

# A Comparison of Protocols for Updating Location Information

Alexander Leonhardi and Kurt Rothermel

Institute of Parallel and Distributed  
High-Performance Systems (IPVR)  
University of Stuttgart  
Breitwiesenstr. 20-22  
70565 Stuttgart, Germany

*alexander.leonhardi@informatik.uni-stuttgart.de*

Technical Report 2000/05  
Department of Computer Science  
University of Stuttgart  
March 2000

## Abstract

The detailed location information of mobile objects, for example that of a user with a mobile computer or phone, is an important input for many location-aware applications. However, constantly updating the location information for thousands of mobile objects is not feasible. Update protocols for location information therefore use the special properties of this information to transmit it as efficiently as possible, that is requiring only few update messages, while still being effective in returning the location information with the desired accuracy. Different classes of such update protocols are described in this paper and a new combined protocol is proposed. To be able to compare their effectiveness and efficiency, we present an analysis for the minimum and average resulting accuracy of the location information on the receiver and the number of messages transmitted between them. We also present some simulation results, which we have performed to back up our analysis.

## 1 Introduction

As detailed information about the location of a mobile device or user has become widely available, mainly through the satellite positioning system GPS, more and more applications are using this information. Examples are car navigation or fleet management systems, as well as location-aware information services for cellular phones (which today use the less accurate information of the communication cell a mobile phone is currently located in). Most of these applications only consider the location of a single user. Further functionality, like determining all mobile objects inside a given area or generating an event whenever a mobile object enters a room, requires a special *location service*, which collects and manages the location information for all of these objects. A universal global location service has been proposed for example by [Leo98] and [Maa97]. Such a location service has to deal with all sorts of mobile objects with various movement characteristics, for example objects that move continuously for a longer period of time, like cars and trucks, or objects that only move seldomly, like pieces of office equipment, for

example a printer. The location information for these objects may have been acquired by different types of sensor systems and therefore has different degrees of accuracy.

If the location information is acquired on the mobile object itself (e.g., via GPS), it has to be transmitted from the mobile object to a location server using wireless communication. In case of a distributed service, where the location information is cached or replicated on a number of servers, the information also has to be transmitted between location servers. To control the transmission of location information, different update protocols with varying properties can be used. In case of a querying protocol the information is pulled by the receiver, while with a reporting protocol it is pushed by the sender. A combined protocol is also possible, where an optimal ratio between the number of updates and the number of queries has to be found.

In this paper we will discuss the characteristics of these classes of update protocols and their subclasses as well as their strengths and weaknesses according to different areas of application. To be able to guarantee the receiver a certain accuracy of the returned location information, it usually has to be transmitted rather frequently. If the location information is transmitted from a mobile device to a location server, a wireless channel has to be used, where bandwidth is low and expensive. It is therefore important to use an update protocol that works efficiently, that is requires as few messages as possible, and effectively, that is achieves a desired accuracy of the location information on the server. In this paper we discuss the efficiency of the basic types of update protocols by means of an analysis of the number of transmitted messages and their effectiveness by analyzing the resulting minimum and average accuracy of the location information at the receiver. Although this work has been performed in the context of updating the location information on a location service, its results are applicable wherever location information is transmitted from a sender to a receiver.

The remainder of this paper is structured as follows: In Section 2 we look at related work and in Section 3 we describe the background for this paper as well as its technical environment. The different classes of update protocols and

their properties are discussed in Section 4, while Section 5, besides defining a data-model for the location information and its accuracy, compares their efficiency and effectiveness by means of an analysis. In Section 6 we present some simulation results, to support the outcome of our analysis. Finally, Section 7 contains our conclusion, as well as the plans for future work.

## 2 Related Work

Architectures for a location service have been discussed in the area of location-aware applications as an important component of such systems. However, different update protocols for the transmission of the location information have not been compared in detail. Early location services were usually developed for a particular positioning system, like the location service for the Active Badge system, which is described in [HH94]. For a similar positioning system, the ParTab system developed at Xerox Parc, a location service has been created, where the location information of a mobile object or user is protected by a dedicated Home Agent, which controls access to it [ST94]. In this context the problem of disseminating the location information to a large number of clients has been discussed in [ST94]. Clients with similar queries receive the location information over dynamically created multicast groups in which the responses to these queries are grouped. In his PhD-thesis, Leonhardt ([Leo98]) proposes a universal location service, which is independent from the types of applications accessing it and the types of positioning systems it gets the location information from. A detailed location model is presented that integrates different types of location information. Another main concern of this work is privacy and security. For this purpose it defines requirements and policies for access control of a location service. The thesis also contains some thoughts about the design of a global, general-purpose location service but does not propose or evaluate a specific architecture.

The management of the location information of their mobile phones is a very important issue in the area of mobile communication networks. When a call has to be forwarded to a mobile phone, a mobile communication network sends a paging message to all the cells the mobile phone may be in, whereupon the corresponding mobile phone answers to accept the call. The mobile phone on the other hand periodically reports its current cell, which is updated in the Visitor Location Register and Home Location Register. Various update strategies have been discussed with the goal of minimizing the overall number of paging and update messages. In [BNKS95] a distance-based, a movement-based and a time-based protocol are compared. In this special application area the location information is a discrete identifier, describing a certain cell of the communication network. Today, these cells have a size of about 1 to 10 miles but are expected to become smaller with the introduction of future technologies. Update policies for location management have been optimized for these special requirements. In [BD99] the LeZi-update approach is described, which uses the cell names as an alphabet for minimizing the updates by a mechanism similar to the Lempel-Ziv compression. In our work we have assumed that the location information is much more accurate, acquired for example by a GPS sensor, and that the location information can have different levels of accuracy. The location service, these update protocols are intended to be used for, also allows different types of queries (e.g., range queries), which are not supported by the location management components of mobile communication networks.

In the DOMINO project a database for the tracking of mobile objects is being developed (see [WSCY99]). One intended area of application is fleet management for the trucks of a transportation company. To reduce the number of updates, it is assumed that the route, on which the mobile object is traveling, is known to the database in advance and that the object informs the database when it changes its route. Different dead-reckoning strategies are proposed and compared, where the database estimates the current location of a mobile object on this route based on its speed and the object sends an update of its location when it differs from this estimation by more than a certain threshold. The different strategies have different ways for determining this threshold or for adapting it dynamically. These dead-reckoning protocols have been designed on the assumption that the route of the mobile objects is known beforehand. In this paper we examine update protocols from a more general point of view, where we assume no knowledge of the mobile objects' future movements.

## 3 Background

This section gives an overview of the technical environment our work is based on, namely the sensor systems the location information is acquired from and the network environment it is transmitted over. Finally, we describe the architecture for a universal distributed location service, which is being developed at our Institute and where we plan to apply the results of this work.

### 3.1 Positioning Sensors

The location information for a mobile object can be determined through various types of positioning systems. A basic distinction can be made between *tracking systems*, where a system of stationary sensors determines the location of a mobile object, and *positioning systems*, where the location information is determined by a sensor on the mobile object itself. A typical tracking-system is the Active Badge system (see [WHFG92]), where an Active Badge carried by a user or attached to an office device periodically emits an infrared beacon, which is detected by an infrared sensor placed in each room of a building. The widely used global positioning system (GPS) is a positioning system, where a sensor determines its own position by taking a bearing on four of a whole of twenty-four special satellites (see [HWLC97]). Different types of positioning systems return the location information with varying formats and degrees of accuracy. It can be *symbolic* (i.e., the identifier of a region, like a room or communication cell) or *geometric*, where it is described by a global geodetic coordinate system like WGS84 (see [IA97]). The classifications in this paragraph have been taken from [Leo98].

Because no positioning system offers a global coverage (e.g., GPS works only outdoors) a universal location service has to be able to integrate the location information acquired by all types of positioning systems.

### 3.2 Network Environment

In our work, we have assumed the following network environment: Location servers and stationary clients, which query the location information, communicate over a fixed network. Mobile devices need to be connected to this network by a wireless link. They have to be able to communicate and to be contacted independently from their geographical location (e.g., using a mechanism like Mobile IP [Pe96]), because the

device has to transmit location information to a server via the wireless link or is queried by the location server through it, if the information is determined locally. Mobile devices may also act as a client of the location service by posting queries over the wireless link. The wireless network can either be a wireless WAN, like GSM [MP92], a MAN, like the Metricom Ricochet network, or a LAN, for example according to the IEEE 802.11 standard [Com97].

A common characteristic of these wireless networks is that a connection can be temporarily lost while the device is at an unfavorable location (e.g., inside of a tunnel), a state which is called a disconnection. In most cases a wireless network can also not offer as good a bandwidth and latency as a fixed networks (see [Sat96]).

### 3.3 A Universal Location Service

This comparison of update protocols has been done as part of our efforts to design and develop a universal distributed location service (see [LK99]). As mentioned before, this service will have to collect and integrate the location information from various sensor systems. Furthermore, it will provide a general interface through which different types of applications can access the location information in a general way. For reasons of scalability and to achieve short response times as well as high update rates, the information will be distributed between a number of location servers.

The location service shall provide the following functionality: It will support *range queries*, that is finding all mobile objects inside a given area, as well as *position queries*, which request the current location of a certain mobile object. A more specialized query is, for example, the search for the nearest object to a certain location. It is intended, that the location service will additionally support an event mechanism, where the client can define a certain predicate and is notified by the service whenever this predicate becomes true. A client program may for example be notified, whenever a mobile object enters or leaves a given area or when a mobile object meets another one, that is comes within a certain distance from it.

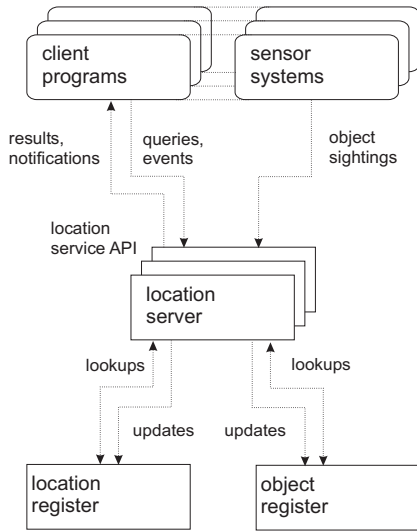


Figure 1: Main components of a universal distributed location service

We have proposed an architecture for such a location service, which is shown in Figure 1. Its components and their functionality are described next:

- *Clients and sensor systems*: Clients of the location service are location-aware applications, which run either on a mobile or on a stationary computer and query the location servers using the interface mentioned earlier in this subsection. A client on a mobile device is usually also equipped with a positioning sensor or is being tracked by one. Sensor systems, which, depending on their type, determine the location information for one or more mobile objects, are associated with a certain location server and update the location information for these objects.
- *Location server*: Each location server is responsible for a certain geographical area and stores the location information for all mobile objects that are currently inside of this area. If a mobile object leaves the area, a hand-over has to be performed with the server, into whose area the object has moved. A location server answers the clients' queries concerning the mobile objects it is responsible for and forwards other queries to the appropriate server(s). This is done by accessing one of the two following registers:
- *Object register*: The data, which belongs to a single mobile object, is stored in the object register. It can contain general information about the object or information that is determined by the location service, like the maximum speed of the object. The object register also stores the addresses of the location servers the location information for this object is currently stored on. It can easily be distributed, so that each register server is responsible for a certain set of objects.
- *Location register*: The location register stores the association between the location servers and the area they are responsible for. It is used to find the appropriate location servers for a range query. If necessary, the location register can be distributed across different servers, which are organized in a tree structure similar to the Domain Name System (DNS) of the Internet.

Because of privacy concerns, security and access control for such a location service are very important issues. A user should be able to control in detail who and to what degree has access to his location information. Some solutions for this problem have been presented in [Leo98]. In this paper we do not discuss privacy any further, but will address the issue in future work.

## 4 Update Protocols

In this section the different classes of update protocols are introduced and their main properties are discussed. The protocols are used to update a remote *secondary copy* of the location information for a certain mobile object based on the contents of a *primary copy*. The goal is to guarantee a given accuracy of the location information in the secondary copy. The information of the primary copy is either determined directly by a positioning system or is the information stored on another location server. We call the component which manages the primary copy the *source* of the location information. The secondary copy is usually stored on a special or general location *server*, which is queried by local or remote applications. Figure 2 shows the components which are involved in the transmission of the location information.

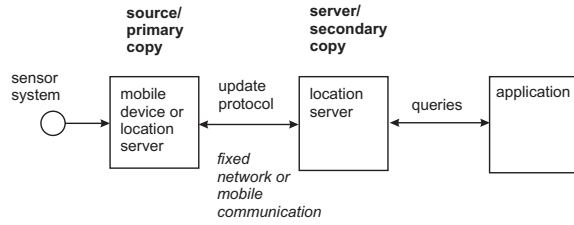


Figure 2: Components involved in the updating of location information.

#### 4.1 Classification

The update protocols can be divided into three main classes, namely querying, reporting and combined protocols, where each class has a number of typical variants. Each of these protocols has its characteristic properties and is suitable for a given environment or for certain requirements. Some of the reporting protocols are similar to the paging/update schemes that are used for Location Management in the research area of Personal Communication Service (PCS) networks and are summarized, for example, in [BNKS95]. The classes of update protocols and their relationships are shown in more detail in Figure 3.

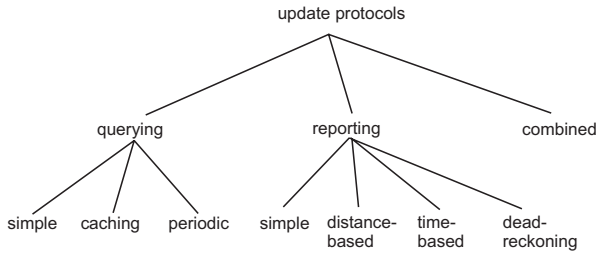


Figure 3: Classification of update protocols for the transmission of location information.

##### 4.1.1 Querying Protocols

A protocol is called a querying protocol, if the server decides when to request the location information from the source. In this case, the source can be very simple, as it does not need to keep extensive information about its state or realize a complicated logic. This may be important for small mobile devices. The simple and the cached querying protocols transmit the location information *on-demand*, that is only when it is queried by an application. If the location information is queried only seldomly, these protocols are more efficient than the reporting protocols described in 4.1.2, because the information is not transmitted unnecessarily (see Section 5).

*Simple:* In the most simple form the server requests the location information from the source each time it is queried by an application and therefore does not need a secondary copy at all. This leads to the highest possible accuracy of the location information, but also to a large number of messages, if the information is queried very often. The response

time of the server is also comparatively large, as the server has to contact the source for each query. On the other hand, this protocol has to be used, if because of privacy concerns the user does not allow his/her location information to be stored on any location server other than on his/her personal device. In this case, the personal device has to be accessed for each query to check for access rights. An example for this is the location service presented in [ST93], where a special program, a so-called User Agent, stores and protects the location information of a certain user. Here, we use this protocol mainly for comparison.

*Cached:* The cached querying protocol is an optimization of the simple protocol, where the server stores a cached copy of the last transmitted location information. When the location information for this mobile object is queried by an application, the server estimates the accuracy of the cached copy and returns it, if it is considered to be accurate enough. Otherwise, the server has to send a request to the source like in the simple protocol. A *pessimistic cached querying* protocol uses the distance, a mobile object may have traveled at its maximum speed, for the estimation of the accuracy. If the maximum speed of the mobile object is much higher than its average speed, the cached copy is often unnecessarily considered not accurate enough. An *optimistic cached querying* approach could use the average speed of the mobile object instead of the maximum one. However, with any optimistic estimation for the accuracy of the location information, the actual location of the mobile object can in the worst case differ from the reported location by more than the requested accuracy. With a cached querying protocol the response time of the server is varying and depends on whether the server can use the cached copy or has to contact the source.

*Periodic:* If the server queries the location information periodically from the source with a certain time interval  $D$ , the protocol is called a periodic querying protocol. Although here the initiative is reversed, this protocol has the same properties as the time-based variant of the reporting protocols described next.

##### 4.1.2 Reporting Protocols

In case of a reporting protocol the initiative is on the side of the source. It remembers which location information it has sent last to the server and therefore knows which location is stored there. An update of the location information is sent, whenever a comparison to the current position of the mobile object exceeds a certain distance or time threshold. The server can therefore conclude the maximum uncertainty of its copy from the value of this threshold. With an unmodified reporting protocol the server always answers the queries of the applications by directly returning the information currently stored in its copy. It can therefore only return the location information with an accuracy specified by the value of the threshold, even if a more accurate information is queried by an application. The response time of the server on the other hand is shorter, as it does not have to contact the source. A reporting protocol is usually more efficient for a given maximum accuracy than a querying protocol, if the location information is queried often (see Section 5) or if the server has to check periodically for the occurrence of events.

*Simple:* The most simple reporting protocol sends the location information each time the value of the primary copy

	Message rate is adjusted to mobility of object.	Message rate is adjusted to query rate and requested accuracy.	Upper bound for spatial uncertainty of returned information.	Allows applications to specify a desired accuracy.	Can detect disconnections.
querying: simple cached periodic		× ×	(×) (×)	× ×	× × ×
reporting: simple time-based distance-based dead reckoning	× ×		(×) × ×		(×) ×
combined	×	×	×	×	(×)

Table 1: Summary of properties for different update protocols.

changes, for example because a sensor system has determined a new location sighting. In this case the number of messages depends on the update rate of the sensor system and can be rather high. Depending on the type of sensor system, the simple reporting protocol also falls into one of the two following categories. We therefore do not consider it any further.

*Time-based:* With a time-based protocol the location information is transmitted periodically, after a certain interval of time  $T$  has elapsed. The update rate is fixed and does not depend on the behavior of the mobile object, which guarantees a certain temporal but not a spatial uncertainty of the information. If the object moves slowly or not at all, there is little or no difference in the location information transmitted by the messages to the server. If the object moves fast, not enough messages are sent to achieve a high accuracy.

*Distance-based:* The distance-based protocol sends an update of the location information whenever the geographic distance between the current location of the mobile object and the last reported location becomes greater than a given threshold  $D$ . As this protocol sends more messages if the mobile object is moving fast and less messages if it is slower or stationary, it is often more efficient for objects that perform sporadic movements between periods of immobility. This is, for example, typical for users in an office environment or for all sorts of office equipment. It is also possible to integrate the time-based and the distance-based protocol to combine their properties.

*Dead-reckoning:* Dead-reckoning is an optimization of the distance-based protocol. Here, the server estimates the current location of the mobile object based on its old location, its speed and the direction of its movement or on information about the route of the object. The source also calculates this estimated location and sends an update when it differs from the actual location by more than a certain distance threshold. A dead-reckoning approach performs very well, if the object is moving with constant speed in a given direction for some time or if the route of the mobile object is known in advance. In the second case the update costs can, according to [WSCY99], be reduced by 85% compared to other reporting protocols.

### 4.1.3 Combined Protocol

While a plain querying protocol can not be adjusted to different mobility characteristics of the mobile objects, a reporting protocol does not consider the query rate and the accuracy requested by the applications. With a *combined* protocol, which integrates the distance-based reporting protocol and the cached querying protocol, both of these features can be achieved. Similar to the distance-based reporting protocol, the source may update a secondary copy on the server to achieve a given spatial accuracy  $D$ . If the location information stored in the secondary copy is not accurate enough for a certain query, the server requests the information from the source as in a querying protocol. To minimize the total number of messages consisting of updates and queries, it has to be decided whether to update a secondary copy at all and what distance threshold  $D$  has to be used, depending on the mobility properties of the mobile object and on the queries by the applications (see Section 5). If the source monitors the mobility properties and the server those of the queries, they can adapt the properties of the protocol dynamically by increasing/decreasing the distance threshold  $D$  or by starting/stopping the updating of the secondary copy altogether. With the combined protocol, the response time is again varying and depends on whether the secondary copy on the server is accurate enough or if the server has to contact the source.

## 4.2 Behavior in Case of Disconnection

A problem that frequently occurs with mobile devices and wireless data transmission is a temporal disconnection of the communication link. A protocol which is intended to transmit location information from a mobile device to a server via a wireless communication link has therefore to be able to deal with such disconnections. In the following paragraphs we discuss the properties of the basic protocols with regard to disconnections, namely how long it takes to detect a disconnection and the maximum uncertainty of the location information returned during that time. If the server has detected a disconnection it can, according to the preferences of its users, either return an error message when queried for the location information or return the last known position together with an indication that it is obsolete. Where necessary we sketch appropriate modifications to these protocols that enable them to deal with disconnections.

*Querying protocols:* In case of a querying protocol, the disconnection is detected as soon as the server receives a query

and requests the location information from the source (and, in case of a cached querying protocol, also can not answer the query from its cached copy). Therefore, the server does not at any time return less accurate information than in the normal case.

*Time-based reporting protocol:* When using a time-based protocol a disconnection can be detected if no update messages have arrived after the time threshold  $T$  has elapsed since the last update. Again the server only returns location information with the specified accuracy.

*Distance-based reporting protocol:* With the basic distance-based protocol the server is not able to detect a disconnection. Instead, it assumes that the mobile object has not moved by more than the distance threshold  $D$  and returns the old location information as being up-to-date. To be able to detect disconnections, the distance-based protocol can be combined with a time-based one, by having it sent a location update at least every time interval  $T_{max}$ . A suitable value for  $T_{max}$  has to be found by considering the message overhead versus the maximum uncertainty, the user is willing to tolerate in case of an error.

*Dead-reckoning reporting protocol:* A basic dead-reckoning protocol has the same problems with disconnections as the distance-based protocol and the same mechanism can be used to deal with them. In [WSCY99] another technique for detecting disconnections has been introduced, where the distance threshold  $D$  is being continuously decreased until the source is forced to send an update message or it has become increasingly likely that a disconnection has occurred.

*Combined protocol:* How the combined protocol reacts to disconnections depends on how accurate a secondary copy is kept on the server. A disconnection is detected as soon as the server has to request the location information from the source. If this is not the case, the combined protocol behaves like the distance-based protocol and can return inaccurate location information. Again, a maximum time interval  $T_{max}$  between update messages should be specified.

A summary of the properties for the different update protocols discussed in this section is contained in Table 1. It shows, whether the message rate of a protocol depends on the mobility characteristics of the mobile objects and whether it depends on the queries of the applications. It also indicates, whether the protocol can guarantee an upper bound for the returned location information and if it allows the applications to request a certain accuracy of the information. The final column shows, whether the protocol is able to deal with disconnections.

## 5 Analytical Comparison of the Protocols

In this section, the effectiveness and efficiency of the update protocols are compared in more detail, by means of an analysis. First, the variables that appear in the following analysis are described and a general data model for the location information and its accuracy is defined. The analysis considers the number of messages required for updating the location information on the server as opposed to the minimum and average accuracy of the location information returned to an application. Corresponding formulas are shown first for the querying and reporting protocols, then for a combined protocol. Finally, we discuss the characteristic properties of

these protocols, based on the results of the analysis. Only the updates and queries for the location information of a single mobile object are considered, as the results can be easily applied to several objects. For reasons of simplicity we also do not consider the delays for the transmission of the messages, which would lead to a further (temporal) uncertainty of the location information. Because it is added uniformly to all messages, the delay does not substantially affect the comparison of the protocols.

The efficiency of an update protocol is here considered to be the average number of messages  $m$  that is transmitted per second between the source and the server. Messages can be location updates generated by the source as well as location requests from the server. The accuracy of the location information on the server, which describes the effectiveness of a protocol, is defined by the maximum and average deviation between the information returned as a result of a query and the actual position of the mobile object. Up to now, we have assumed that applications are only interested in the maximum uncertainty of the location information, for example if they want to be able to determine which room of a building a certain user is in. However, other types of applications may be more concerned with the average accuracy of the information. An example for this could be a map, where the current locations of a number of mobile objects is shown. In our analysis we have therefore considered both, the maximum and the average accuracy,  $d_{max}$  and  $d_{avg}$ .

The calculations for  $m$ ,  $d_{max}$  and  $d_{avg}$  are based on the behavior of the mobile object, defined by its average and maximum speed,  $v_{avg}$  and  $v_{max}$ , and on the uncertainty  $u_p$  with which the location information is available at the source. Moreover, the calculations depend on how frequently and to what accuracy the location information is queried by the applications, which is described by the average number of queries per second  $q$  and the average of the requested uncertainty  $u_q$ . The calculations are also affected by characteristic parameters of the different protocols, namely the maximum uncertainty required of the secondary copy  $u_s$ , the speed that is assumed for the mobile object by the cached querying protocol  $v_{asd}$ , and the time threshold  $T$  for a time-based or the distance threshold  $D$  for a distance-based reporting protocol. Table 2 shows a summary of these variables.

Some important restrictions between these variables are as follows: Because the information transmitted to the server can not be more accurate than the information of the source, the uncertainty of the secondary copy is always equal to or greater than the uncertainty of the primary copy,  $u_s \geq u_p$ . Also, it is not practical for an application to request more accurate location information than the one available at the source. Hence, we also assume that  $u_q \geq u_p$ . For the calculation of the average deviation we assume that the average uncertainty of the location information on the source is half of the maximum uncertainty. This may not be true for all sensor-systems (e.g., the average uncertainty is less than half of the maximum if the source is a GPS sensor), but does not have much influence on the outcome of the analysis. The restrictions described here apply to all formulas in this section and are not explicitly stated again.

### 5.1 Location and Uncertainty Model

For the following analysis we use a uniform model of the location information returned by the different positioning systems and of its accuracy.

The location information is supposed to be given by global geodetic coordinates, for example of the WGS84 stan-

$v_{max}$	Maximum speed of mobile object.
$v_{avg}$	Average speed of mobile object.
$q$	Average number of times per second, the location information of the mobile object is queried.
$u_q$	Average of requested accuracy for this location information.
$u_q^*$	Accuracy requested in one certain query.
$u_p$	Uncertainty of primary copy.
$T_p$	Update interval of primary copy.
$t_l$	Time at which a certain location sighting $l$ has been acquired.
$u_s$	Uncertainty of secondary copy.
$v_{asd}$	Speed that is assumed for estimating the accuracy of the location information.
$T$	Time threshold of the time-based reporting protocol.
$D$	Distance threshold of the distance-based reporting protocol.
$m$	Average number of messages transmitted per second between source and server.
$d_{max}$	Maximum deviation between the location information the server returns as result of a query and the actual position of the mobile object.
$d_{avg}$	Average deviation.

Table 2: Variables used in the analysis.

dard, where the distance between two locations can easily be calculated. Symbolic location information (e.g., the name of a room) returned by some of the positioning systems can easily be converted to geometric coordinates. The geometric coordinate for a region described by a certain symbolic name is given by the center of the region, while the spatial accuracy is the distance from the center to the farthest point in the region.

A location sighting can have a temporal as well as a spatial accuracy. The temporal accuracy is given by the time that has elapsed since the location sighting has been acquired, while the spatial accuracy is defined by the maximum distance between the position reported by the sighting and the actual position of the mobile object. For many location-aware applications the spatial accuracy of a sighting is more important, because the applications are concerned with the spatial relationship between (mobile) objects. The uncertainty  $u_l(t)$  of a certain location sighting  $l$  describes the spatial accuracy at a given time  $t \geq t_l$ .

At the time of the sighting the uncertainty is determined by the accuracy  $u_p$  of the sensor system. The uncertainty at a later time  $t$  can be estimated by the distance the mobile object may have traveled during the time  $t - t_l$ . If a maximum bound for the velocity of the mobile object ( $v_{max}$ ) exists, the maximum uncertainty of the location sighting can be calculated by adding the distance the object can have traveled to the uncertainty of the sensor system (see Figure 4). This is described by the following equation:

$$u_l(t) = u_p + v_{max}(t - t_l) \quad (1)$$

For example, the maximum velocity of a car can be set to 300 km/h (to be on the safe side) and the maximum velocity

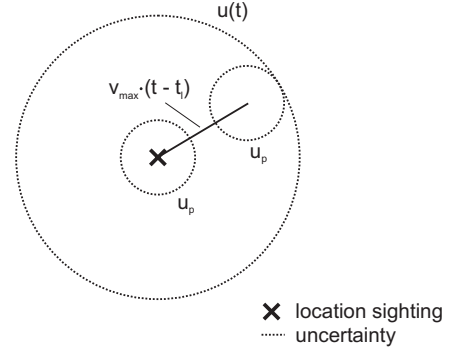


Figure 4: Uncertainty of location information depending on the accuracy of the sensor system and the elapsed time.

of a person to 36 km/h. By evaluating the history of the location sightings of a mobile object, it is also possible to determine a maximum velocity separately for each object.

In many cases, however, the location sighting will be more accurate than the uncertainty given by this equation, as the mobile object will not be moving at its maximum speed or not in a straight line. A person in an office will usually remain relatively stationary at a certain location (e.g., his office or a conference room) for a longer interval of time, while moving at a comparatively high speed between these locations. If the uncertainty does not need to be limited by an upper bound or the maximum velocity is not known, an assumed velocity  $v_{asd}$  (e.g., the average velocity  $v_{avg}$  of the mobile object) can be used to predict the uncertainty instead of the maximum velocity  $v_{max}$ . In some cases this will lead to an actual deviation, which is greater than the predicted one. This error will be worse for the sporadic movement of a person in an office, compared to the more steady movement of a traveling car.

## 5.2 Basic Protocols

The equations for an approximate calculation of the number of transmitted messages as well as for the maximum and average uncertainty of the location information are shown next for the basic querying and reporting protocols. We have not yet performed the analysis for the dead-reckoning protocol, because it depends to a great extent on the movement characteristics of the mobile object and we did not want to assume prior knowledge about the route of the object. In general the dead-reckoning protocol can be expected to behave similar to the distance-based reporting protocol.

*Simple querying protocol:* In the simple querying protocol, which is presented here only for reasons of comparison, the number of exchanged messages is equal to the number of queries as every query is forwarded to the source.

$$m = q \quad (2)$$

If we do not consider the communication delay, the location information presented to the applications has the same accuracy as the location information of the source.

$$d_{max} = u_p \quad (3)$$

$$d_{avg} = \frac{u_p}{2} \quad (4)$$

*Cached querying protocol:* The cached querying protocol uses Formula 1 to estimate the uncertainty of the location information stored on the server and if accurate enough returns it without contacting the source. For the speed of the mobile object used in this formula it can take a pessimistic approach with  $v_{max}$  (as shown in Formula 1), or an optimistic approach with  $v_{avg}$  instead of  $v_{max}$ . In the following considerations we represent the assumed speed of the mobile object by  $v_{asd}$ , which is then replaced by the speed used in a concrete version of the protocol. Compared to the simple protocol, the caching approach eliminates all messages where the query falls into the period of time in which the cached copy of the location information is considered accurate enough,  $\Delta t = (u_q - u_p)/v_{asd}$  (derived from Formula 1). The number of messages transmitted between source and server is therefore the total number of queries, multiplied by the fraction of queries that can not be answered from the cache, which is approximately  $1/(q(u_q - u_p)/v_{asd} + 1)$ .

$$m = \frac{q}{q \frac{u_q - u_p}{v_{asd}} + 1} \quad (5)$$

The maximum uncertainty of the returned location information is given by the time  $\Delta t$  the cached copy is considered accurate, multiplied with the maximum speed of the mobile object. To this the initial uncertainty of the location information of the primary copy has to be added. The average uncertainty can be approximated by the uncertainty of the sensor system, if the location information has to be requested from the source. Otherwise, the average uncertainty is half the distance that the mobile object can have traveled at average speed during the time in which a cached copy is considered valid. The corresponding Formula 7 is shown in a simplified form.

$$d_{max} = \frac{u_q - u_p}{v_{asd}} v_{max} + u_p \quad (6)$$

$$d_{avg} = \frac{\frac{u_q - u_p}{v_{asd}} \cdot \frac{v_{avg}}{2}}{\frac{v_{asd}}{(u_q - u_p)q} + 1} + \frac{u_p}{2} \quad (7)$$

If the assumed speed  $v_{asd}$  of the mobile object is set to its maximum speed  $v_{max}$ , the uncertainty of the location information presented to the querying application is limited by  $u_q$  (substituting the variables in Formula 6). The protocol can therefore in this case guarantee the accuracy requested by the applications.

*Time-based reporting protocol:* As the time-based protocol sends a location update periodically, the number of messages depends only on the time threshold  $T$ . To achieve a given uncertainty  $u_s$  for the secondary copy,  $T$  has to be set to the time, the mobile object needs to cover the distance given by  $u_s$  minus the uncertainty of the primary copy at maximum speed,  $T = (u_s - u_p)/v_{max}$ .

$$m = \frac{1}{T} \quad (8)$$

The maximum deviation can be calculated by adding the distance that the object may have traveled during time  $T$  to the uncertainty of the location information at the source. For the average deviation the average speed is used, instead of the maximum one.

$$d_{max} = T \cdot v_{max} + u_p \quad (9)$$

$$d_{avg} = \frac{T \cdot v_{avg}}{2} + \frac{u_p}{2} \quad (10)$$

*Distance-based reporting protocol:* The number of messages transmitted with the distance-based protocol is also independent of the number of queries. To guarantee a given uncertainty  $u_s$  of the location information on the server, the distance threshold  $D$  is set to  $u_s - u_p$ . The average number of messages per second can then be calculated by the time interval that the mobile object requires to cover this distance at average speed.

$$m = \frac{v_{avg}}{u_s - u_p} \quad (11)$$

If no messages are lost due to problems with the network connection (compare Section 4), the distance-based protocol achieves the requested maximum uncertainty  $u_s$  for the secondary copy on the server. On average the uncertainty is half of it.

$$d_{max} = u_s \quad (12)$$

$$d_{avg} = \frac{u_s}{2} \quad (13)$$

### 5.3 Combined Protocol

The combined protocol comprises elements of the distance-based reporting protocol and the pessimistic cached querying protocol, described in the previously. Therefore, it also combines elements of their behavior. The transmitted messages are the updates sent to keep the secondary copy up-to-date as well as the requests for the location information on the source sent by the server, if the secondary copy is not accurate enough. The number of updates can be calculated similar to the number of messages in the distance-based reporting protocol. The number of requests is the number of messages in the cached querying protocol, where the query rate is reduced by the probability that the uncertainty demanded in a query is lower than the accuracy of the location information already stored in the secondary copy,  $P(u_q^* < u_s)$ . The probability distribution depends on how the location information is queried and used by the applications.

$$m = \frac{v_{avg}}{u_s - u_p} + \frac{P(u_q^* < u_s) \cdot q}{P(u_q^* < u_s) \cdot q \frac{u_q - u_p}{v_{max}} + 1} \quad (14)$$

Because the combined protocol queries the source whenever the location information stored in the server is less accurate than the accuracy demanded in a query, it can always return the desired accuracy similar to a cached querying protocol. The maximum uncertainty returned is therefore the uncertainty demanded in the queries. The average uncertainty is again a combination of the average accuracy of the reporting protocol, in case the accuracy requested in a query can be met by the secondary copy, and of the average accuracy of the querying protocol, otherwise.

$$d_{max} = u_q \quad (15)$$

$$d_{avg} = P(u_q^* \geq u_s) \cdot \frac{u_s}{2} + P(u_q^* < u_s) \cdot \left( \frac{\frac{u_q - u_p}{v_{max}} \cdot \frac{v_{avg}}{2}}{\frac{v_{max}}{(u_q - u_p)q} + 1} + \frac{u_p}{2} \right) \quad (16)$$

The behavior of the combined protocol can be controlled through the uncertainty of the secondary copy. If it is set to a low value, the protocol behaves like the distance-based reporting protocol, if it is set to infinity, the combined protocol can be made to behave like a cached querying protocol.



## 5.4 Discussion

To compare the properties of the protocols, we first look at their efficiency, that is the number of messages that are transmitted between source and server. These are shown in Figure 5 for the querying and reporting protocols depending on the number of queries per second. For the other parameters we have assumed values taken from GPS traces that we have obtained from a Differential GPS sensor during a car ride and which are described in more detail in the next section. The average and maximum speed for the mobile object is  $v_{avg} = 12$  m/s and  $v_{max} = 40$  m/s. The uncertainty of the primary copy  $u_p$  is that of the sensor system and is set to 5 m. For the queries we have assumed a fixed demanded uncertainty of  $u_q = 100$  m.

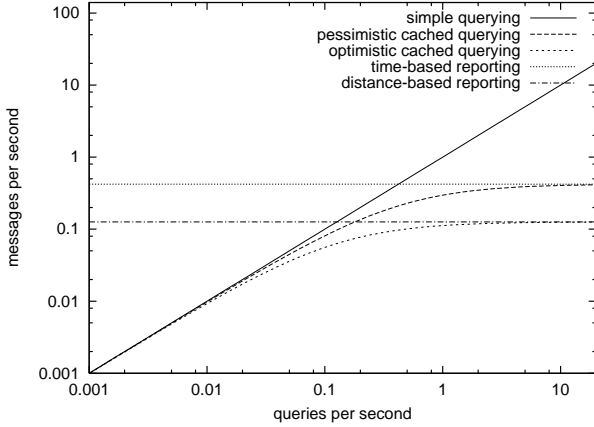


Figure 5: Analytical results: A comparison of the number of messages transmitted with the querying and reporting protocols for a given query rate.

Generally, a distance-based reporting protocol performs better than the time-based one and the optimistic cached querying protocol better than the pessimistic one. While the properties of the reporting protocols are independent from the query rate, the number of messages increases with a querying protocol with the number of queries. If the query rate is low, a querying protocol is therefore better than a reporting protocol. For very low query rates a simple protocol is sufficient. If the query rate is higher, the distance-based reporting protocol requires fewer messages than the pessimistic cached querying protocol, which is in many cases to be preferred to the optimistic protocol as it can guarantee a maximum uncertainty of the returned location information. The query rate at which the performance of the reporting protocol becomes better than that of the querying protocol is given by the following inequation:

$$q > \frac{1}{\frac{u_s - u_p}{v_{avg}} - \frac{u_q - u_p}{v_{max}}} \quad (17)$$

If  $v_{avg}$  is low compared to  $v_{max}$ , which is the case for mobile objects with sporadic movements, the limit for the query rate in Formula 17 approaches 0. For such objects, the distance-based reporting protocol is always better than the cached querying one. If  $v_{avg}$  approaches  $v_{max}$ , the limit for the query rate approaches infinity. In this case, the cached querying protocol always performs better. However, the number of messages for high query rates is only a little higher with the reporting than with the querying protocol.

The other aspect of the update protocols, which is discussed in this paper, is their effectiveness, that is to what degree the server can fulfill the accuracy demanded in a query. The maximum and average uncertainty ( $d_{max}$  and  $d_{avg}$ ) of the location information returned by the server is shown separately for the two cached querying and the two reporting protocols in Figure 6. The uncertainty is shown depending on the *speed ratio* between the average and the maximum speed of the mobile object ( $v_{avg}/v_{max}$ ), which gives a good indication of its mobility characteristics. A mobile object with a low ratio moves sporadic, an object with a ratio approaching 1 moves steadily. For the graphs we have assumed a query rate  $q$  of 0.1 queries per second.

The pessimistic cached querying approach can meet the required accuracy independently from the mobility of the object. For a low speed ratio it also has a very low average uncertainty. In comparison, the optimistic cached querying protocol, which is much more efficient, results in an almost constant average uncertainty, a little below the uncertainty requested by the client. However, for a low speed ratio the maximum uncertainty can be much higher than the one demanded by the client. This protocol is therefore not suitable for objects with sporadic movement, if the querying application can not tolerate variations in the uncertainty.

Both reporting protocols can meet a fixed uncertainty demanded by the client. The time-based protocol has a low average uncertainty for a small speed ratio, compared to the distance-based protocol, which offers a constant average uncertainty for all types of mobile objects. As the former requires more messages, the later is to be preferred. The reporting protocols, however, can only return the location information with the accuracy currently stored on the server. They can not meet the demand of applications which query the information with a lower requested accuracy,  $u_q^* < u_s$ .

From the viewpoint of their effectiveness, the distance-based reporting protocol is equal to the cached querying protocol if only a fixed requested accuracy is considered (although, the later has a better average uncertainty, especially for a low speed rate). The reporting protocols are not suitable, if the location server allows the client to flexibly specify the requested accuracy and it is important to meet these requirements.

As mentioned before, the performance of the combined protocol depends on the uncertainty with which the secondary copy is updated as well as on the query rate. In Figure 7, the number of transmitted messages is shown for the combined protocol depending on the uncertainty of the secondary copy and the query rate. For the uncertainty demanded in the queries we have assumed a Gaussian distribution with an expected value of  $u_q = 100$  m and a standard deviation of 20 m. All other parameters have the same values as before.

Figure 7 indicates an optimal value for the uncertainty of the secondary copy for higher query rates, whereas for lower query rates the number of messages is lowest, if the uncertainty is set to infinity. It is therefore possible to use Formula 14 to decide, whether it is useful to update a secondary copy at all and to find an optimal uncertainty value for it. The uncertainty value can either be calculated with the necessary parameters set to the average values determined for a certain environment, the protocol is going to be used in. Or it can be adjusted dynamically according to changes in the environment.

For comparison with the other protocols, Figure 8 shows the number of messages transmitted with the combined protocol depending on the query rate. Optimal values for the uncertainty of the secondary copy have been taken from Fig-

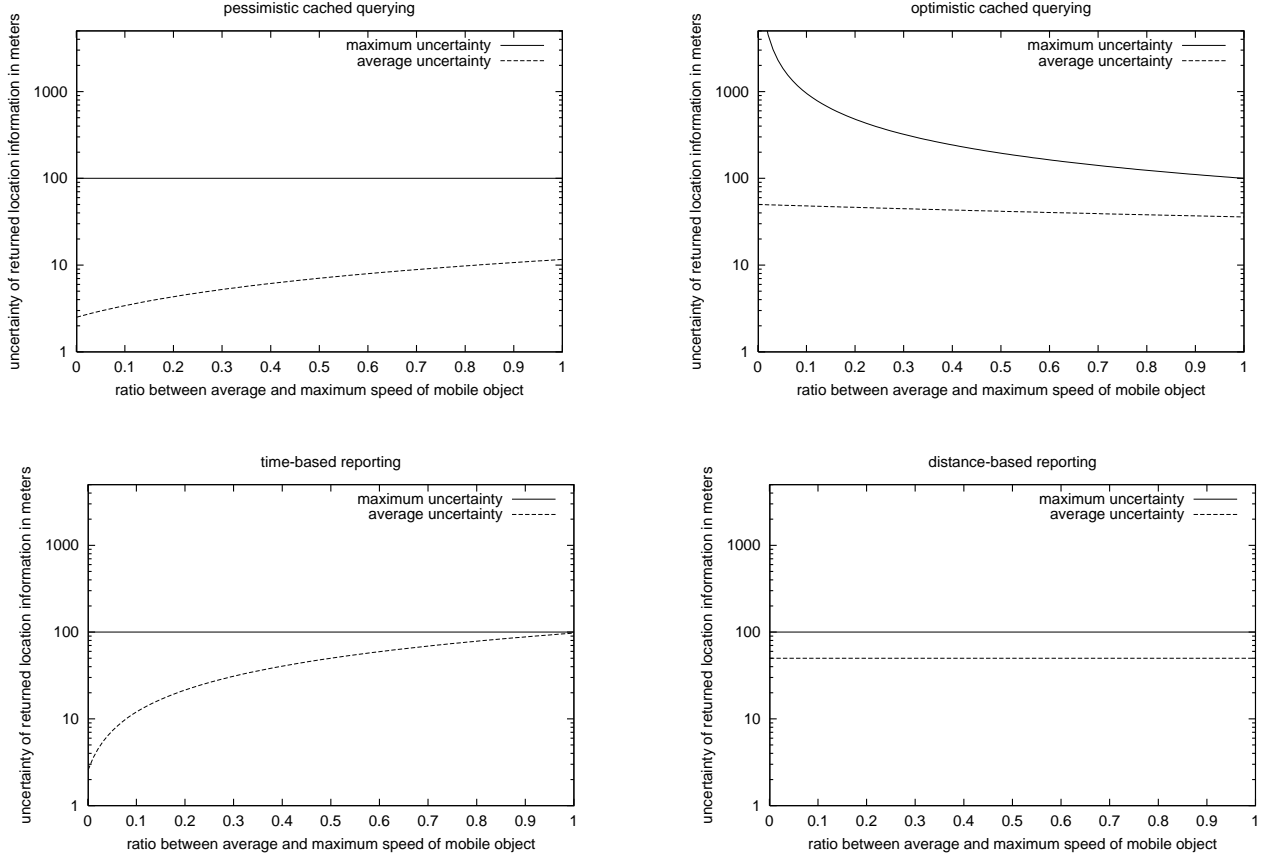


Figure 6: Analytical results: Maximum and average uncertainty of the location information on the server for the optimistic and pessimistic cached querying protocols and the time-based and distance-based reporting protocols.

ure 7.

The combined protocol integrates the positive features of the more simple querying and reporting protocols. Although, for a higher query rate it is less efficient than the distance-based reporting protocol, it requires much less messages for a low query rate, similar to the querying protocols. Unlike the reporting protocols, it is also able to meet an arbitrary requested accuracy. Its disadvantages are that it is more complicated to implement, because an optimal value for the uncertainty of the secondary has to be found, and that communication errors are more difficult to handle.

In summary, the combined protocol is suitable for most cases, especially if a protocol is needed that performs equally well in different environments or even can adapt to them. Of the querying and reporting protocols the pessimistic cached querying protocol and the distance-based protocol are to be preferred, because they combine an upper bound for the spatial accuracy of the returned location information with a good efficiency. Of these two, the former is decidedly better for objects with sporadic movements and for higher query rates, but does not allow the accuracy of the location information to be requested on application-level.

## 6 Simulation Results

For the update protocols discussed in Section 5, we have performed various simulation runs based on ten actual GPS traces. To this end we have developed a simple simulator, which implements a location source with a primary copy

and a location service with a secondary copy for the location information of one mobile object. The information on the source is updated from the GPS trace, while the server receives randomly created location queries with an exponentially distributed time interval and a fixed or, in case of the combined protocol, a Gaussian distributed accuracy. The mean value for the exponential distribution of the time interval between queries is the inverse of the query rate  $q$ . The fixed accuracy requested in the queries is 100 m for the basic protocols. The Gaussian distributed accuracy for the combined protocol has a mean value of 100 m and a standard deviation of 20 m. Into this simulation environment different update protocols can be inserted that specify, how the source and server react to incoming updates or queries and how the location information is transmitted. As in the analysis, the delays for transmitting the messages are not considered.

The ten GPS traces used in our simulations have been acquired on a typical car ride in the commuting traffic around the city of Stuttgart. These traces have an average length of about 28 minutes and an average speed of 44.68 km/h. A Differential GPS sensor with an accuracy between 2 and 5 meters has been used to determine the location information, which is written to a file every second.

For each trace we have performed various simulation runs, where the number of transmitted messages has been calculated depending on a given query rate. Figure 9 shows the results of these simulation runs as the mean values of 100 separate simulations. The parameter values for the protocols (e.g., the average and maximum speed) have been taken

speed ratio	pessimistic cached querying		optimistic cached querying		distance-based reporting		time-based reporting		combined	
	max.	avg.	max.	avg.	max.	avg.	max.	avg.	max.	avg.
0.260	74.167	5.922	335.798	22.725	99.817	49.664	99.998	16.971	99.926	6.131
0.291	88.299	6.507	287.795	23.286	99.964	49.005	88.299	17.067	99.506	6.604
0.302	56.288	6.138	294.754	19.696	99.982	50.597	100.078	13.096	101.970	6.208
0.352	104.133	6.761	253.763	20.917	99.899	49.091	100.233	19.121	88.127	7.034
0.354	73.911	6.998	263.790	21.720	99.887	45.730	75.895	17.598	104.243	7.238

Table 3: Simulation results: Maximum and average accuracy of the location information for the different update protocols.

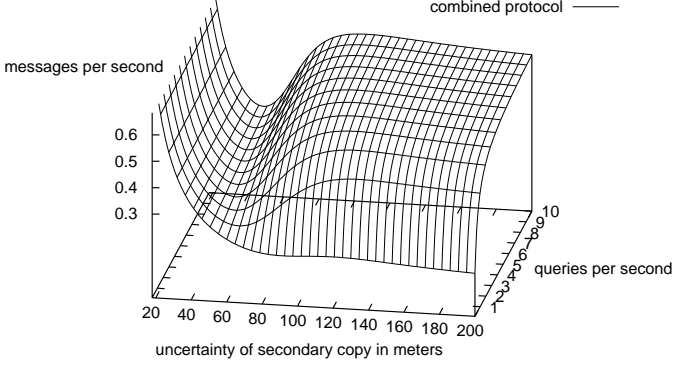


Figure 7: Analytical results: Number of messages transmitted with the combined protocol depending on the number of queries and the uncertainty of the secondary copy on the server.

from the respective trace.

In general, these simulation results confirm the assumptions we have made for the analysis. The number of messages for the cached querying protocols is a little higher than in the analysis. One reason for this is that the location information queried at the source is not determined exactly when queried, as we have assumed there. Instead, it has already an average age of  $T_p/2$  where  $T_p$  is the update interval of the sensor system, in our case 1 second. Therefore, the location information on the server has to be updated a little sooner than calculated. Also, the maximum velocity of the mobile object is sometimes difficult to obtain from the GPS traces, because the uncertainty of the positioning systems can have a great influence on the distance of two neighboring entries. We therefore take the average of 5 seconds to determine the maximum speed. It happens a number of times that the flow of location information from the DGPS sensor is interrupted for a short time, which may also lead to variances in the behavior of the protocols.

For the combined protocol, we have again determined the message count depending on the uncertainty value for the secondary copy as well as on the query rate. The results of this simulation are shown in Figure 10. Although the results are again a little higher than the values determined in our analysis, the analysis reflects their characteristics nicely.

For five GPS traces with different speed ratios ranging from 0.26 to 0.354 the maximum and average resulting accuracies are depicted in Table 3. The query rate has been set to 0.1, as the results tend to vary more for higher rates, while the other parameters have the same values as before. For the accuracies, as compared to the number of transmitted messages, the results depend more on the characteristics

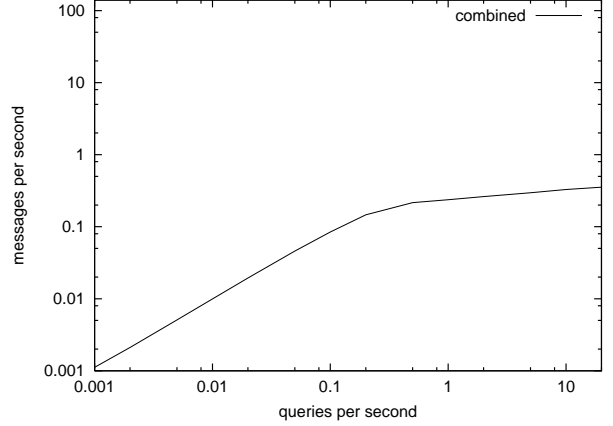


Figure 8: Analytical results: Number of messages transmitted with the combined protocol depending on the number of queries. For the uncertainty of the secondary query the optimal values taken from Figure 7 are used.

of a certain trace. Because we have averaged the maximum speed over 5 seconds, the apparent speed of two neighboring entries in the trace may be higher than the one we have assumed for the parameters of our protocols. This can lead to a maximum accuracy that is greater than the requested one (100 m), when using a cached querying or the combined protocol.

## 7 Conclusion and Future Work

In this paper, we have described different protocols for transmitting location information and have compared them according to their effectiveness and efficiency. These protocols may be used, wherever location information is queried or updated continuously. With the help of the analysis a suitable protocol can be found for a given environment. Furthermore, the analysis gives an estimation for the expected network load and the average uncertainty of the transmitted location information, as well as an upper bound for its maximum uncertainty. Based on the querying and reporting protocols, we have proposed a combined protocol that integrates most of the advantageous properties of the basic protocols. Here, the analysis can give optimal parameter settings for a certain environment.

For our future work, we plan to run further simulations with different types of mobile objects, for example a person while window-shopping in a city or riding a bicycle, to look at various movement characteristics. The results will then be used, to further improve the analysis, where necessary. We also plan to extend the combined protocol by adding a mechanism that allows it to adapt to changes in

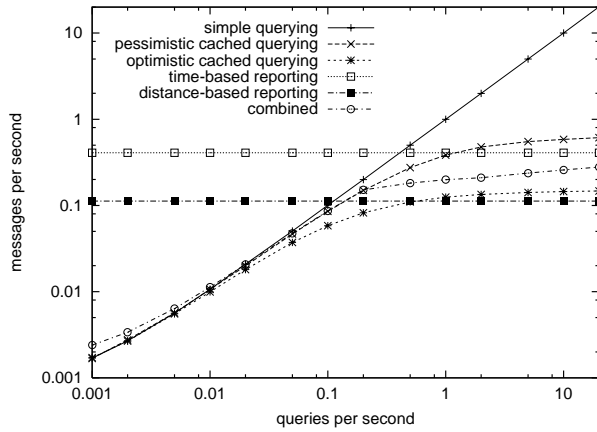


Figure 9: Simulation results: Number of messages transmitted with the querying, reporting and combined protocols determined by simulation.

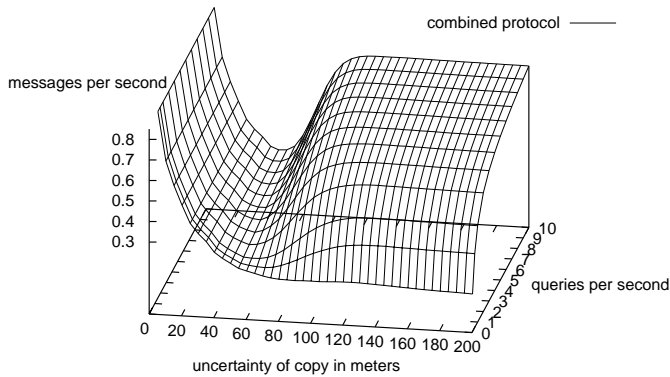


Figure 10: Simulation results: Number of messages transmitted with the combined protocol determined by simulation depending on uncertainty of secondary copy as well as on the query rate.

the environment, dynamically. This mechanism will have to monitor critical parameters (e.g., the average speed of the mobile object) and to adjust the uncertainty of the secondary copy or to stop updating it at all, accordingly. Such a mechanism can increase the efficiency of the protocol in a dynamic environment. It can also be the basic mechanism for an adaptive location service, which is able to adjust its whole architecture.

Because we wanted to look at the protocols from a general viewpoint, we have not examined the dead reckoning protocols in more detail, as they work best, when they have some knowledge about the future movement of their users. Nevertheless, they can increase the efficiency of a location service to a great deal. The prediction of a mobile object's future movements can be based on four possible sources: Without much further effort, a simple protocol could use the current speed and direction of the mobile object to predict its future movements. Also, information about the route of the mobile object can come from the user or an application program, for example a car navigation system (compare [WSCY99]). In a more flexible solution, the route can be predicted based on the former movements of a mobile

object, as for example a person will mostly use the same route when commuting to his/her office (compare [BD99]). Finally, a map-based approach can find the street a user is moving on and predict his future movements according to the course of the street, possibly also considering the usual movement pattern of mobile objects on this street. We plan to integrate a dead-reckoning protocol into our comparison of update protocols and to implement a map-based protocol for the planned location service.

Our main goal is the development of a universal distributed location service, as it has been described shortly in Section 3. We have already developed a prototype for a centralized service, and are currently looking at and comparing different architectures for distributing this service efficiently. Future work includes aspects of efficiently managing the location data and of creating a security mechanism, which allows a user to control in detail who and to what degree has access to the information about his location.

## References

- [BD99] A. Bhattacharya and S. Das. LeZi-update: An information-theoretic approach to track mobile users in PCS networks. In *Proceedings of the Fifth Annual ACM/IEEE International Conference on Mobile Computing and Networking*, pages 1–12, Seattle, WA, USA, 1999. ACM Press.
- [BNKS95] A. Bar-Noy, I. Kessler, and M. Sidi. Mobile users: To update or not to update? *Wireless Networks*, 1(2):175–185, July 1995.
- [Com97] IEEE Computer Society LAN MAN Standards Committee. *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification*. The Institute of Electrical and Electronics Engineers, New York, 1997. IEEE Std. 802.11.
- [HH94] A. Harter and A. Hopper. A distributed location system for the active office. *IEEE-Network*, 8(1):62–70, 1994.
- [HWLC97] B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins. *Global Positioning System: Theory and Practice*. Springer-Verlag TELOS, 1997.
- [IA97] National Imagery and Mapping Agency. DoD World Geodetic System 1984, its definition and relationship with local geodetic systems. Technical Report 8350.2, Third Edition, National Imagery and Mapping Agency, 1997.
- [Leo98] U. Leonhardt. *Supporting Location-Awareness in Open Distributed Systems*. PhD thesis, Imperial College of Science, Technology and Medicine, University of London, 1998.
- [LK99] A. Leonhardt and U. Kubach. An architecture for a universal, distributed location service. In *Proceedings of the European Wireless '99 Conference*, pages 351–355, Munich, Germany, 1999. VDE Verlag.
- [Maa97] H. Maaß. Location-aware mobile applications based on directory services. In *Proceedings of The Third International Conference on Mobile Computing and Networking (MobiCom '97)*,

- pages 23–33, Budapest, Hungary, 1997. ACM Press.
- [MP92] M. Mouly and M.-B. Pautet. *The GSM System for Mobile Communications*. Cell & Sys, France, 1992.
  - [Pe96] C. Perkins (editor). IP mobility support. RFC 2002, IETF, 1996.
  - [Sat96] M. Satyanarayanan. Fundamental challenges in mobile computing. In *Proceedings of the Fifteenth ACM Symposium on Principles of Distributed Computing*, pages 1–7, Philadelphia, PA, USA, 1996. ACM Press.
  - [ST93] M. Spreitzer and M. Theimer. Providing location information in a ubiquitous computing environment. In *Proceedings of the Fourteenth ACM Symposium on Operating System Principles*, pages 270–283. ACM Press, 1993.
  - [ST94] B. Schilit and M. Theimer. Disseminating active map information to mobile hosts. *IEEE Network*, pages 22–32, 1994.
  - [WHFG92] R. Want, A. Hopper, V. Falcao, and J. Gibbons. The active badge location system. *ACM Transactions on Information Systems*, 10(1):91–102, 1992.
  - [WSCY99] O. Wolfson, A. P. Sistla, S. Chamberlain, and Y. Yesha. Updating and querying databases that track mobile units. *Distributed and Parallel Databases Journal*, 7(3):1–31, 1999.