networks systems. Indeed, reducing the search cost increases the update cost and vice versa. That's why, a trade off between the search cost and update cost must be defined. This paper proposes a location management scheme based on quorums. The communication cost is reduced with our method because the information on a mobile location is saved in an efficient manner in a subset of location registers that change with the mobile moves. The proposed algorithm is evaluated in term of total cost of search and update, and is compared to Ihn-Han Bae algorithm.

1 INTRODUCTION

In the actual distributed systems and applications, the notion of mobility appears in different forms. More and more people use mobile computers or wireless phones. In a mobile environment, the cost of the communication cost is increased by the location cost. When a server tries to send a message to a user that is moving, it must first determine his current position, i.e. locates it and then transmits it the message.

To be able to locate mobiles efficiently, information on their current positions must be saved somewhere in the network: in a location database. When a server has to locate a mobile, it queries this location database. Therefore, every mobile that moves, must signal its new position in order to update its address in the location database.

There are two operations in location management: Search and update. Search is the operation that permits to locate a mobile node when there is a need to contact it. The update is the operation that permits to inform the network (the location database) about the position of a mobile when this one moves towards a new position in the network.

In this paper, we propose a scheme for a location database, based on quorums.

The present paper is organized as follows: In section 2, we describe the system model used. Section 3 is dedicated to previous work related to the location problem. Section 4 presents the principle of our solution and its evaluation. The last section is the conclusion.

2 SYSTEM MODEL

We consider a system composed of static sites linked by a classic wireline network (static) and mobile sites. Some static sites are provided with wireless communication interface and play the role of base stations. Every base station covers a geographical area called a Cell. A mobile site can only communicate with the base station that covers the cell in which it is. This base station represents its current address.

The cells of the network are grouped into Registration Areas (RA). Every RA has a Location Register (LR) that maintains the addresses of all mobile sites that are in this RA.

Chetta N. and Badache N. (2005).
QUORUMS BASED MOBILES LOCATION SCHEME.
In *Proceedings of the Second International Conference on e-Business and Telecommunication Networks*, pages 92-97
DOI: 10.5220/0001410900920097

3 PREVIOUS WORK

The simplest solution for location management is the use of a centralized server. This solution, even if simple to implement, resents a major inconvenience: If this server crashes, the location of all mobiles becomes unreachable. Besides, this solution doesn't exploit the geographical distribution of the mobiles in the system and the locality of calls and moves to reduce the cost of search and update. That's why it is necessary to use a distributed location database.

The Standards IS-41 (EIA/TIA, 1991) and GSM MAP (Mouly, 1992) use the two-tier scheme. IS-41 associates a HLR (Home Location Register) to every mobile. When a mobile moves from a RA to another, the mobile saves its address in the VLR (Visitor Location Register) of the LR towards which it moved. This VLR informs the HLR associated to this mobile to update its address.

The search and the update in this scheme imply an important communication cost. Therefore, several improvements have been proposed to reduce the search and update costs (Pitoura, 2000) as the use of cache, Forwarding Pointers (Krishna , 1996), Local anchoring...

In (Rajagopalan, 1995), authors use the notion of working sets. A mobile has a tendency to work, in particular, with a certain number of nodes and that represents its working set. Thus, when a mobile moves, its working set is informed. This means that, its address is updated in each node of its working set.

In (Awerbuch, 1995), a Regional Directory (RD) is created to maintain the location information of mobiles. The RD of the i^{th} level allows a node to search for a mobile in a distance of 2^i of it. To every level i, reading and writing sets are associated with the nodes u and v such as $read_i(u) \cap write_i(v) \neq \emptyset$ $\forall$ u,v where the distance between them doesn't exceed 2^i. The reading set of a node is the set of nodes that has tendency to search for this node. The writing set of a node is the set of nodes where the address of this node is saved.

Updating the address of a mobile every time it moves from a cell to another can be very expensive. Several alternatives have been proposed to reduce this cost (Pitoura, 2000). They are based on the time, the distance, and the number of movements...

In (Prakash, 1996) and (Prakash, 2001), Prakash and Al. proposed a dynamic location strategy that permits a load balancing in the system. In this strategy, the address of a mobile is saved in a subset of LRs called quorum. This quorum is determined by a dynamic hashing function that defines the writing set of a mobile when it moves and its reading set when it is searched.

Several solutions were inspired from this model and proposed some improvements.

In (Peleg, 1996) and (Kumar, 2000), authors propose solutions assuring a trade off between the load, the fault tolerance and the quorums size. In (Ihn-Han, 1999) and (Ihn-Han, 2000), a solution is proposed to reduce the communication cost with regard to these last solutions.

4 NEW SOLUTION

4.1 Basic idea

This solution proposes a mobile location database architecture based on quorums. It is characterized by:

- The broadcast of messages to update a mobile address to a set of LRs.

- The broadcast of messages to search for a mobile to a set of LRs.

The LRs of the network are grouped into quorums. So, in addition to the information about the mobiles that are in the base stations, a LR maintains information related to all mobiles that are in the other LRs belonging to its quorum.

Let LR_A be the location register of the Registration Area of A (RA_A) that is in the quorum Q_A.

- LR_A contains the information of all mobiles in RA_A in terms of base stations. Let *x* be a mobile that is in RA_A, information saved in LR_A concerning the address of *x* is the base station number in which it is.

- LR_A contains the information of the mobiles of the other LRs that are in the same quorum that RA_A in terms of LRs. Let *y* be a mobile that is in a RA that belongs to Q_A, information saved in LR_A concerning the address of *y* is the number of the LR in which it is.

4.2 Updating a mobile address

The update of a mobile address is made when this one moves from a quorum to another.

When a mobile *x* moves from one base station A to another base station B, three cases are possible:

1. A and B are in the same LR *y*: In this case, the address of *x* is updated at *y* level.

2. A and B are not in the same LR, but remain in the same quorum: if *x* moves from the LR of A (LR_A) to the LR of B (LR_B), then LR_B and LR_A update the address of *x*. LR_A deletes the address of *x* in its database associated to LR_A and adds it to the one associated to LR_B. The inverse is done at the LR_B level. Then LR_B broadcasts a message to all

LRs that are in the same quorum with it (except LR_A) to update the address of *x*. Each of these LRs is going to delete the address of *x* in its sub database associated to LR_A and to add it to the sub database associated to LR_B.

3. A and B are in different quorums: if LR_A is in the quorum Q_A and LR_B is in the quorum Q_B, then:

3.1. The LRs that belong to Q_A-Q_B delete the address of *x* from their databases: LR_A deletes the address of *x* from its database and broadcasts a delete message to all LRs of Q_A-Q_B so that they delete the address of *x* from their databases.

3.2. The LRs that belong to Q_B-Q_A add the address of *x* in their databases: LR_B adds the address of *x* to its database (here the address of *x* is B) and broadcasts a message of addition to the LRs of Q_B-Q_A so that they add the address of *x* in their databases (here the address of *x* is LR_B).

3.3. The LRs that belong to $Q_A \cap Q_B$ update the address of *x*: LR_B broadcasts an update message to the LRs of $Q_A \cap Q_B$ so that they delete the address of *x* from the database associated to LR_A and add it to the database associated to LR_B. Here also, the address of *x* is the LR to which it belongs (LR_B).

Update Procedure:

Let Mh_id be the mobile identity of a mobile *x*, old_SB the departure base station of *x*, new_SB the arrival one, old_LR the departure LR of *x*, new_LR the arrival one, old_quorum the departure quorum of *x* and New_quorum the arrival one.

```
If old_LR=New_LR
Then update the address of x in old_LR
Else
   if old_quorum = New_quorum
   then
      - Update the address of x in old_LR and New_LR
      - Broadcast an update message in
      Old_quorum-{old_LR,New_LR}.
   else
      - Delete the address of x in old_LR;
      - Broadcast a delete message to
      Old_quorum-New_quorum-{old_LR}
      - Add the address of x in New_LR;
      - Broadcast an add message in
      New_quorum-old_quorum-{New_LR}
      - Broadcast an update message in
      New_quorum ∩old_quorum.
```

4.3 Locating a mobile

When a mobile *y* that is in a base station A wants to communicate with a mobile *x* that is in the base station B, considering that A belongs to the Location Register LR_A and B belongs to LR_B, A asks LR_A for the address of *x*. Three cases are possible:

1. If *x* and *y* are in LR_A (i.e. $LR_A = LR_B$), then LR_A sends the address of *x* (here B) to A;

2. If LR_A and LR_B are in the same quorum, then LR_A has the address of *x* in terms of LR: LR_A sends this address (LR_B) to A. The message that *x* must receive, is sent to LR_B that will send it to B.

3. If LR_A and LR_B are not in the same quorum, three (03) cases are possible:

3.1. LR_A checks its cache to see if it contains the address of *x*. In a LR, the cache is a structure that saves the address -in termS of LRs - of the last address of every mobile that have been searched by the LR.

If *x* possesses an entry in the cache of LR_A (Let LRx be this entry), LR_A sends a search message to LRx. If *x* is still in LRx, LRx sends the address of *x* to LR_A which will send it to A; otherwise, LRx sends a failure message.

3.2. If LR_A doesn't find the address of *x* in its cache, LR_A checks its profile. The profile is a structure that replicates usual mobility models of a mobile. It is managed on the basis of history, where each mobile has P LRs (and therefore P entries in the profile) ordered by their popularity. The popularity of a LR is the probability that the mobile is located in the LR for each period of time. The profile saves for a mobile, the LRs in which it often resides.

LR_A broadcasts the search request to the LRs of the profile of *x*. When a LR receives the search message, if *x* is in this LR, it sends the address of *x* to LR_A that sends it to A; otherwise it sends a failure message.

3.3. If LR_A doesn't find the address of *x* in the cache and all LRs associated to *x* in the profile answer negatively to its search message, then LR_A doesn't possess the information of the address of *x*. LR_A broadcasts a search message to all LRs that belong to the intersection of the quorum of LR_A with the other quorums of the network. When LR_A receives the address of *x*, it sends it to A.

Search Procedure:

Let SB_id1 be the base station that searches for the mobile *x* whose identity is MH_id and SB_id2 the base station in which *x* is.

Let LR_id1 be the LR that contains SB_id1 and LR_id2 the LR that contains SB_id2.

Let QR_id1 be the quorum that contains LR_id1 and QR_id2 the quorum that contains LR_id2.

Let $LR_{\cap}$ be the set of LRs that belong to the intersection of QR_id1 with other quorums of the network.

```
If LR_id1=LR_id2
Then
  LR_id1 sends MH_id address (SB_id2) to SB_id1
Else
  If QR_id1=QR_id2
  Then
    LR_id1 sends MH_id address (LR_id2) to SB_id1.
  Else
    LR_id1 searches MH_id address in its cache.
    If it exists, let LR_x be this address,
    Then
      LR_id1 sends a search message to LR_x.
      If LR_x=LR_id2
      Then
        LR_id2 sends MH_id address to LR_id1 that
        will send it to SB-id1
      Else LR_x sends a failure message.
      Goto L1
    Else
      L1: LR_id1 broadcasts the search message to the
      LR of the MH_id profile.
      If LR_id2 is in MH_id profile,
      Then
        LR_id2 sends the address of x to LR_id1 that
        will send it to SB_id1 and the other LR send
        failure messages.
      Else
        LR_id1 sends a search message to LR∩ ;
        Let LRj є LR∩ / LRj є QR_id2 : LRj sends
        MH_id address (LR_id2) to LR_id1 ; LR_id1
        sends this address to SB_id1
```

4.4 Algorithm complexity

During a search or an update, the number of exchanged messages can vary according to the following cases.

- When a mobile moves from a base station to another, exchanged messages between the LRs are:

1. If the mobile stays in the same LR, no messages are sent.
2. If the mobile stays in the same quorum, the new LR sends $q-2$ messages to the LRs of the quorum so that they update the mobile address. *q* is the maximum size of the quorum.
3. If the mobile moves to a new quorum:

Let *i* be the number of LRs that belong to the intersection of the departure quorum and the arrival quorum, the number of exchanged messages is $2*(q-1)-i$. This number can up to $2*q-1$ if *i*=1.

- When there is a search:

1. If the calling mobile and the called mobile are in the same LR, the number of exchanged messages is four (04).
2. If the calling mobile and the called mobile are in the same quorum: the number of exchanged messages is six (06).
3. If the calling mobile and the called mobile are in two different quorums:

3.1. If the calling LR finds the searched mobile address (the address of its LR, let it be LR_MH) in its cache, it sends the search request to LR_MH. If the mobile is located, the number of exchanged messages then is six (06).

3.2. If the search in the cache fails and the called mobile is in the calling LR profile, the calling LR queries the LR of the profile of the called mobile. If P is the size of the profile, the number of exchanged messages is $2*(p+3)$.

3.3. If the mobile is still not located, the calling LR broadcasts a search request to all LRs belonging to the intersection of its quorum with the other quorums. The number of these LRs is $(q-1)$ LRs as maximum. Therefore the number of exchanged messages in this case is $2*\{P+q+2\}$.

4.5 Evaluation

To evaluate our work, we have compared it to the Ihn-Han Bae algorithm (Ihn-Han, 2000).

The first difference between the Ihn-Han Bae algorithm and our solution is that Ihn-Han Bae associates to every LR two quorums: a line quorum and a column quorum. When a mobile moves, these quorums are updated according to the state of the concerned mobile. If this mobile is hot, the two quorums (line and column) are updated. If it is cold, one of the two quorums is updated. In our solution, the notion of reading and writing quorums or line and column quorums doesn't exist. The LRs of the network are organized in quorums where each LR can belong to several quorums and is affected to only one quorum. When a mobile is searched, a LR queries the other LRs belonging to its quorum.

Besides, the update in the Ihn-Han Bae solution is made when a mobile moves from a LR to another. In our solution, the address of a mobile is updated when it moves from a quorum to another.

Finally, the quorums construction used is different. Ihn-Han Bae uses the triangular scheme (construction in line and in column) where quorums have the same size that is roughly $\sqrt{2}\sqrt{N}$. In our solution, the used construction is the tree based scheme where the size of quorums is roughly Log_2 N with *N* the number of LRs in the network.

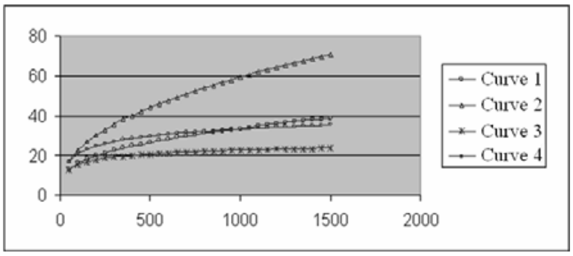

Figure 1: Communication cost according to the number of cells

In the following, we present a comparison between the performances of the Ihn-Han Bae algorithm and our solution in the conditions that were used by Ihn-Han Bae to compare his algorithm to the Prakash one. That is: C_{fixed}=1, $C_{wireless}$=3, C=7, P=3, P_{cache}=0.3, $P_{profile}$=0.2, P_{Hot}=0.5 with C_{fixed} is the wireline communication cost, $C_{wireless}$ is the cost of the wireless communication between a mobile and a base station, P_{cache} is the probability that the address of the called mobile is in the cache of the calling mobile, $P_{profile}$ is the probability that the address of the called mobile is in the profile, P_{Hot} the probability that a mobile is hot, C is the number of base stations (or of cells) in a LR and P the profile size.

To evaluate our solution, we compare it in terms of communication cost with Ihn-Han Bae solution in the case where the scheme used for the construction of quorums is the triangular scheme and in the case where it is the tree based scheme.

Figure 1 shows the difference of performances between the two solutions.

Curve 1: Shows the performance of our solution with $q = \sqrt{2}\sqrt{n/C}$

Curve 2: Shows the performance of the solution of Ihn-Han Bae with $q = \sqrt{2}\sqrt{n/C}$

Curve 3: Shows the performance of our solution with $q = Log_2(n/C)$

Curve 4: Shows the performance of the solution of Ihn-Han Bae with $q = Log_2(n/C)$

First, it is clear that the choice of the quorums construction is very important. In the two solutions, the gap of the communication cost is considerable between the tree based construction and the triangular construction.

In the case of the triangular construction or the tree based construction, our solution gives better results than the Ihn-Han Bae algorithm, even if the gap is more important in the triangular construction.

If Figure 1 shows globally that our solution is better than the Ihn-Han Bae one, Figure 2 and Figure 3 show the detail of this curve by comparing search costs and update costs independently.

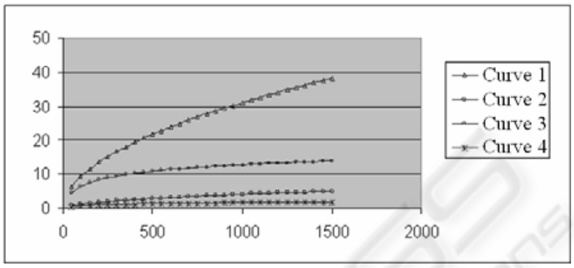

Figure 2: Update cost according to the number of cells

Figure 2 compares the update cost between the two solutions as:

Curve 1: Shows the performance of the solution of Ihn-Han Bae with $q = \sqrt{2}\sqrt{n/C}$

Curve 2: Shows the performance of our solution with $q = \sqrt{2}\sqrt{n/C}$

Curve 3: Shows the performance of the solution of Ihn-Han Bae with $q = Log_2(n/C)$

Curve 4: Shows the performance of our solution with $q = Log_2(n/C)$

Our solution reduces the update cost much more than the Ihn-Han Bae solution. In fact, the update cost of our solution with a triangular construction is weaker with regard to the update cost of the of Ihn-Han Bae solution with a tree based construction, when we know that the size of a quorum in the triangular construction is $q = \sqrt{2}\sqrt{n/C}$ and the one in the tree based construction is $q = Log_2(n/C)$. It is due to the fact that in the case of the Ihn-Han Bae solution, when a mobile is hot, four quorums are updated and two only if it is cold, whereas in our solution two quorums are updated. Besides, a mobile has more prone to pass from a LR to another, than to pass from a quorum to another. So, the number of updates in our solution is lower than the number of updates in the Ihn-Han Bae solution.

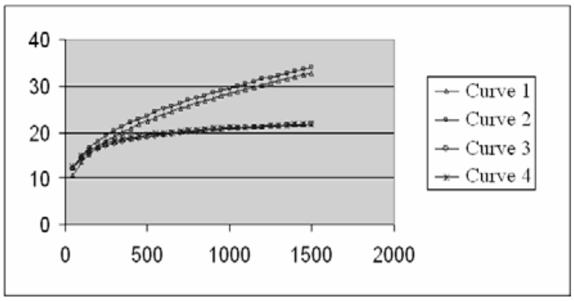

Figure 3: Search cost according to the number of cells

Figure 3 compares the two solutions in terms of search cost where:

Curve 1: Shows the performance of the solution of Ihn-Han Bae with $q = \sqrt{2}\sqrt{n/C}$

Curve 2: Shows the performance of our solution with $q = \sqrt{2}\sqrt{n/C}$

Curve 3: Shows the performance of the solution of Ihn-Han Bae with $q = Log_2(n/C)$

Curve 4: Shows the performance of our solution with $q = Log_2(n/C)$

In the case of triangular construction, the Ihn-Han Bae solution is appreciably better than ours but in the tree based construction the two curves are nearly identical with a light advantage for the Ihn-Han Bae curve.

5 CONCLUSION

In this paper, we proposed a location management scheme based on quorums. The information of a mobile location is mobile too. It moves with displacements of the mobile, the address of a mobile is updated when it goes from a quorum to another. The total communication cost induced by our solution is smaller than the one induced by the Ihn-Han Bae solution.

The communication cost depends on the chosen quorums system. Indeed, the tree based scheme reduces the communication cost with regard to the triangular scheme whereas the induced load by the tree based diagram is more important. Besides, to evaluate the performance of our solution, we supposed that the mobiles are uniformly distributed in the LRs. It would be interesting to simulate the behaviour of our algorithm for real cases that don't necessarily verify this hypothesis.

6 REFERENCES

Awerbuch, B. and Peleg, D., 1995. *Online Tracking of Mobile Users*. Journal of the ACM, 42(5).

EIA/TIA IS-41.3 (révision B), , July 1991. *Cellular Radio Telecommunications Intersystem Operations*.

Ihn-Han, B., 1999. *A quorum-Based Dynamic Location Management Method for Mobile computing.* Dept. Of Computer Engineering, Catholic University of Taegu-Hyosung. Kyungsan, Kyungpook 712-702, Korea.

Ihn-Han, B., Sun-Jin, O. and Jae-Gyeong, L., 2000. *A distributed dynamic Location Management Scheme for Mobile computing Systems.*

Krishna, P., Vaidra, N. H. and Pradhan, D. K., March 1996. *Static and Dynamic Location Management in Mobile Wireless Networks.* Journal of Computer Communications (spatial issue on mobile Computing), 19(4).

Kumar, A. and , Camp, T., 2000. *A new Matching Algorithm for Managing Location Information in Mobile Computing.*

Mouly, M., and Pautet, M. B., 1992 . *The GSM System for mobile Communications.* Book published by authors, 49, rue Louise Bruneau, F-91120 Palaiseau, France.

Peleg, D. and Wool, A., August 14, 1996. *Crumbling Walls: A Class of Practical and Efficient Quorum Systems.*

Pitoura, E., and Samaras, 2000. *Locating Objects in Mobile Computing.* IEEE Transactions on Knowledga and Data Engineering.

Prakash, R. and Singhal, M., 1996. *A dynamic Approch to Location Management in Mobile computing systems.* Dept. Of Computer and Information Science. The Ohio State University. Columbus, Ohio 43210.

Prakash, R., Singhal, M. and Haas, Z., 2001. *Load-Balanced Location Management for Cellular Mobile systems using Quorums and Dynamic Hashing.* Wireless Networks 0.

Rajagopalan, S. and Badrinath, R., November 1995. *An Adaptative Location Management Strategy for Mobile IP*. Proceeding of First ACM Mobicom 95.