

# An Efficient Positioning Solution to Evaluate Location-Dependent Queries in Wireless Environments

Marie Thilliez, Thierry Delot, Sylvain Lecomte

LAMIH Laboratory - CNRS UMR 8530  
University of Valenciennes  
Le Mont Houy  
59313 Valenciennes Cedex 9 France  
[\[firstname.lastname\]@univ-valenciennes.fr](mailto:{firstname.lastname}@univ-valenciennes.fr)

## Abstract.

The recent emergence of handheld devices and wireless networks has provoked an exponential increase in the number of mobile users. These users are potential consumers of new applications, such as the Location-Dependent Applications (LDA) examined in this article. As their name implies, these applications depend on location information, which is used to adapt and customize the application for each user. In this article, we focus on the problem of information localization, particularly the evaluation of Location-Dependent Queries (LDQ). Such queries allow, for example, a mobile user who witnesses an accident to locate and contact the closest ambulance service. To evaluate these queries, the client position must be retrieved. Often, positioning systems such as GPS are used for this purpose; however, not all mobile clients are equipped with such systems. To remedy this lack, we propose a positioning solution that can provide an approximate client position, sufficient for evaluating LDQs. This positioning system can be used at any time, in any environment, and does not require any particular infrastructure. It can also be adapted to the mobile user environment to minimize battery consumption in handheld devices. This paper presents both the positioning system, and its optimization with regard to minimizing response time and economizing mobile device resources.

Keywords: Positioning Solution, Location Dependent Query, Query Evaluation, Optimization

## 1. Introduction

The recent emergence of handheld devices and wireless networks [1] implies an exponential increase of mobile users, requiring services providers to propose new applications. This article focuses on one kind of new applications: Location-Dependent Applications (LDA) and their requirements due to user mobility. To be effective, LDA must be able to access the location of mobile device users—anywhere and anytime. This information is used to adapt and customize the LDA for each user in terms of mobility-related requests, such as access to navigational aid, emergency health and safety relays, and entertainment information.

In this context, we focus on the problem of localization information, particularly the evaluation of location-dependent queries (LDQ) [2]. Such queries are useful in emergencies, allowing the closest ambulance service to be located and contacted immediately after an accident, for example. Given that LDQ are usually executed by mobile users whose hand-held devices have limited resources, and that response times, especially in emergency situations, must be kept as short as

possible, our goal is to allow LDQ evaluations to be effected quickly without consuming a lot of resources.

To evaluate location-dependent queries, the client position has to be retrieved, often via positioning systems such as GPS. However, not every mobile terminal (PDA, cell phone...) is equipped with such a system. In addition, specific positioning systems are often dedicated to a particular environment; the GPS technology, for example, generally doesn't work indoors. To resolve these problems, we propose a positioning solution that provides an approximate position, which is nonetheless sufficient for evaluating LDQs. This positioning system can be used at any time, in any environment (indoors or outdoors), and does not require any particular infrastructure. This positioning system, and its optimization with regard to minimizing response time and economizing mobile device resources, is presented in the following pages.

This presentation is organized as follows: Section 2 reviews the existing positioning systems and their limits in terms of LDQ evaluation. In section 3, we introduce our positioning system, dedicated to the LDQ evaluation process. In section 4, we describe the simulations and the results to demonstrate the sufficient accuracy of the proposed positioning system. Section 5 describes the prototype used to validate the feasibility of using our positioning solution in the LDQ evaluation process and discusses the limits of our solution in terms of response time and energy consumption. Two optimization strategies, as well as the different experiments that were conducted to compare them in terms of environment, are presented in section 6. In section 7, we submit our conclusions and offer some perspectives for future research.

## **2. Existing Location Solutions**

Many systems over the years have addressed the problem of the location determination. Because each approach solves a slightly different problem or supports different applications, they vary in many parameters, such as the physical phenomena used for location determination, the

form of the sensing apparatus, their power requirements, the infrastructure or portable elements required, and their resolution in time and space. Some studies [3, 4] have proposed broadcasting geographic information to wireless device users so that they can evaluate their position. However, such solutions make it difficult for a particular user to estimate the accuracy and the precision of the broadcast information. To increase the precision of the location information broadcast to users, one solution is to increase the number of servers used to broadcast information. Unfortunately, since the information may come from several different servers, this solution complicates matters by making it very difficult for the handheld device to choose which location information to use.

Hightower & Borriello [5] have presented a taxonomy of location sensing systems. In it, they describe a variety of strategies for characterizing location systems. They can be grouped according to whether they use physical or symbolic location, absolute or relative location, or localized or distributed computation; and/or according to whether they give priority to accuracy, scalability, recognition or cost. They can be classified according to the environment they were designed for (indoor or outdoor) or their infrastructure requirements, if any (e.g. sensor networks and databases).

Location systems can also be characterized according to the type of positioning system they use: network-based or handset-based. The first—network-based Mobile Positioning Technologies (such as the Angle of Arrival Method [6], the Time Difference of Arrival Method [7] or the Radio Propagation Technique [8])—uses the mobile network to localize the mobile device, and thus the mobile user. The second—handset-based Mobile Positioning Technologies (such as SIM Toolkit, Enhanced Observed Time Difference (E-OTD), GPS or A-GPS)—relies primarily on the handset to localize the user. Some of the existing positioning systems are presented in the following paragraphs.

GPS, one of the most widely used positioning system, is perhaps the best-known or most recognized handset-based positioning system. It locally computes an accurate physical and absolute location (within a few meters) without needing to know the identity of nodes. Its infrastructure of satellites gives the system a planetary scale. Because satellite signals are blocked by walls and roofs, GPS is primarily intended for outdoor use; however it can be used indoors if GPS repeaters are employed. Assisted GPS (A-GPS) is a faster evolution of non-assisted GPS that makes use of additional network equipment to provide a position even when GPS is unsuccessful in acquiring any/enough satellites. Exact Location Identification [9] is another type of positioning system for outdoor use. Its infrastructure includes master and base stations which allow triangulation to be used to find the physical absolute position of identified mobile nodes. This system works well in the absence of signal reflection; otherwise, the accuracy is considerably affected.

The first archetypal indoor location sensing system, called Active Badges, was developed at AT&T Cambridge [10]. Each person wears a small infrared (IR) beacon that periodically emits a globally unique identifier. A server collects the emitted data from fixed IR sensors placed around the building, aggregates the data into a central repository, and provides an API that will allow applications to take advantage of this data. One of the primary disadvantages of this system is price: installing the Active Badge infrastructure in a building requires a significant initial investment. This system has been extended in the Xerox ParcTAB system [11] to address the problem of IR directionality.

Like the Active Badge location system, the Active Bat system [12] relies on electronic badges for personnel. However, instead of IR technology, Active Bat uses Radio Frequency (RF) to trigger ultrasound transmitters in the badges. According to its creators, the Active Bat system is more accurate than the Active Badge system (nine cm versus the size of a room). Within its range, Active Bat can pinpoint the exact locations of people and equipment, as long as they are

tagged with active badges. Cricket is another location system along these lines [7]. Also ultrasound based, it is less costly but also less accurate.

The Smart-Its Friends proximity-location sensing technique is also worth mentioning [13]. This technique involves a device that contains both a movement sensor and a transceiver. When a device senses movement, it broadcasts both its identity and the sensed movement to other devices, which receive and compare this information. The operating heuristic considers that two devices are in proximity if the recent movements to which they have been subjected are the same. Some indoor location systems (e.g. RADAR [8], WPS [14] and SpotON [15]) use RF signal strength to determine physical location. The first two use the signal strength of WLAN transmissions from/to WLAN Access Points to determine the position of the mobile user. The third, an ad hoc location sensing system, also uses signal strength to determine location, but it needs no expensive fixed infrastructure or central control like the systems mentioned above. Ad hoc systems like this one can be very useful in determining the position of mobile users in our environment, however SpotON is adapted to small scale environments such as offices with floor space less than  $16\text{m}^2$ .

Thus, all of the positioning and localization solutions described above present disadvantages for use in location-dependent applications, particularly for the evaluation of location dependent queries. In addition, such solutions are often expensive, are not universal, and the majority are dedicated to a specific environment: for example, GPS is generally used outdoors. Location-dependent applications, however, must be able to retrieve client position anywhere, at any time, even when the client devices are not equipped with geo-localization technologies. Therefore, in this article, we propose an approximate positioning solution that can be used in any environment.

### 3. Our proposition to locate mobile users

#### 3.1 Context

In this article, we focus on location-dependent applications deployed in wireless environments. To facilitate the management of these applications, we selected the hybrid Peer-To-Peer (P2P) architectural model. There are two types of peers—central and light. They are distinguished at the deployment of the application by the kind of resources they use [16]. Central peers are generally the more powerful of the two types. Because they are stationary devices, such as laptops, desktop PCs or servers, they have more vast resources. Light peers, on the other hand, are generally highly mobile handheld devices with very limited resources, such as PDAs and cell phones.

Our goal is to evaluate location-dependent queries (LDQ) in such P2P environments. As mentioned in the introduction, LDQ are queries, such as “What is the closest ambulance service to me?”, for which knowledge of user location is required in order to answer the question. We have already introduced three proximity operators—inside, closest and close—which can be used to express LDQs in a variety of query languages, such as SQL, XQuery and LDAP Protocol, for example [17]. (Fig. 1 shows an example of an LDQ using the closest operator, expressed in XQuery.)

```
<dsml:dsml xmlns:dsml="http://www.dsml.org/DSML">
  <TVRepairer>
  {
    for $i in document("pec.xml")//
      dsml:entry[dsml:objectclass/dsml:oc-value = "Vendor"]
    for $j in $i//dsml:entry[dsml:objectclass/dsml:oc-value = "Services"]
    where $j/dsml:attr[@name = "description"]/dsml:value = "repair"
    and closest($i)
    return
      $i/dsml:attr[@name = "name"]/dsml:value
  }
  </TVRepairer>
</dsml:dsml>
```

**Fig. 1. Example of location-dependant query expressed in XQuery**

We have also proposed a query execution model [18] that can evaluate LDQs. However, in order to work correctly, the query engine has to be able to determine the position of the terminal which expressed the query. Therefore, in this paper, we present the logical continuation of our previous work : an original positioning solution that will allow the position of a mobile client in a wireless

environment to be determined so that the location dependent queries can be evaluated. After this solution is described, we will discuss the optimization problems related to such a mechanism.

### 3.2 Positioning mobile clients in wireless environments

One of the original aspects of our approach is the adaptability of the localization process, which can be adapted to the resources of the underlying peer. Our solution can exploit positioning technologies like GPS; however, given that today's handheld devices often do not have GPS-like features and that such features are basically designed for outdoor use, the localization information used by our query engine to evaluate LDQs can also be computed using the information shared between neighboring peers. Thus, the user's position can be computed by anybody in any environment. In our opinion, the positioning solution can be approximate. In fact, when evaluating LDQs, the efficiency of the positioning solution in terms of response time and resource consumption is more important than its perfect accuracy. Given this point of view, we propose a positioning solution that relies on information-sharing between neighboring peers in a P2P environment.

Because stationary computers always have the same location, which is specified only once, evaluating the location of a query issued from such a computer is an easy task. On the other hand, defining the location of mobile peers is very difficult, since that location can change at any moment and so must potentially be re-evaluated for each query. So, we propose to compute an approximate location for mobile peers that is, nonetheless, sufficiently precise to allow LDQ to be evaluated. This approximate location is determined using environmental metadata, specifically information about an individual peer in relation to the peers to whom it is connected.

```

<locationDescription>
  <symbolicLocation>
    <city name = « Valenciennes »>
      <district name = « downtown »>
        <street name = « Bd Watteaux »>
          <number name = « 21b »>
          </number>
        </street>
      </district>
    </city>
  </symbolicLocation>
  <physicalPosition>

```

```

<lat> 27.7 </lat>
<long> -15.1 </long>
<alt> 197.4 </alt>
</physicalPosition>
</locationDescription>

```

**Fig. 2. Example of a locationDescription attribute file**

To compute an approximate location, five attributes for each remote neighboring peer must be managed. Using these attributes, each peer will be able to evaluate LDQs. These five attributes are listed below:

- The *locationDescription* attribute describes the position of the referenced peer. As shown in Fig. 2, the symbolic location (building, room, ...) and/or the physical position (latitude, longitude, altitude) [5] is/are represented in XML files, which allows location data semantics to be tracked and also facilitates location data management.
- The *locationLastUpdate* attribute represents the date of the last update of the *locationDescription* attribute file.
- The *connectionRange* attribute provides the network signal strength between the client and one neighboring peer. This range is used to estimate the distance between two peers. Indeed, the greater the network signal strength, the shorter the distance separating the two peers. However, this distance does not correspond to the Euclidean distance since the signal strength varies due to the physical obstacles (walls, windows, doors, ...).
- The *mobilityProfile* attribute describes the mobility profile of one neighboring peer. For each mobile peer, a category is selected: very mobile, mobile, not very mobile, not mobile. Then, according to its locomotion type (car, bus, pedestrian, ...), an average travel rate is deduced and stored as the *mobilityProfile*.
- The *connectionState* attribute determines if the remote peer is connected or not.

The details related to treating and updating these attributes will be described in the following section.

We propose an algorithm, which can be used to evaluate the location of a mobile peer. It uses an approximation degree, calculated using the *locationLastUpdate*, *mobilityProfile* and

*connectionRange* attributes, to estimate the quality of the computed location. The smaller the degree, the better the quality of the location. This algorithm produces a list of couples (location, approximation degree), sorted by approximation degree. In this algorithm, the *Nb* variable is defined in terms of environmental constraints and indicates the maximal number of locations to be computed. The smaller *Nb* is, the shorter the execution time of the evaluation will be.

First, if a light peer is connected to other peers, then the variables  $m_i$ ,  $d_i$  and  $r_i$  are defined for each connected referenced peer. The  $m$  variable corresponds to the *mobilityProfile* attribute. The  $d$  variable corresponds to the difference between the current date and the *LocationLastUpdate* attribute. Thus, the larger  $d$  is, the more out-of-date the location data is. The  $r$  variable defines a proximity index corresponding to the *connectionRange* attribute in order to select the closest peers. The  $m_i$ ,  $d_i$  and  $r_i$  variables are stored in three tables, called respectively tables  $m$ ,  $d$  and  $r$ . For central peers, the  $m_i$  and  $d_i$  variables will always be null, the first because these peers are not mobile and the second because their location is always up-to-date.

Then, the approximation degree can be calculated for each connected peer. This degree is  $a*r_i$  for a neighboring central peer, or  $m_i*d_i + b*r_i$  for a neighboring light peer. Coefficients  $a$  and  $b$  may be used to balance the priorities set to compute the “best” locations. Finally, a location list of *Nb* couples is sorted according to this approximation degree. *Nb* corresponds either to the *Nb* variable defined in the evaluation function, or to the number of connected peers (*nbConnectedPeers*) if  $Nb > nbConnectedPeers$ .

To illustrate, let us consider a configuration in which a peer  $p$  has 6 neighbors: 3 central peers and 3 light peers. The different attributes of these peers are stored on  $p$ . (These attributes are described in table 1.) The current date in the example is Apr-14th-2004 at 2:43 pm. Coefficients

<i>Referenced Light Peers</i>	<i>Referenced Central Peers</i>
LP = LP1 locationDescription = «LP1.xml » locationLastUpdate = Apr-14th-2004 1:00 pm connectionRange = 2 mobilityProfile = 10 connectionState = false	CP = CP1 locationDescription = «CP1.xml » locationLastUpdate = sysdate connectionRange = 60 mobilityProfile = 0 connectionState = true

LP = LP2 locationDescription = «LP2.xml » locationLastUpdate = Apr-14th-2004 2:33 pm connectionRange = 3 mobilityProfile = 10 connectionState = true	CP = CP2 locationDescription = «CP2.xml » locationLastUpdate = sysdate connectionRange = 90 mobilityProfile = 0 connectionState = false
LP = LP3 locationDescription = «LP3.xml » locationLastUpdate = Apr-14th-2004 2:16 pm connectionRange = 30 mobilityProfile = 10 connectionState = true	CP = CP3 locationDescription = «CP3.xml » locationLastUpdate = sysdate connectionRange = 30 mobilityProfile = 0 connectionState = true

**Table 1. Example of environment metadata**

$a$  and  $b$  are set to 1 in order to avoid privileging either type of peer.  $Nb$  is set to 3. To evaluate a location-dependent query expressed by peer  $p$ , its location must be computed. The execution of the Evaluation function shown in Fig. 3 is presented below.

Function Evaluation

```

Nb := 3 ;
// Firstly, the algorithm retrieves the set of neighbored peers referenced on the peer client.
RefPeerList = {LP1, LP2, LP3, CP1, CP2, CP3} ;

//Then, the set of connected neighbored peers is selected.
ConnectedPeerList = {LP2, LP3, CP1, CP3} ; nbConnectedPeers := 4 ;

//The set of variables is defined :
d = [10,27,0,0] ; r = [3,30,60,30] ;
//For each connected peer, an approximation degree is calculated (a*r[i] for a central peer or //b*d[i] + c*r[i]
for a light peer).
locationList = {(LP2.xml, 103), (LP3.xml, 300), (CP1.xml, 60),(CP3.xml, 30)} ;

//Then, the list is sorted in function of approximation degrees to obtain the final result
LocationList = {(CP1.xml, 30), (CP3.xml, 60), (LP2.xml, 103)} ;

```

This algorithm returns a list of locations sorted according to their approximation coefficients. These coefficients are very important since they facilitate the evaluation of the LDQ [THI04]. In rare cases, the light peer location cannot be evaluated because the evaluation function returns a null/empty list. In such cases, an interactive system can be proposed, allowing the light peer user to describe his/her location.

## 4. Accuracy Evaluation

### 4.1. Simulations

To evaluate a LDQ, the localization of mobile client does not need to be very accurate. Nevertheless, it is very important to evaluate the average accuracy of our algorithm in different environments. To estimate the accuracy of mobile client's computed position, a simulator has

been implemented. In this simulator, an environment constituted of a building which can be a train station for example and the close surroundings of this building. During the simulations, the number of participants and the density of central peers vary. To each mobile client, the position algorithm is executed and the result is represented by a list of couples  $\{(location.xml, approximation\ degree)\}$ . The connectionRange attribute based on the signal strength between the client and one of his neighbours is computed in function of the distance and the different obstacles (like walls or doors for example) separated the client and his neighbour. Indeed, the environment representation is also based on the representation of the obstacles and for each obstacle type, an attenuation coefficient is associated. This coefficient corresponds to the attenuation of signal when it meets the obstacle. These different attenuation coefficients are based on experiments with 2 laptops and a WiFi network in ad hoc mode.

Then, for each location xml file in the list, a weight is associated to it. This weight is the inverse of approximation degree. To evaluate the accuracy of computed location, the simulator computes the distance between the exact position of mobile client and the balanced barycentre of different locations of the list. Let us consider L the location of mobile client issued of the algorithm :

$$L = \{(loc_1.xml, Approx\_degree_1) ; (loc_2.xml, Approx\_degree_2) ; \dots \}$$

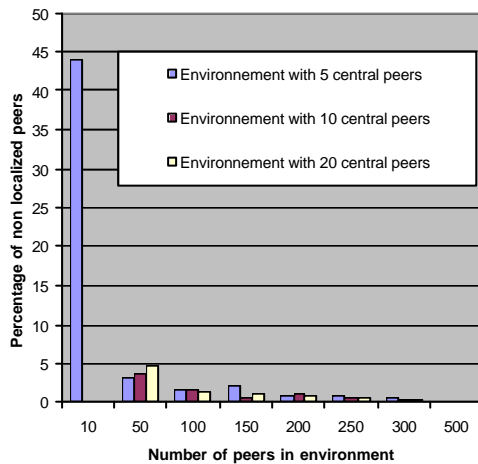
Based on L, we calculated the L' list composed by xml location files and their associated weights :  $L' = \{(loc_1.xml, weight_1) ; (loc_2.xml, weight_2) ; \dots \}$ . To each xml location, a latitude, a longitude and an altitude are associated. Thus, we calculate the balanced barycentre :

$$lat_{barycentre} = \frac{\sum weight_i * lat_i}{\sum weight_i} \text{ for each } i \text{ comprised between } 1 \text{ and the number of elements in } L.$$

Thus, we obtain as many balanced barycentres of elements in the list L. To each barycentre, the distance between exact location of client and the barycentre is computed. This distance represents the different accuracy of barycentre position compared to exact client position.

#### 4.2. Analysis & Conclusions

The selected environment represents a train station : the different obstacles have been parameterized and the total surface is around 7000 m<sup>2</sup>. The number of peers in the environment vary from 10 to 500 and the number of central peers vary from 5 to 20. Thus, these different configurations allow to represent a variety of environments: dense environments with a high number of participants and sparse environments with a relatively low number of peers; well connected environments with a single communication area that permits direct peer-to-peer access to all participants, and "poorly" connected environments with several communication areas that do not allow direct peer-to-peer access to all participants.



**Fig. 3. Evolution of localized peers in function of number of peers in the environment**

Overall, the results are quite conclusive. The accuracy of computed localization of mobile client are good. Based on the simulations results, we have highlighted several conclusions. Firstly, in "poorly" connected environment, the localization of client can not always be computed. Indeed, because the pool of available peers in such an environment is often quite small, it is more difficult to obtain a position. As shown in Fig. 3, the percentage of non-localized peers is important when there are only 10 peers in the environment. However, this percentage becomes very small when the number of peers in the environment increases. On the contrary, the number of central peers do not influence the percentage of localized peers.

We have also highlighted that more elements in the computed localization list, better the localization accuracy. As shown figures 4 a-b-c-d, the accuracy of computed localization is good and this accuracy increases when the number of computed localizations (from the list of couples) increases. Nevertheless, the accuracy is better and is about stabilized when the number of computed localizations is around 7. This attenuation of the accuracy's increase is also illustrated by the Fig. 5.

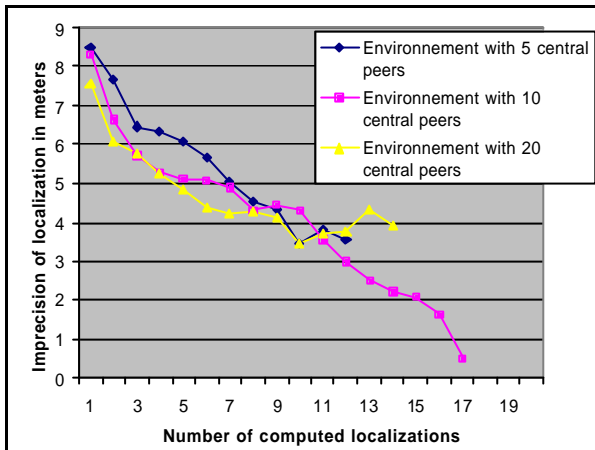


Fig. 4a. Environment with 50 peers

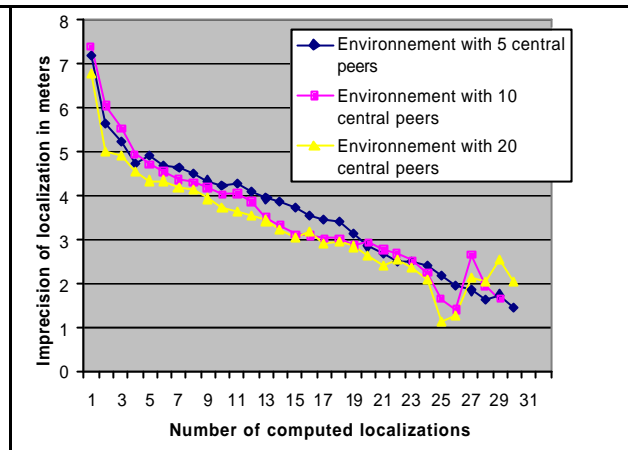


Fig. 4b. Environment with 100 peers

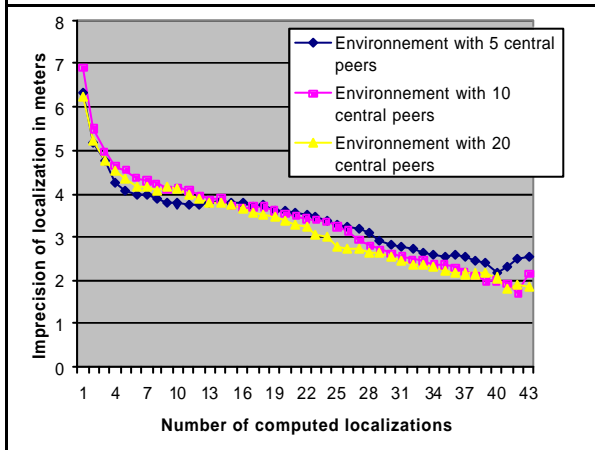


Fig. 4c. Environment with 200 peers

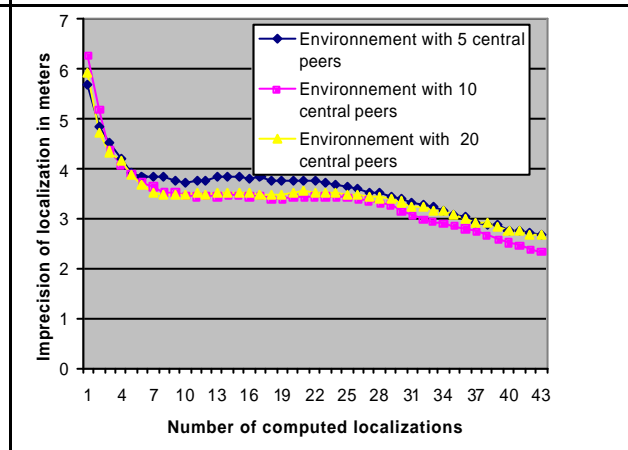


Fig. 4d. Environment with 300 peers

Finally, we can conclude that the computed localizations thanks to our algorithm is enough accurate to evaluate LDQs. If the client peer has at least one neighbor, the average maximal imprecision of his computed localization is less than 9 meters.

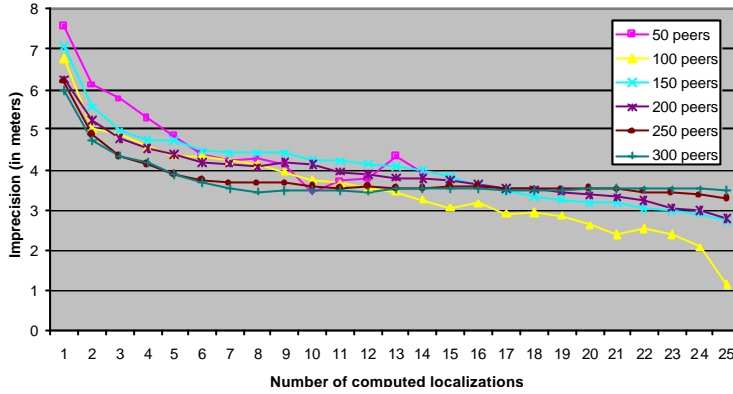


Fig. 5. Environment with 10 central peers

## 5. Prototype

A prototype of the work presented in this article has been tested and demonstrated at the French BDA conference [19]. This prototype exploits a proximity electronic commerce application that allows potential customers equipped with a PDA to receive broadcast offers targeting their preferences when they are in the shopping mall. In addition, a client can express location-dependent queries, such as “What is the closest music store to me?”, and retrieve the results. In our prototype example, the potential clients are equipped with iPAQs H3950 with a 802.11b network access, whereas vendors have desktop computers equipped with a Wifi card. A storage manager is deployed on each peer to store the set of available data, as well as the query engine needed to express and evaluate LDQ. The query engine uses the positioning algorithm described in the previous section.

However, that first version also highlighted some of the limits to our approach, particularly in terms of resource consumption and response time. Some of the limitations can perhaps be partially explained by the naïve way that the positioning solution in the first version of prototype was implemented. In fact, our original goal was to quickly validate the quality of the LDQ evaluation’s results, without optimizing the performances. Because it provided an entire positioning solution for each LDQ evaluation, the implementation required a lot of P2P communication, with much of the local computing done on the handheld device.

In this first prototype, the entire set of environmental metadata was retrieved each time a LDQ was evaluated: the client peer broadcasts a call over the communication area in order to retrieve the environmental metadata; stored all the attributes (*locationDescription*, *locationLastUpade*, *mobilityProfile* and *connectionState*); retrieved the *connectionRange* attributes and associated them to each location file for each neighbored peer; and executed the algorithm to evaluate the client's position. Obviously, this was a drain on the handheld device's resources.

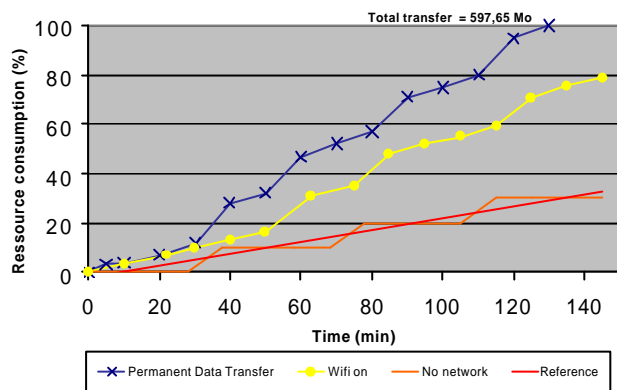


Fig. 6. Evolution of the energy consumption

The limitations in terms of response time and resource consumption can also be attributed to the use of a Wifi network. First of all, as shown in Fig. 6, connecting a handheld device via a Wifi card consumes a fair amount of energy, even if no file transfer is performed.

In addition, the data rates obtained in our experiments are very low and thus the times of transfers are long. Indeed, the devices' underlying resources themselves have an impact on transfer times. For example, the data rates obtained on a transfer from an iPAQ to a laptop (or visa versa) are much lower than those obtained for a transfer between 2 laptops.

And finally, the cost of file manipulation on the handheld device can also involve the limitations in terms of response time and resource consumption. Indeed, the writing and reading costs are very high.

## 6. Optimization

Given the observations in section 4 concerning the limitations of our first prototype, we propose two optimization strategies that will help to minimize resource consumption and response time.

### 6.1. Strategies

In the first version of our positioning solution, the recovery and computing of the environmental metadata are clearly the most costly steps. That first prototype required a client peer sending an LDQ to retrieve and locally compute all the metadata for each LDQ evaluation. To reduce this burden, we propose two optimization strategies for the treatment of environmental metadata, designed to reduce both response time and energy consumption.

#### **A strategy based on Neighboring Peers**

In the first strategy, the client peer broadcasts a message over the communication area, and based on the network strength, each connected peer calculates the *connectionRange* attribute in order to provide the client peer with the approximate distance between itself and the client peer. Once the client peer has received all the “distances”, it selects the best neighboring peers and sends a call to these peers. Finally, the client peer retrieves their location files.

#### **Selection Level strategy**

The second strategy is a 2-step strategy. These two steps can be computed separately. In the first step, the client peer sends a broadcast call to discover the set of connected peers in the communication area. Based on the number of connected peers and the type of networks used by each connected peers, it calculates a selection level. This selection level corresponds to the “maximal distance” authorized between the client peer and a neighbored peer to select the localization file of the neighbored peer. The computing of the selection level is based on the average density of neighbored peers around the client peer calculated in function of the average maximal ranges of types of used networks. Indeed, the client peer receives the used type of network to communicate from each neighbored peer. Also, it is able to determine the number of

accessible neighbored peers for each available network. In the following, we illustrate the computing of the selection level using an environment example described in the figure 8. In this environment, the client peer is connected with peers thanks to 2 types of networks  $R_1$  and  $R_2$ . An average maximal range (in meters) is associated to each network :  $P_1$  for  $R_1$  and  $P_2$  for  $R_2$ . The neighbored peers which use the network (respectively the network ) are in the sphere (respectively ). The center of these spheres are the client peer, the radius are respectively  $P_1$  and  $P_2$  and the volumes are respectively  $V_1 = \frac{4}{3} * \boldsymbol{p} * R_1^3$  and  $V_2 = \frac{4}{3} * \boldsymbol{p} * R_2^3$ .

$V_1 = \frac{4}{3} * \boldsymbol{p} * R_1^3$  and  $V_2 = \frac{4}{3} * \boldsymbol{p} * R_2^3$ .

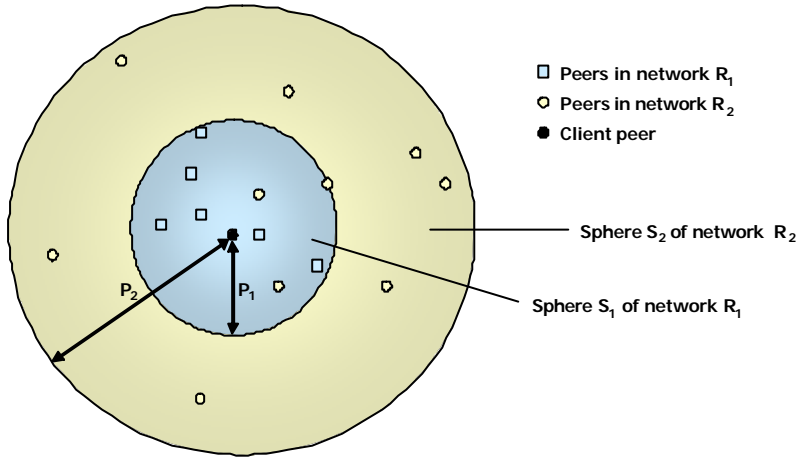


Fig. 7. Environment example of a client peer

$n_1$  and  $n_2$  represent respectively the number of peers contained in  $S_1$  and  $S_2$ . If  $S_1$  is smaller than  $S_2$  like in the figure 7,  $n_2'$  represents the number of neighbored peers using the network  $R_2$  in the

sphere  $S_1$ , thus  $n_2' = \frac{(n_2 * V_1)}{V_2}$ .

$nb_{peers}$  represents the number of required localization files to allow the client peer localization (to allow a correct evaluation of the client location, 7 localization files to represent the client localization are sufficient).

If  $nb_{peers} \leq n_1 + n_2'$ , the sufficient and required volume to communicate with  $nb_{peers}$  is

$$V_{suff} = \frac{V_1 * nb_{peers}}{n_1 + n_2'} \text{ and thus } Selection\_Level = \left[ \frac{3 * V_{suff}}{4 * \boldsymbol{p}} \right]^{\frac{1}{3}}.$$

If  $nb_{peers} > n_1 + n_2'$ , the sufficient and required volume to communicate with  $nb_{peers}$  is

$$V_{suff} = V_1 + \frac{(nb_{peers} - (n_1 + n_2')) * (v_2 - v_1)}{n_2 - n_2'} \text{ and thus } Selection\_Level = \left[ \frac{3 * V_{suff}}{4 * p} \right]^{\frac{1}{3}}.$$

Else,  $Selection\_Level = R_2$ .

To compute the selection level, the types (central or light) of neighbored peers can also be used to refine the selection in the following. Each neighbour's response contains the type of neighbour. For example, if the client peer is in an environment which contains numerous central peers, the client peer can base the computing of the selection level only on the number of neighbored central peers.

Then, in the second step, when it needs to send an LDQ, it is only necessary to execute the second step to retrieve the environmental metadata:

- The client peer broadcasts the selection level over the communication area.
- Each connected peer calculates the *connectionRange* attribute, and uses this attribute to estimate the distance separating it from the client.
- If this distance is shorter than the selection level calculated in step 1, the connected peer sends its location file and the approximation degree (the approximate distance) to the client peer.

## 6.2. Experiments and Results

This section presents the results of our experiments concerning the response times for retrieving the client peer's position relative to its environment. Based on these results, the energy consumption of the client peer can be deduced. It is very difficult to obtain accurate measurements of battery consumption because the iPAQ battery management tool is very imprecise. Thus, we compared the execution times of the different strategies and used the relationship between them to deduce the various consumption levels. The experiments used a handheld device as client peer.

Fig. 8 illustrates the response times obtained using our original "naïve" implementation. In this naive implementation, every neighbouring peers' location file must be stored on the client peer to allow the client's position to be computed. The response times using the naïve implementation are not negligible, particularly when the mobile client is surrounded by numerous peers. However, our most recent tests demonstrated that 7 location files will produce a satisfactory response to an LDQ. Thus, in the two optimization strategies, the maximal number of location files to be retrieved is 7. Our experiments highlighted that, by using only central neighbored peers, the transfer times for the two optimization strategies can be stabilized, even if the neighbored peers are numerous. This stabilisation is important for minimizing resource consumption.

Fig. 9 presents the results for the Neighbouring-Peers Strategy. The red bars, signifying computation by remote light peers, represent the maximal time obtained if every connected peer is a light peer, and if all of them agree to compute the *connectionRange* attribute. (Although the connected light peers are part of the positioning process, they are free to agree, or not, to make the computation.)

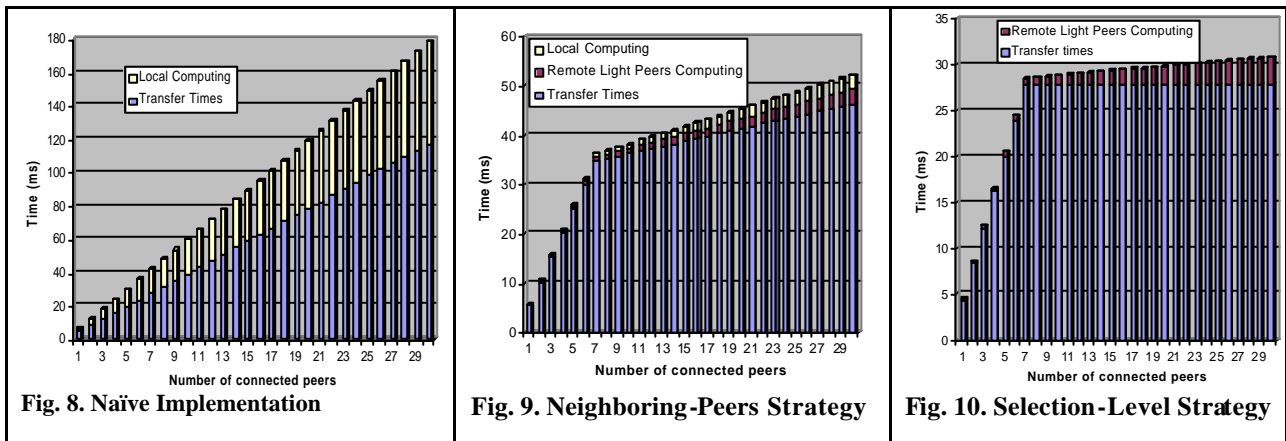


Fig. 10 presents the results obtained for the second step of the Selection-Level strategy. Though not depicted in Fig 10, the first step is computed normally and is taken into account in calculating the client peer's overall resource consumption.

The times obtained using two optimization strategies are much better than those obtained using the naïve implementation. The Selection-Level Strategy provides a client location more quickly

than the Neighboring-Peers strategy. Nevertheless, the Selection-Level Strategy can not be used in every environments by every peers. Indeed, both optimization strategies can be adapted to client behaviour and environment. For example, the Neighbourhood-Peers Strategy would be more appropriate for clients who do not often express LDQ; this strategy does not require that the handheld device be continually aware of its environment, which is the effect of step 1 of the Selection-Level Strategy.

## **7. Conclusion & Perspectives**

In this article, we have presented a positioning system to locate mobile users in wireless environments and to evaluate location-dependent queries. The position computed by our positioning system is approximate, but it can be computed rapidly and therefore, economizes battery consumption. Short response times are very important in emergency situations, when looking for the closest ambulance in the event of an accident, for example. Moreover, given that mobile devices are often equipped with limited resources, energy consumption is also an important consideration. For these reasons, the prototype of our proposed positioning mechanism, and its several optimization strategies, works to minimize response time and energy consumption during the positioning process.

Because the information required to respond to LDQ is most often not stored on the client mobile device, queries must be distributed on a set of connected peers to be evaluated. However, as we have demonstrated in this article, the wireless communications are very costly, and so this query distribution must be optimized. The proposed optimized positioning solution, provides a very interesting perspective on a cost-effective model for query evaluation. Indeed, if the costs of the different proximity operators (close, closest and inside) can be determined, the query evaluator can then determine the optimal evaluation plan for each location-dependent query that must be evaluated.

## 8. References

- [1] U. Varshney, R. Vetter, Emerging Mobile and Wireless Networks, *Communications of the ACM*, Vol. 43, Issue 6, 2000.
- [2] A. Y. Seydim, M. H. Dunham, V. Kumar, Location Dependent Query Processing, Proceeding of MobiDE, 2001.
- [3] J. Zhang, L. Gruenwald, Prioritized Sequencing for efficient Query on Broadcast Geographical Information on Mobile-Computing, *ACM Int. Symp. on Advances in geographic information systems (GIS)*, 2002.
- [4] J. Il-dong, Y. Young-ho, L. Jong-hwan, K. Kyungsok, Broadcasting and Caching Policies for Location-Dependent Queries in Urban Area, *Int. workshop on Wireless sensor networks and applications (WMC)*, 2002.
- [5] J. Hightower, G. Borriello, Location Systems for Ubiquitous Computing, *IEEE Computer*, 2001.
- [6] D. Niculescu and B. Nath, Ad Hoc Positioning System (APS) using AOA, *IEEE INFOCOM*, 2003.
- [7] N.B. Priyantha, A. Chakraborty and H. Balakrishnan, The Cricket Location-Support System, *6th Annual Int. Conf. on Mobile Computing and Networking (Mobicom)*, 2000.
- [8] P. Bahl and V. Padmanabhan, RADAR: An In-Building RF-Based User Location and Tracking System, *IEEE INFOCOM*, 2000.
- [9] K. Sinha and N. Das, Exact Location Identification in Mobile Computing Networks, *Int. Workshop on Parallel Processing (ICPP)*, 2000.
- [10] R. Want et al., The Active Badge Location System, *ACM Transactions on Information Systems*, 1992.
- [11] R. Want et al., The Parctab Ubiquitous Computing Experiment, *Book Chapter: "Mobile Computing"*, Kluwer Publishing, Edited by T. Imielinski, Chap. 2, 1997.

- [12] A. Harter, A. Hopper, P. Steggle, A. Ward, and P. Webster, The Anatomy of a context-Aware Application, *5th Annual ACM/IEEE Int. Conf. on Mobile Computing and Networking (Mobicom)*, 1999.
- [13] L.E. Holmquist, F. Mattern, B. Schiele, P. Alahuhta, M. Beigl and H.-W. Gellerson, Smart-Its Friends: A Technique for Users to Easily Establish Connections between Smart Artefacts, *UBICOMP*, 2001.
- [14] Y. Wang, et al. An indoors wireless positioning system based on wireless local area network infrastructure, *Int. Symp. on Satellite Navigation Technology Including Mobile Positioning & Location Services (SatNav)*, 2003.
- [15] J. Hightower, R. Want and G. Borriello, SpotON: An Indoor 3D Location Sensing Technology Based on RF Signal Strength, *Technical Report #2000-02-02*, 2000.
- [16] M. Thilliez, T. Delot, S. Lecomte, N. Bennani, Hybrid Peer-to-Peer Model in Proximity Applications, *Int. Conf. on Advanced Information Networking and Applications (AINA)*, 2003.
- [17] M. Thilliez, T. Delot, A Localization Service for Mobile Users in Peer-To-Peer Environments, in *LNCS book on Mobile and Ubiquitous Information Access*, pp. 271-282, ISBN 3-540-21003-2, Volume 2954, 2004.
- [18] M. Thilliez, T. Delot, Evaluating Location Dependent Queries Using ISLANDS, *Int. Symp. and School on Advanced Distributed Systems (ISSADS)*, 2004.
- [19] M. Thilliez, Y. Colmant, T. Delot, Query Evaluation in ISLANDS, *Les 19èmes journées de Bases de données Avancées (BDA)*, 2003.