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

Abstract. 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. Therefore, special update protocols for location information are required that transmit the information 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 in comparison with the number of messages transmitted. We also present the results of simulations that we have performed to back up our analysis.

Keywords: location-awareness, location service, location update protocols

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 *location-aware* applications are using this information. Examples are car navigation or fleet management systems, as well as location-aware information services for cellular phones. Most of these applications only consider the location of a single user that can be determined through a local positioning sensor. 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 [7] and [9]. 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 move only rarely, like pieces of office equipment, for example a printer. Also, the location information for these objects may have been acquired by different types of sensor systems and therefore may have 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. Two different classes of such update protocols are *querying protocols* and *reporting protocols*. In case of a querying protocol the information is pulled by the receiver, while with a reporting protocol it is pushed by the sender.

In this paper we will discuss the characteristics of these classes of update protocols and their subclasses as well as their strengths and weaknesses with respect to different areas of application. In order to be able to guarantee a certain accuracy of the returned location information, it usually has to be transmitted rather frequently. Also, 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, still achieves the 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 as the resulting minimum and average accuracy of the location information at the receiver. We also introduce a combined protocols, where an optimal ratio between the number of updates and the number of queries can be found and which is able to dynamically adapt to different types of environments.

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 contains an analysis of the efficiency and effectiveness of the protocols. In section 6 we present simulation results to support the outcomes of our analysis.

2. Related work

The first systems providing location-awareness have been based on the Active Badge in-door locating system [14] or the similar ParcTab system [13]. In this context, a location service has been developed that manages the location information for an installation of the Active Badge system or its

Published in Cluster Computing, Volume 4, Number 4 / Oktober 2001, pp. 355-367, 2001.

much more accurate successor, Active Bat [4], using a simple update protocol. Leonhardt [7] proposes a wide-scale distributed location service that is independent of applications and sensor systems. In his Ph.D. thesis, he examines fundamental issues of a location service and classifies them in an abstract service model. In addition, he proposes a location model for the integration of different types of sensor data and requirements as well as policies for access control of a location service. The thesis also discusses architectural aspects of a global, general-purpose location service but does not propose or evaluate a specific architecture or update protocol in detail.

The management of the location information of 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 and Home Location Registers. Various reporting protocols have been discussed with the goal of minimizing the overall number of paging and update messages. In [2] a distance-based, a movement-based and a time-based reporting protocol are compared. Here, the location information is a discrete identifier, describing a certain cell of the communication network. Update policies for location management have been optimized for these requirements. In [1], for example, 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, for which we intend to use these update protocols, also allows further 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 [15]). One intended area of application is fleet management, where the trucks of a transportation company are being tracked. 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 specialized 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.

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 briefly describe the architecture for a large-scale location service, which is being developed at our Institute and where the results of this work will be applied.

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 external 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 (for details, see [7]). A typical tracking-system is the Active Badge system (see [14]), 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. An example for a positioning system is the widely used global satellite positioning system GPS (see [5]). 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 [6]). Because no positioning system offers a global coverage (GPS, for example, works only outdoors) a universal location service has to be able to integrate the location information acquired by different 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 [11]). 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 [10], a MAN, like the Metricom Ricochet network, or a LAN, for example according to the IEEE 802.11 standard [3]. 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 (see [12]).

3.3. A universal location service

The comparison of update protocols in this paper has been done as a part of our efforts to design and develop a large-scale distributed location service. A mobile object can be registered at this service, which then manages its position information with the accuracy specified by the registration

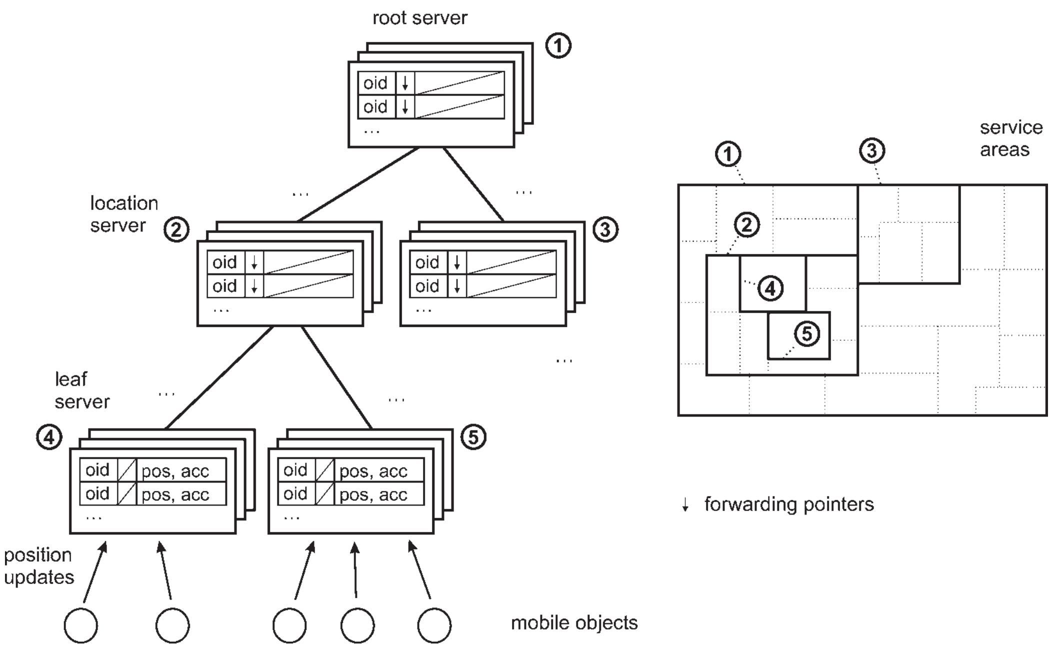


Figure 1. Basic architecture of the location service.

procedure (*register*). To this end, the information has to be updated by the mobile object itself or an external tracking system according to a suitable protocol (*position update*), which are the focus of this paper. Clients of the service may query the current location of a mobile object (*position query*) or all objects that are currently inside a certain geographic area (*range query*). Another type of query returns the mobile object(s) nearest to a certain location (*nearest neighbor query*).

To ensure its scalability, the location service is structured in a hierarchical fashion (see figure 1). Each leaf node of this hierarchy is responsible for a certain geographic area – its service area – and manages the position information for the mobile objects located in it. The position information on a leaf server is updated by the mobile objects or sensor systems inside its service area, using the protocols that are discussed in this paper. If a mobile object moves out of the service area of one leaf server into that of another one, a handover is performed between them. A position query can be issued to any leaf server, which then finds the server with the appropriate information following the forwarding pointers that are maintained by the higher-level servers. Range queries are performed in a similar way.

In [8] the architecture and protocols of the location service are presented in detail. Our experiments with a first prototype show that the location service, using a main memory DB for the position information and UDP as communication mechanism, can cope with a large number of objects and updates.

4. Update protocols

In this section the different classes of update protocols are introduced and their main properties are discussed. The protocols are assumed 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. We call the component that manages the primary copy, which may be a sensor system or another location server, the *source* of the location information. The secondary copy is usually stored on a 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.

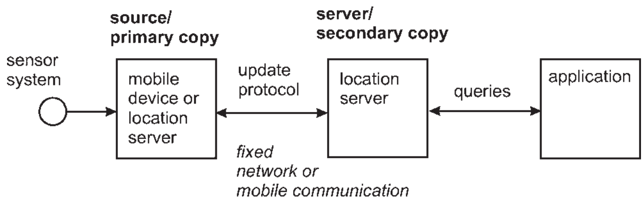


Figure 2. Components involved in the updating of location information.

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. 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 [2]. True querying protocols cannot be used in PCS as they would involve the paging of all cells. The classes of update protocols and their relationships are shown in more detail in figure 3.

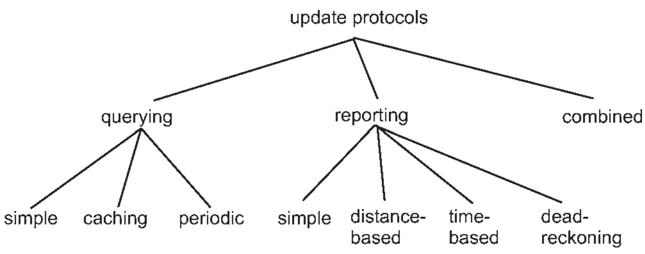


Figure 3. Classification of update protocols for the transmission of location information.

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 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 rarely, these protocols are more efficient than the reporting protocols described in section 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. 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 similar as with the simple protocol. A *pessimistic cached querying* protocol uses the distance a mobile object may have traveled at its maximum speed for its 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 therefore 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 may in some cases differ from the reported location by more than the requested accuracy. With a cached querying protocol the response time of the server will vary 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 with 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 data of the mobile object shows that a certain distance or time threshold has been exceeded. If there are no transmission errors, the server can therefore infer 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 more accurate information is requested by an application. The response time of the server on the other hand is rather short, 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 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 is usually very high. We therefore consider it here only for comparison.

Time-based. With a time-based protocol the location information is transmitted periodically, each time 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 maximum temporal but not a maximum 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 the desired 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 moves slowly or is stationary, it is more efficient for objects that perform sporadic movements between periods of immobility. This is, for example, typical for users in an office environment. 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 or if the route of the mobile object is known in advance. In the second case the update costs can, according to [15], be reduced by up to 85%.

4.1.3. Combined protocol

While a plain querying protocol cannot be adjusted to the 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 that 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 sent 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. The decision whether to use the secondary copy or to query the source or whether to change the distance threshold needs only to be performed when an update or query arrives. The combined protocol therefore adds some overhead to query and update processing but requires no additional timer(s). 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. 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 cannot 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 distance-based protocol alone 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 send 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 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.

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.

Table 1
Summary of properties for different update protocols.

	Message rate depending on mobility	Message rate depending on queries	Upper bound for uncertainty of results	Applications may specify desired accuracy	Can detect disconnections
Querying:					
Simple		×	(×)	×	×
Cached		×	(×)	×	×
Periodic					×
Reporting:					
Simple			(×)		(×)
Time-based					×
Distance-based	×		×		
Dead reckoning	×		×		
Combined	×	×	×	×	(×)

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 are used in the 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 equations 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. For reasons of simplicity we do not consider transmission delays for 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.

Here, the efficiency of an update protocol is 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 distance between the position 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.

Table 2
Variables used in the analysis.

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

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, $u_{\text{s}} \geqslant u_{\text{p}}$. 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 $u_{\text{q}} \geqslant u_{\text{p}}$. 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. The restrictions described here apply to all equations in this section and will not be 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 standard, 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 (see [7]). To this end, 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 $t \geqslant t_l$.

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:

$$u_l(t) = u_\mathrm{p} + v_\mathrm{max}(t - t_l). \quad (1)$$

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.

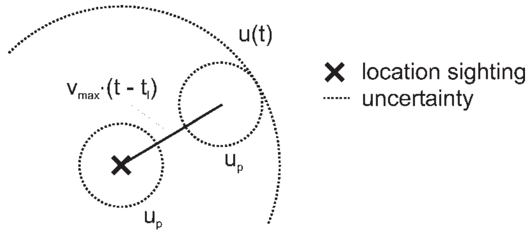


Figure 4. Uncertainty of location information depending on the accuracy of the sensor system and the elapsed time.

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 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 similarly 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

$$d_\mathrm{max} = u_\mathrm{p}, \qquad d_\mathrm{avg} = \frac{u_\mathrm{p}}{2}. \quad (3)$$

Cached querying protocol. The cached querying protocol uses equation (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 assumed speed of the mobile object used in this equation, it can take a pessimistic approach with v_max (as shown in equation (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, $\Delta t = (u_\mathrm{q} - u_\mathrm{p})/v_\mathrm{asd}$ (derived from equation (1)). The number of messages transmitted between source and server is therefore the total number of queries, multiplied by the fraction of queries that cannot be answered from the cache, which is approximately $1/(q(u_\mathrm{q} - u_\mathrm{p})/v_\mathrm{asd} + 1)$,

$$m = \frac{q}{q(u_\mathrm{q} - u_\mathrm{p})/v_\mathrm{asd} + 1}. \quad (4)$$

The maximum uncertainty of the returned location information is given by the time Δt the cached copy is considered accurate, multiplied by 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 equation (5) is shown in a simplified form

$$\begin{aligned} d_{\mathrm{max}} &= \frac{u_{\mathrm{q}} - u_{\mathrm{p}}}{v_{\mathrm{asd}}} v_{\mathrm{max}} + u_{\mathrm{p}}, \\ d_{\mathrm{avg}} &= \frac{((u_{\mathrm{q}} - u_{\mathrm{p}})/v_{\mathrm{asd}}) \cdot (v_{\mathrm{avg}}/2)}{v_{\mathrm{asd}}/((u_{\mathrm{q}} - u_{\mathrm{p}})q) + 1} + \frac{u_{\mathrm{p}}}{2}. \end{aligned} \tag{5}$$

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 equation (5)). 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 which the mobile object needs to cover the distance given by u_{s} (minus the uncertainty of the primary copy) at maximum speed, $T = (u_{\mathrm{s}} - u_{\mathrm{p}})/v_{\mathrm{max}}$,

$$m = \frac{1}{T}. \tag{6}$$

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, half of the average speed is used, instead of the maximum one as well as half of the uncertainty at the primary copy

$$\begin{aligned} d_{\mathrm{max}} &= T \cdot v_{\mathrm{max}} + u_{\mathrm{p}}, \\ d_{\mathrm{avg}} &= \frac{T \cdot v_{\mathrm{avg}}}{2} + \frac{u_{\mathrm{p}}}{2}. \end{aligned} \tag{7}$$

Distance-based reporting protocol. The number of messages transmitted using the distance-based protocol is also independent of the number of queries. To guarantee a given maximum uncertainty u_{s} of the location information on the server, the distance threshold D is set to $u_{\mathrm{s}} - u_{\mathrm{p}}$. 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

$$m = \frac{v_{\mathrm{avg}}}{u_{\mathrm{s}} - u_{\mathrm{p}}}. \tag{8}$$

If no messages are lost due to problems with the network connection, 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

$$d_{\mathrm{max}} = u_{\mathrm{s}}, \qquad d_{\mathrm{avg}} = \frac{u_{\mathrm{s}}}{2}. \tag{9}$$

5.3. Combined protocol

The combined protocol comprises elements of the distance-based reporting protocol and the pessimistic cached querying protocol. 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 similarly 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, $P(u^*_{\mathrm{q}} < u_{\mathrm{s}})$. This probability distribution depends on how the location information is queried and used by the applications

$$m = \frac{v_{\mathrm{avg}}}{u_{\mathrm{s}} - u_{\mathrm{p}}} + \frac{P(u^*_{\mathrm{q}} < u_{\mathrm{s}}) \cdot q}{P(u^*_{\mathrm{q}} < u_{\mathrm{s}}) \cdot q(u_{\mathrm{q}} - u_{\mathrm{p}})/v_{\mathrm{max}} + 1}. \tag{10}$$

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

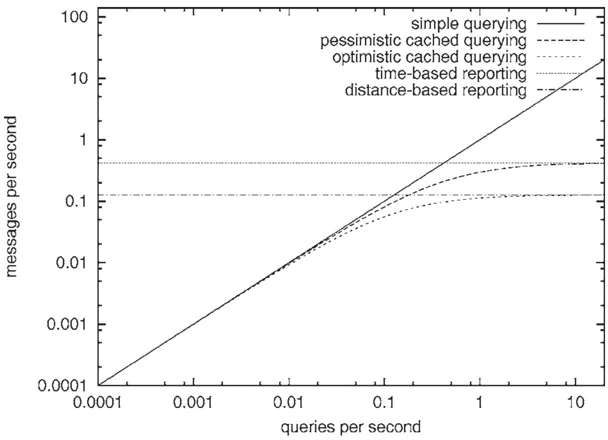
$$\begin{aligned} d_{\mathrm{max}} &= u_{\mathrm{q}}, \\ d_{\mathrm{avg}} &= P(u^*_{\mathrm{q}} \geqslant u_{\mathrm{s}}) \cdot \frac{u_{\mathrm{s}}}{2} + P(u^*_{\mathrm{q}} < u_{\mathrm{s}}) \\ &\quad \times \left(\frac{((u_{\mathrm{q}} - u_{\mathrm{p}})/v_{\mathrm{max}}) \cdot (v_{\mathrm{avg}}/2)}{v_{\mathrm{max}}/((u_{\mathrm{q}} - u_{\mathrm{p}})q) + 1} + \frac{u_{\mathrm{p}}}{2} \right). \end{aligned} \tag{11}$$

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.

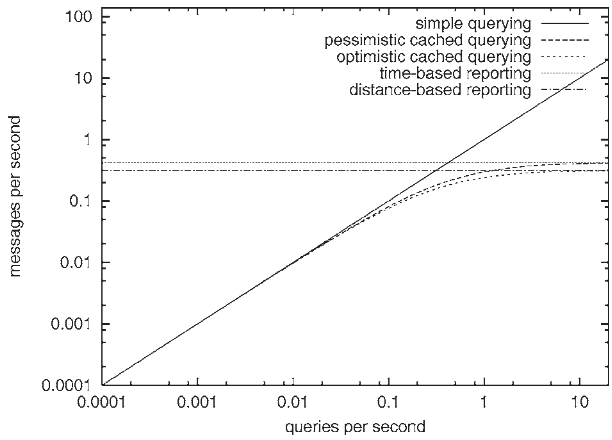
5.4. Discussion

In this subsection, we will use the results of the analysis for a comparison of the protocols. For the discussion we look at four different types of mobile objects, o_1 to o_4:

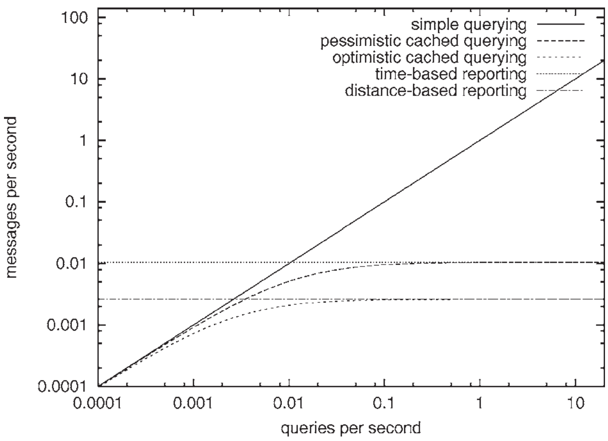
- o_1: A car in commuting traffic. This object is moving most of the time and its maximum and average speed are $v_{\mathrm{max}} = 40$ m/s and $v_{\mathrm{avg}} = 12$ m/s. These values are taken from GPS-traces, which are used in the next section for a simulation of the protocols.



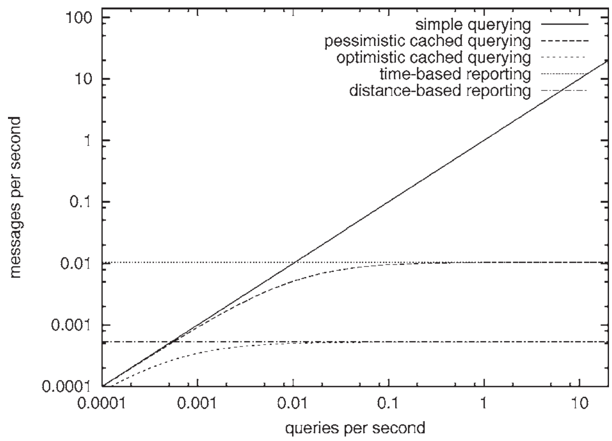
(a)



(b)



(c)



(d)

Figure 5. Analytical results: a comparison of the number of messages transmitted with the querying and reporting protocols for a given query rate. From upper left to lower right for objects o_1–o_4 (a–d).

- o_2: A car on a highway, which has the same maximum speed of $v_{\max} = 40$ m/s, but an average speed of $v_{\text{avg}} = 30$ m/s.
- o_3: A person while window-shopping. This object moves about half of its time with a maximum speed of $v_{\max} = 1$ m/s. As it does not always move with maximum speed, we assume an overall average speed of $v_{\text{avg}} = 0.25$ m/s.
- o_4: A person in an office environment. With a maximum speed of again $v_{\max} = 1$ m/s, this object moves only about 10% of its time. We therefore assume an overall average speed of $v_{\text{avg}} = 0.05$ m/s.

For the sensor system we assume in all cases a Differential GPS with an accuracy of 5 m ($u_{\text{p}} = 5$ m). The accuracy demanded in the queries is fixed to $u_{\text{q}} = 100$ m.

We first look at the efficiency of the protocols, 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.

Generally, a distance-based reporting protocol performs better than the time-based one and the optimistic cached querying protocol better than the pessimistic one. As can be seen in figure 5, the difference is more marked for a mobile object with a lower ratio of average to maximum speed, such as o_4. While the properties of the reporting protocols are independent from the query rate, for a querying protocol the number of messages increases with the number of queries. The query rate at which the performance of the distance-based reporting protocol becomes better than that of the pessimistic cached querying protocol, which is also able to guarantee a certain accuracy of the returned information, is given by the following inequation:

$$q > \frac{1}{(u_{\text{s}} - u_{\text{p}})/v_{\text{avg}} - (u_{\text{q}} - u_{\text{p}})/v_{\max}}. \tag{12}$$

If v_{avg} is low compared to $v_{\max}$, which is the case for mobile objects with sporadic movements such as o_4, the distance-based reporting protocol is better than the cached querying protocol most of the times.

The other important aspect of the update protocols is their respective 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 informa-

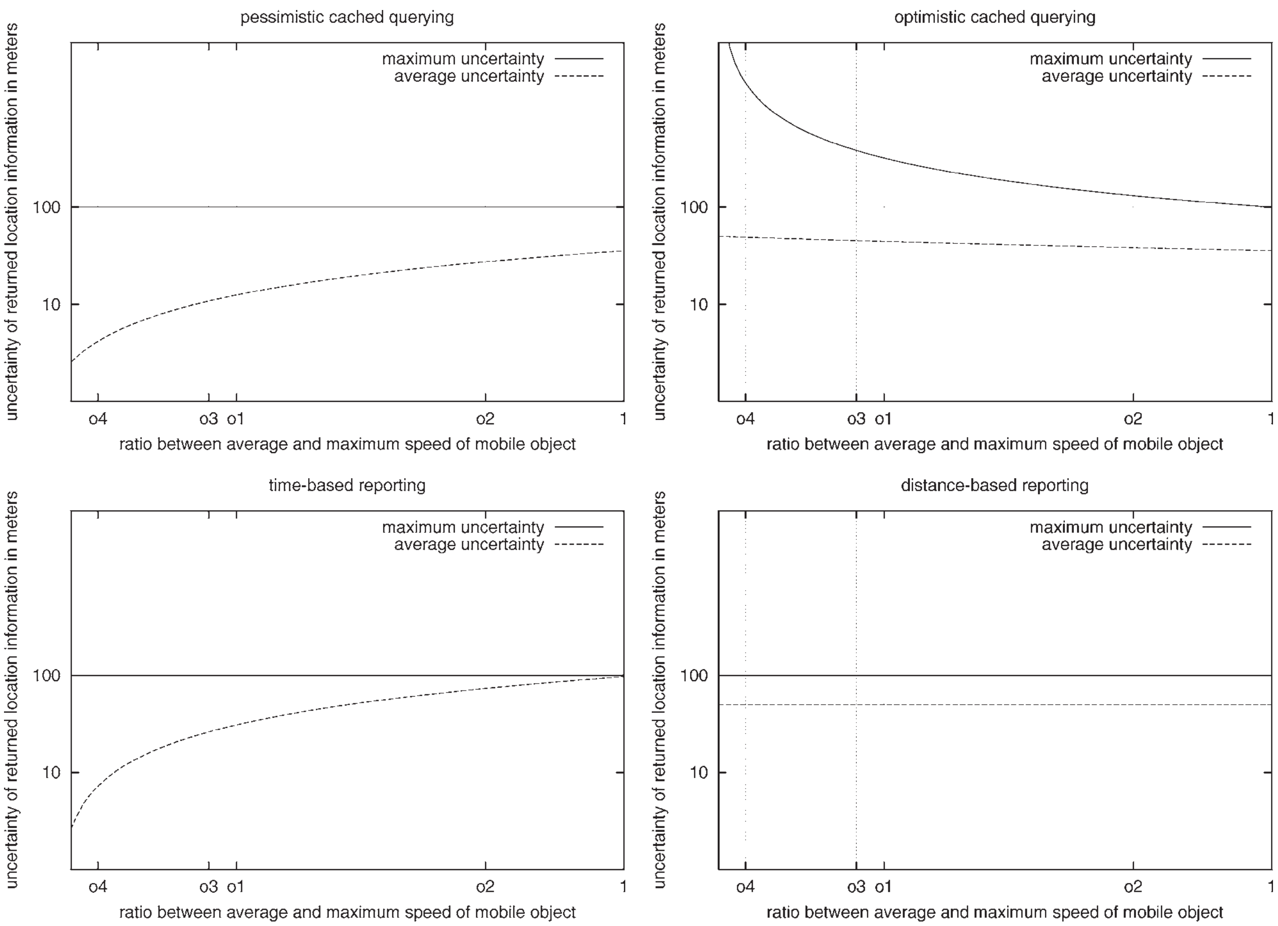
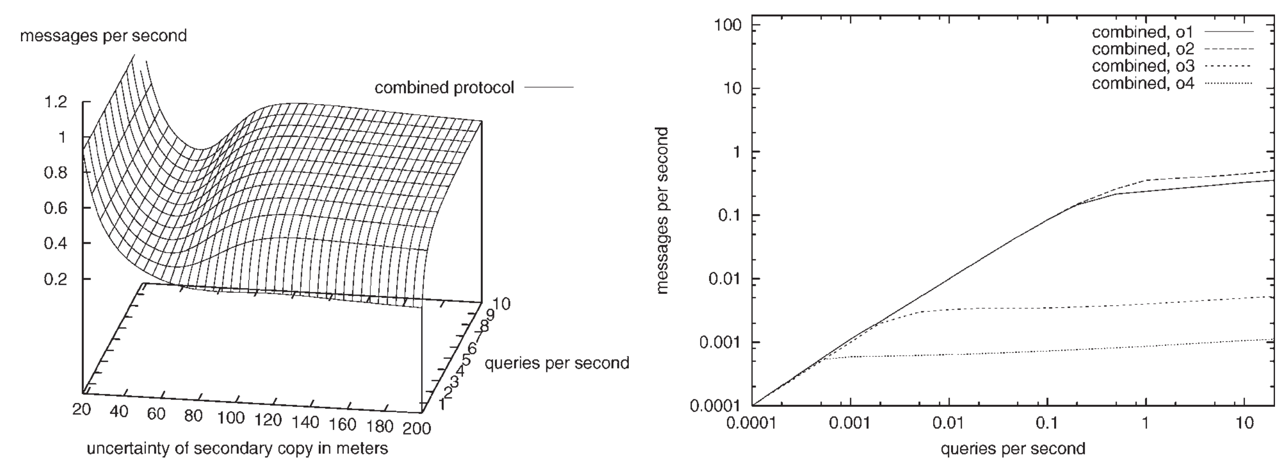


Figure 6. Analytical results: maximum and average uncertainty of the location information on the server for the optimistic and pessimistic cached querying protocols and the time-based and distance-based reporting protocols.

tion 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 ($v_{\text{avg}}/v_{\text{max}}$), which gives a good indication of its mobility characteristics. A mobile object with a low ratio moves sporadically (o_4, e.g., has a ratio of 0.05), an object with a ratio approaching 1 moves steadily (e.g., o_2). For the graphs we have assumed a query rate q of 0.1 queries per second. Figure 6 shows that all except the optimistic cached querying protocol can meet the required accuracy independently of the mobility of the object. The optimistic cached querying protocol is also not suited for objects with sporadic movements as in this case its maximum uncertainty becomes very bad. Compared to the distance-based reporting protocol, the pessimistic cached querying protocol additionally has a better average uncertainty of the position information for objects that move rarely.

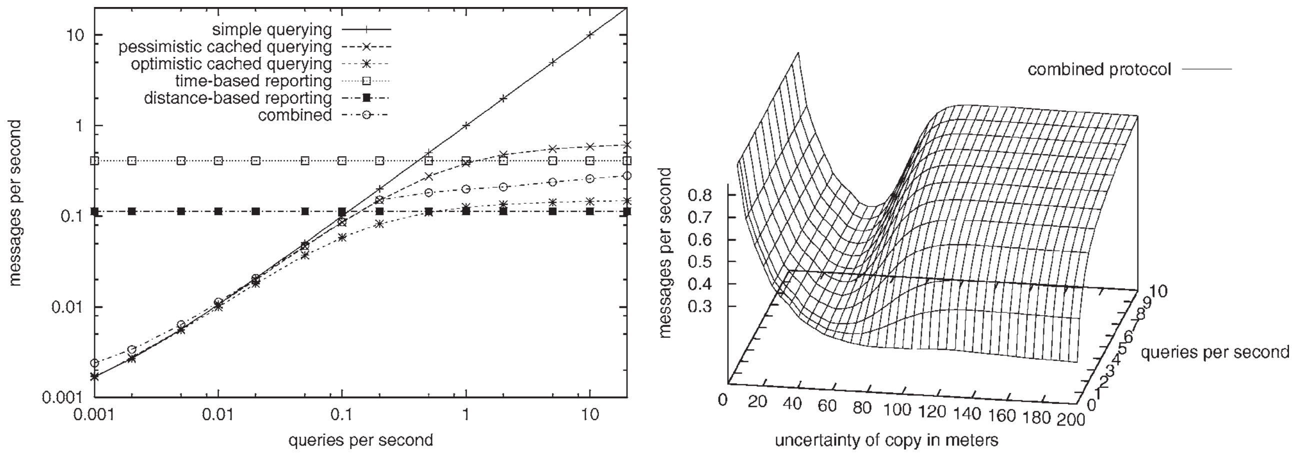
As mentioned before, the performance of the combined protocol depends on the uncertainty with which the secondary copy is updated as well as on the query rate. In figure 7(a), the number of transmitted messages is shown for o_1 and the combined protocol depending on the uncertainty of the secondary copy and the query rate. For the uncertainty demanded in the queries we have assumed a Gaussian distribution with an expected value of $u_{\text{q}} = 100$ m and a standard deviation of 20 m. All other parameters have the same values as before. Figure 7(a) 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 equation (10) to decide whether it is useful to update a secondary copy at all and to find an optimal uncertainty value for it. For comparison with the other protocols, figure 7(b) shows the number of messages transmitted using the combined protocol depending on the query rate for objects o_1–o_4. Optimal values for the uncertainty of the secondary copy have been taken from figure 7(a) and similar figures. 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 copy has to be found.

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



(a) (b)

Figure 7. Analytical results: number of messages transmitted (a) with the combined protocol for o_1 depending on the number of queries and the uncertainty of the secondary copy on the server and (b) depending on the number of queries for o_1 to o_4.



(a) (b)

Figure 8. Simulation results: number of messages transmitted (a) with the querying, reporting and combined protocols determined by simulation and (b) with the combined protocol depending on uncertainty of secondary copy as well as on the query rate.

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 at 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, which have been acquired on a typical car ride in the commuting traffic around the city of Stuttgart (object o_1 in our discussion in section 5). The traces have an average length of about 28 min and an average speed of 44.68 km/h. A Differential GPS sensor with an accuracy between 2 and 5 m has been used to determine the location information, which is written to a file every second. We have implemented 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 requested accuracy. The mean value for the exponential distribution of the time difference 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. Figure 8 shows the results

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

Speed ratio	Pessimistic cached query		Optimistic cached query		Dist-based reporting		Time-based reporting		Combined	
	Max.	Avg.	Max.	Avg.	Max.	Avg.	Max.	Avg.	Max.	Avg.
0.26	74.16	5.92	335.79	22.72	99.81	49.66	99.99	16.97	99.92	6.13
0.29	88.29	6.50	287.79	23.28	99.96	49.00	88.29	17.06	99.50	6.60
0.30	56.28	6.13	294.75	19.69	99.98	50.59	100.07	13.09	101.97	6.20
0.35	104.13	6.76	253.76	20.91	99.89	49.09	100.23	19.12	88.12	7.03
0.36	73.91	6.99	263.79	21.72	99.88	45.73	75.89	17.59	104.24	7.23

of the simulation runs for the different protocols as the mean values of 100 separate simulations. The parameter values for the protocols (e.g., the average and maximum speed) have been taken 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 s. Also, the maximum velocity of the mobile object is sometimes difficult to obtain from the GPS traces because of the uncertainty of the positioning system.

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 also shown in figure 8. 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 0.35 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 of a certain trace. Because we have averaged the maximum speed over 5 s, 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 for different types of mobile objects. These protocols may be used wherever location information is queried or updated continuously. With the help of the presented 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 in the context of our work on a large-scale distributed location service. We also plan to extend the combined protocol by adding a mechanism that allows it to dynamically adapt to changes in the environment. A further interesting area of future work is the integration of a dead-reckoning protocol into the comparison of update protocols.

References

[1] A. Bhattacharya and S. Das, LeZi-update: An information-theoretic approach to track mobile users in PCS networks, in: *Proc. 5th Annual ACM/IEEE Int. Conf. on Mobile Computing and Networking*, Seattle, WA, USA (ACM Press, 1999) pp. 1–12.

[2] A. Bar-Noy, I. Kessler and M. Sidi, Mobile users: To update or not to update? Wireless Networks 1(2) (July 1995) 175–185.

[3] IEEE Computer Society LAN MAN Standards Committee, IEEE Std. 802.11, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification, The Institute of Electrical and Electronics Engineers, New York (1997).

[4] A. Harter, A. Hopper, P. Steggles, A. Ward and P. Webster, The anatomy of a context-aware application, in: *Proc. 5th Int. Conf. on Mobile Computing and Networking (MobiCom '99)*, Seattle, WA, USA (ACM Press, 1999) pp. 59–68.

[5] B. Hofmann-Wellenhof, H. Lichtenegger and J. Collins, *Global Positioning System: Theory and Practice* (Springer, TELOS, 1997).

[6] National Imagery and Mapping Agency, DoD World Geodetic System 1984, its definition and relationship with local geodetic systems, Technical report 8350.2, 3rd edn, National Imagery and Mapping Agency (1997).

[7] U. Leonhardt, Supporting location-awareness in open distributed systems, Ph.D. thesis, Imperial College of Science, Technology and Medicine, University of London (1998).

[8] A. Leonhardi and K. Rothermel, Architecture of a large-scale location service, Technical Report TR-2001-01, University of Stuttgart, Faculty of Computer Science (2001).

[9] H. Maaß, Location-aware mobile applications based on directory services, in: *Proc. 3rd Int. Conf. on Mobile Computing and Networking (MobiCom '97)*, Budapest, Hungary (ACM Press, 1997) pp. 23–33.

[10] M. Mouly and M.-B. Pautet, *The GSM System for Mobile Communications* (Cell & Sys, France, 1992).

[11] C. Perkins, ed., IP Mobility Support, RFC 2002, IETF (1996).

[12] M. Satyanarayanan, Fundamental challenges in mobile computing, in: *Proc. 15th ACM Symp. on Principles of Distributed Computing*, Philadelphia, PA, USA (ACM Press, 1996) pp. 1–7.

[13] M. Spreitzer and M. Theimer, Providing location information in a ubiquitous computing environment, in: *Proc. 14th ACM Symp. on Operating System Priciples* (ACM Press, 1993) pp. 270–283.

[14] R. Want, A. Hopper, V. Falcao and J. Gibbons, The active badge location system, ACM Transactions on Information Systems 10(1) (1992) 91–102.

[15] 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) (1999) 1–31.