Universität Stuttgart
Fakultät Informatik

A Map-based Dead-reckoning Protocol for Updating Location Information

Authors:
Dipl.-Inf. A. Leonhardi
Dipl.-Inf. C. Nicu
Prof. Dr. K. Rothermel

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

A Map-based Dead-reckoning Protocol for Updating Location Information

A. Leonhardi, K. Rothermel

Technical Report 2001/09
December 2001

A Map-based Dead-reckoning Protocol for Updating Location Information

Alexander Leonhardi, Christian Nicu and Kurt Rothermel

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

email: Alexander.Leonhardi@informatik.uni-stuttgart.de

Abstract. An important aspect of location-aware services is the management of location information. To this end, location information needs to be transferred from a mobile device, which determines this information by means of a local positioning sensor (such as GPS), to a location service, where the location information can be queried by applications. Because bandwidth in wide area mobile communications is still scarce and expensive, it is important to use an update protocol that requires as few messages as possible while still guaranteeing a desired accuracy of the location information. To decrease the number of necessary update messages, so-called dead-reckoning strategies have been proposed. In this paper we give an overview of different variants of dead-reckoning protocols for updating location information and propose a new map-based protocol. While a simple dead-reckoning protocol already reduces the number of update messages by up to 83%, the map-based protocol further reduces their number by again up to 60%.

1. Introduction

Advanced location-aware services are considered to be important applications for the up-coming third generation of mobile communication networks. Such applications often rely on a location service [7], [5] or moving objects database [11], to store highly accurate information about the location of the mobile objects. Such a service provides, for example, the functionality to find the nearest taxi cab depending on the user's current location or to address all users that are currently inside a department of a store. Tracked objects may be the users' mobile devices, vehicles or any other mobile object capable of determining (e.g., through a GPS sensor [3]) and transmitting (e.g., via GSM or GPRS [9]) its current location. To achieve a high accuracy of the location information, frequent update messages are required (e.g., more than one update message every 2 s in case of a car travelling with 100 km/h to achieve an accuracy of 50 m), while bandwidth in wireless WAN communication is still scarce and expensive. An import goal when designing a suitable update protocol is therefore, to reduce the number of update messages as much as possible while still providing a given desired accuracy.

One promising approach to reduce the number of update messages when transmitting location information, are the dead-reckoning protocols proposed in [12]. In this approach the mobile object reports an expected future route to the location service, which assumes the object to keep on moving on this route with its last reported speed. The mobile object monitors its actual position and if it differs from the predicted position by more than the desired accuracy, sends an update containing its current position and speed. Thus, a maximum deviation between the position assumed by the service and the actual position can be guaranteed. In many cases, however, the route a mobile object will take is not known by the system beforehand, for example for most car rides and for a walking user.

In this paper we first present a general overview and an appraisal of possible variants of dead-reckoning protocols, which mostly do not rely on a pre-known route. Based on this, we propose a new map-based dead-reckoning protocol, which uses a road map to predict the object's future movements by matching the object's position to a road in the map and assuming that it keeps on moving on this road with its current speed. To evaluate these protocols, we have performed extensive simulations based on real-world GPS traces with different movement characteristics for the mobile objects. Our simulations show that, in case of a car on a freeway, the map-based protocol can reduce the number of necessary updates by more than half as compared to a more simple dead-reckoning protocol, which already is a great improvement against a non dead-reckoning approach.

The remainder of the paper is structured as follows: Section 2 describes the basic idea behind dead-reckoning protocols and discusses possible variants. In Section 3 the proposed map-based dead-reckoning protocol is described in detail and is compared in Section 4 to a more simple dead-reckoning and a non dead-reckoning protocol by means of simulation. Finally, Section 5 discusses related work and Section 6 concludes the paper, giving an outlook on future work.

2. Dead-reckoning Protocols

In general, a dead-reckoning protocol works in the following manner (see Fig. 1): the *source* of the location information determines the current state of the mobile object o_p from an attached *sensor system*. In most cases this state information includes the position of the mobile object *o.pos* with an uncertainty of u_p, its current speed *o.v* and direction of movement *o.dir*[1] as well as a timestamp *o.t*. The source reports its current state to a *location server* (denoted as server, for short), which stores the location information for the mobile object and answers queries from applications regarding its position.

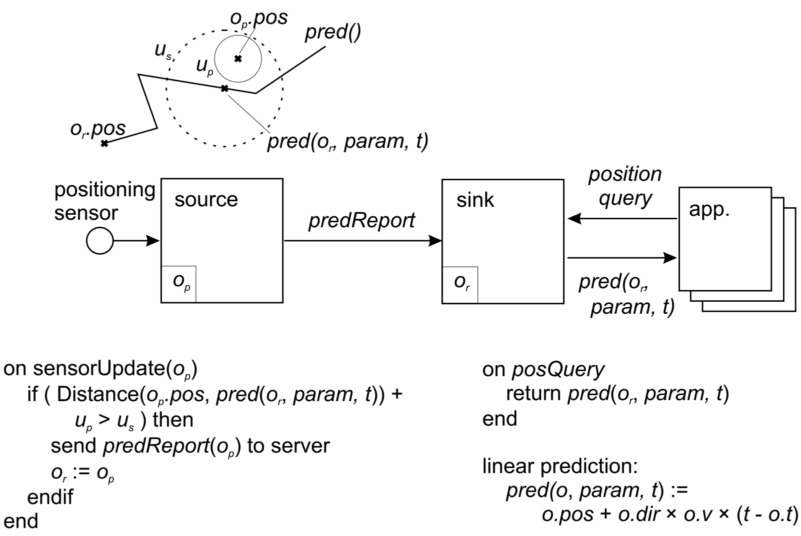

Fig. 1. General functionality of a dead-reckoning protocol.

Both the source and the server use an identical function *pred()* to predict a current position of the mobile object based on the last reported object state o_r, additional parameters *param* (e.g., map information) and the current time *t*. If the source detects that the distance between the mobile object's actual and its reported position is greater than a certain accuracy u_s requested at the server, it sends an update with the object's current state. To be able to guarantee this maximum deviation, it is important that both the server and the source use the same prediction function and parameters.

Variants of this general dead-reckoning protocol mainly differ in their prediction function. While simple prediction functions only use the current state of the mobile object (e.g., direction of

1. If speed and direction are not directly available, they can be inferred from the last *n* position sightings.

movement, acceleration), other variants assume that the mobile object is moving on a route network or on a pre-known route. For an overview of the variants of dead-reckoning protocols, see Fig. 2.

Linear prediction: This simple dead-reckoning protocol assumes that the mobile object keeps on moving along a line given by the reported position and direction and with the reported speed. As shown in our simulations, this simple to implement protocol is in many cases already a significant improvement compared to traditional update protocols, especially if the object is a car moving on a freeway. Also, it can be used as a fall-back for the more advanced protocols. Fig. 3 shows an example of the update messages generated by the linear prediction protocol in one of our simulations. As can be seen, a position update becomes necessary, either because the object has changed its course or its speed. The following protocols try to further reduce the number of updates by addressing these issues.

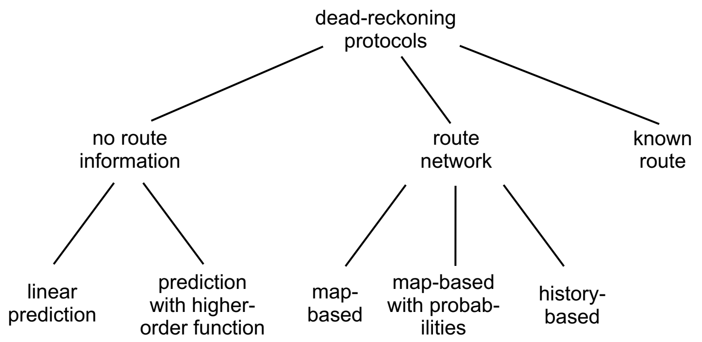

Fig. 2. Overview of different variants of dead-reckoning protocols.

Higher-order prediction function: Instead of predicting the future course as a straight line, it is also feasible to use higher-order functions (curves or splines) which, for example, could capture the object's movements in a curve of the road (see Fig. 3). Similar functions could also be used to predict the speed of the object, for example taking into account its acceleration. However, as the map-based protocol, which is described next, is already able to predict the object's movements more accurately, we have decided not to consider a higher-order prediction function in favour of a map-based protocol.

Map-based dead-reckoning: Mobile objects are often moving along a road network, as a person walking along the streets of a city or a car travelling on a freeway. A map-based dead-reckoning protocol tries to match the object's position to a road of an integrated map and assumes that the object keeps on moving along this street with its current speed. At an intersection the protocol tries to choose a direction the mobile object is most likely to follow. A highly detailed road map is, for example, available as part of a car navigation system, which is included already in a large number of new cars. Further information, like information about main roads or the speed limit on a road, can be extracted from this road map, to further improve the performance of the map-based protocol. The map-based dead-reckoning protocol is described in detail in Section 3.

Map-based with probability information: To improve the prediction of the subsequent direction after a mobile object has passed an intersection, the links in the map can be enhanced with probability information. These probabilities may describe what percentage of all users follows a certain link (user-independent) or how many times a certain object follows this link when moving over the intersection (user-specific). The prediction function then assumes that the object is following the link with the highest probability. To capture these probabilities requires a certain effort, but the protocol can be expected to perform slightly better than a map-based one in case of frequent intersections.

History-based dead-reckoning: If no map is available, it can be generated from traces of the user's past movements. A user will often use routes repeatedly, for example when commuting to work or when driving to his/her favorite shopping center. If the movements are observed over a long time, the

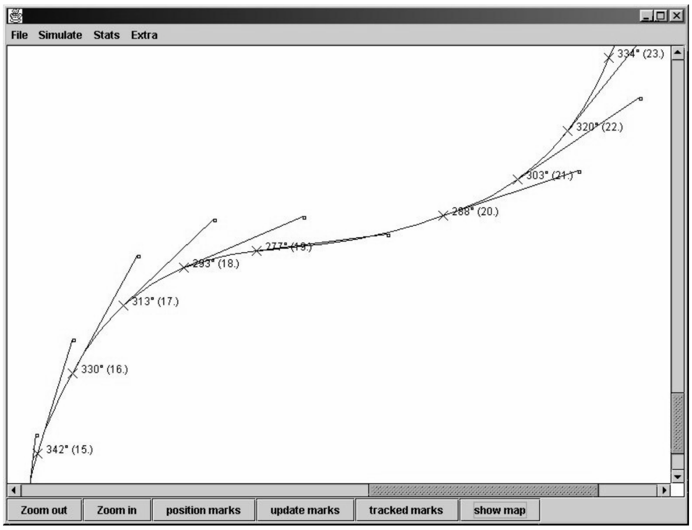

Fig. 3. Screenshot of our simulator, showing 9 position updates with a linear prediction dead-reckoning protocol.

result is a map, which can be used as in the map-based protocols. The generation of the map can again be done for all users (*user-independent*) or for a single user (*user-specific*). Overall, the generation of the maps causes a great overhead and its performance is, when the map-generation is fully completed, equivalent to that of the map-based protocol (with probability information).

Dead-reckoning with known route: If the route of the mobile object is known beforehand, the protocol only needs to consider the object's speed and not the direction of its movement (see [12]). With a known route, a dead-reckoning protocol has the same performance as an optimal map-based protocol, which chooses the right direction at all intersections. However, in this work we do not assume that the object knows its route beforehand.

3. Map-based Dead-reckoning

In our work we have concentrated on the map-based dead-reckoning protocol, which is described in this section. As compared to the basic protocol, the map-based protocol basically executes a map-matching algorithm when monitoring the sensor information at the source and uses a prediction function that is enhanced by map information.

The map information used for the map-based protocol needs to contain information about all available intersections, which are described by a unique identifier and their exact geographical location, and links, which are placed between two such intersections and have again a unique identifier. To be able to model roads more exactly, a link can be divided into a number of sub links by specifying intermediate shape points with their geographic location (see Fig. 4). The map that we have used in our simulations has been extracted from a map used in car navigation systems.

The map matching algorithm selects a current link for the mobile object and places the position returned by the sensor system p_p perpendicularly on this link, to obtain a corrected position p_c (see Fig. 5). A position can be mapped to a link, if it has at most the distance u_m to the link's nearest point. The parameter u_m determines how exact the position must be matched to a link and reflects the accuracy of the sensor system. On initialization, potential links of the map are found by querying a spatial index for the map information with the mobile object's current position. The link with the shortest distance is then selected, if it is not farther away than u_m. An update is sent, when the distance between the actual position of the mobile object and its predicted position is greater than u_s, as

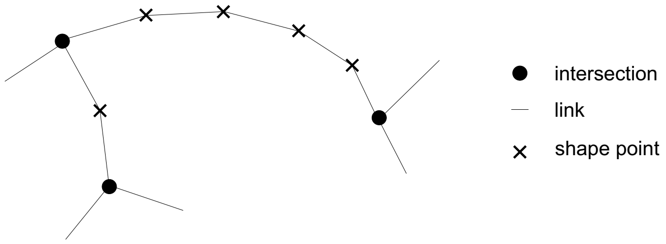

Fig. 4. Example for the map information used by the map-based dead-reckoning protocol.

stipulated by the general mechanism. An update of the map-based protocol contains the mobile object's corrected position $o.p_c$, its speed $o.v$ and the identifier of the current link $o.l$.

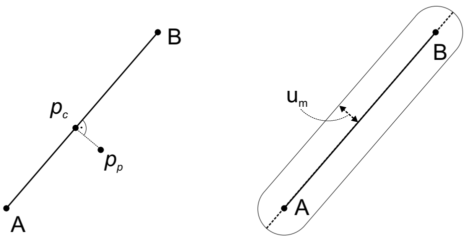

Fig. 5. Map matching used in the map-based dead-reckoning protocol: matching a position p_p *to a position on the link of the map* p_c *(left) and the maximum distance* u_m *that a mobile object's position may have from its current link (right).*

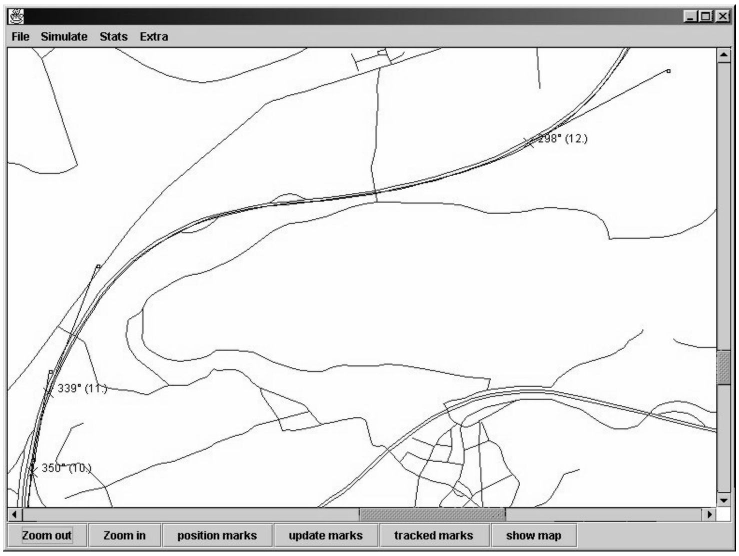

Fig. 6. Screenshot of our simulator, showing 3 position updates with a map-based dead-reckoning protocol on the same route as in Fig. 3.

If the source detects that the object's position is farther than u_m away from its current link (i.e., the mobile object can not be matched to the current link anymore), it tries to find the right link by either forward- or backward-tracking. If the object has passed the end of the current link B (assuming the object is moving from A to B), it has probably reached an intersection and forward tracking is used. To this end, the distance of the object's position to all outgoing links of this intersection is measured and the nearest link is selected as new current link. The transition to a new link is thus delayed, as it is impossible to select the right link directly at B because of the uncertainty of the positioning sensor.

If the object has not passed B, when leaving its current link, the source assumes that it has previously selected the wrong link and tries to correct this with the use of back-tracking. It goes back to the last intersection(s) and checks the other outgoing links. When after forward- or back-tracking no matching link could be found, the source sends an update message with an empty link to the server. In this case, the linear prediction protocol is used as a fall-back. The source periodically compares the object's position with suitable links of the map (found through the spatial index) trying to return to the map-based protocol.

The prediction function assumes that the object goes on following the reported link with its current speed starting from the reported position. When coming to an intersection, the prediction function selects an outgoing link, which it assumes the object to keep on following in the same manner. In our implementation, the link with the smallest angle to the previous link is selected. Ideally, the function would select the main road. However, this information was not easily available in our road map. In our simulations we have found this alternative to be a good approximation.

Fig. 6 depicts the update messages generated by a map-based protocol for the same scenario as shown in Fig. 3. It shows clearly the map-based protocol's capacity to reduce the number of update messages as the prediction function is able to follow the curves of the freeway.

4. Simulation Results

To evaluate the behavior of the presented dead-reckoning protocols we have implemented a linear prediction and a map-based dead-reckoning protocol as well as a non dead-reckoning distance-based reporting protocol as described in our previous work [6]. The distance-based protocol sends an update whenever the actual position deviates from the last reported position by more than a given threshold. We have performed a number of simulations based on real-world GPS traces, to compare the number of update messages required for these protocols.

As the efficiency of the protocols is expected to depend to a great deal on the movement characteristics of the mobile object, we have recorded GPS traces for the following four characteristic movement patterns: (a) a car on a freeway, (b) a car in inter-urban traffic, (c) a car in city traffic and (d) a walking person. Table 1 summarizes the properties of these traces[1]. The traces have been obtained using a Differential GPS receiver, which has an accuracy of 2-5 m. Its output has been written to a file every second.

To evaluate the protocols we have simulated the movements of a mobile object and in our simulator provided the functionality for transmitting the location information between a source and a server. Different variants of update protocols can be plugged into the simulator and be compared according to the number of updates transmitted and the resulting accuracy on the server. Additionally, the simulation program is able to visualize map information and updates generated by the protocols (as seen in Fig. 3 and Fig. 6) and has therefore also been used for optimization and debugging purposes.

	length	**duration**	**average speed**	**max. speed**
car, freeway	163 km	1:35 h	103 km/h	155 km/h
car, inter-urban	99 km	1:39 h	60 km/h	116 km/h

Table 1: Characteristics of the GPS traces used for the simulation.

1. Note that it is difficult to obtain an exact value for the maximum speed from a GPS trace because of the inherent uncertainty of the sensor information.

	length	duration	average speed	max. speed
car, city traffic	89 km	2:25 h	34 km/h	65 km/h
walking person	10 km	2:08 h	4.6 km/h	7.2 km/h

Table 1: Characteristics of the GPS traces used for the simulation.

In Fig. 7 to Fig. 10 the number of update message per hour are shown when using a non dead-reckoning distance-based reporting, a linear prediction dead-reckoning or a map-based dead-reckoning protocol for the different types of movement characteristics. On the left side the absolute number of updates is depicted, while the right side shows them in relation to the non dead-reckoning protocol. In each graph the uncertainty requested at the server is varied from 20 m to 500 m in case of a car and from 20 m to 250 m in case of a walking person.

The speed and direction have been interpolated from 2 consecutive positions reported by the positioning sensor in case of a freeway traffic, from 4 positions in case of city or inter-urban traffic and from 8 positions in case of a walking person. These numbers, which depend on the uncertainty of the sensor system and the speed of the mobile object, have been found to be optimal for that purpose. For more details see [8].

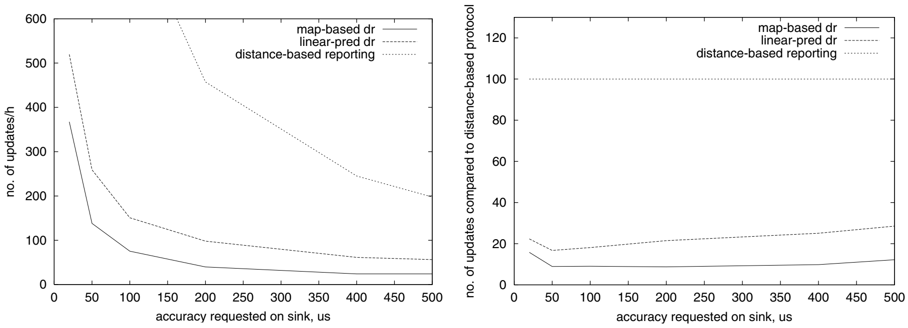

Fig. 7. Freeway traffic: number of update messages per hour (left) or relative to the non dead-reckoning distance-based reporting protocol (right) for different requested accuracies caused by a linear prediction and map-based protocol.

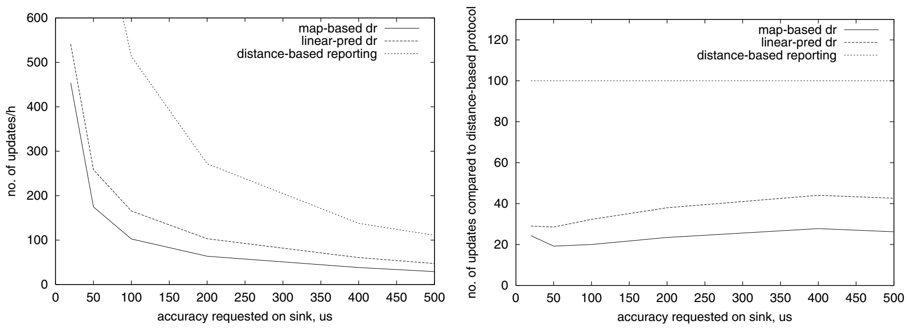

Fig. 8. Inter-urban traffic: number of update messages per hour (left) or relative to the non dead-reckoning distance-based reporting protocol (right) for different requested accuracies caused by a linear prediction and map-based protocol.

The results show that the dead-reckoning algorithm with linear prediction is already a great improvement on the simple reporting protocol. As expected, the improvement is more marked for a car on a freeway (reducing the update rate by up to 83%) than in city traffic (up to 63%). The reason is the much more steady traffic, with respect to changes in speed and direction, as well as the lower number of intersections on a freeway. Also, improvements decrease for a larger requested uncertainty of the location information as with a longer update interval it becomes more likely that the object changes its course. There is a small performance decrease at the lowest considered uncertainty of 20 m, because of the beginning influence of the uncertainty of the GPS receiver.

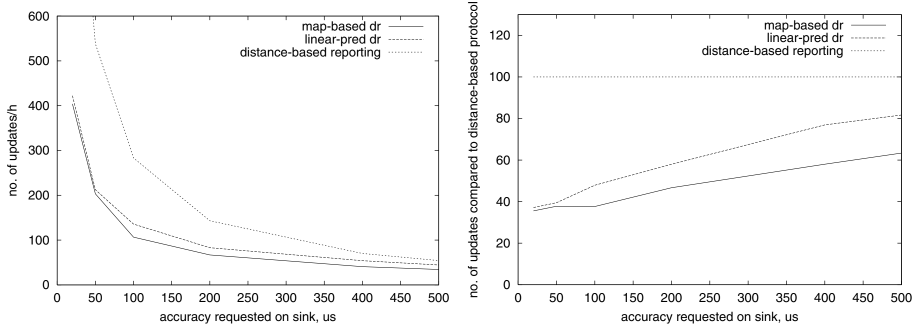

Fig. 9. City traffic: number of update messages per hour (left) or relative to the non dead-reckoning distance-based reporting protocol (right) for different requested accuracies caused by a linear prediction and map-based protocol.

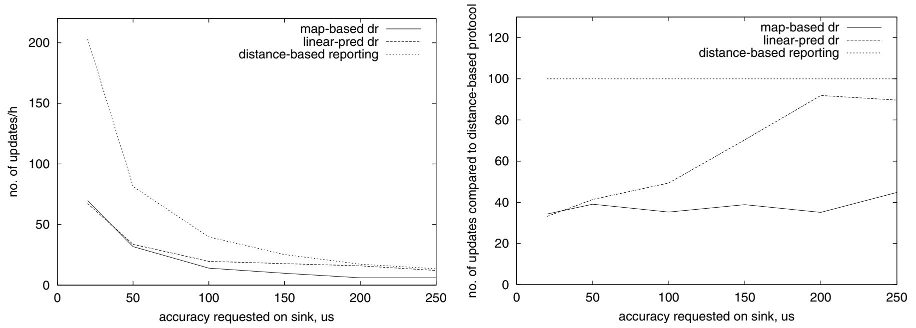

Fig. 10. Walking person: number of update messages per hour (left) or relative to the non dead-reckoning distance-based reporting protocol (right) for different requested accuracies caused by a linear prediction and map-based protocol.

In case of freeway traffic, the map-based dead-reckoning protocol reduces the number of updates again by up to 60%, as compared to the linear prediction strategy. Only for a walking user and the highest requested accuracy does the linear prediction protocol require fewer updates. In addition, the map-based dead-reckoning protocol is able to uphold these improvements for a higher requested uncertainty, as it is able to predict the mobile object's movements better (see Fig. 10).

5. Related Work

An important aspect in location management for the mobile phones in personal communication systems (PCS) is the updating of the location information (e.g., [4]). In today's systems (e.g., GSM) a mobile phone reports its current location area to the PCS system, each time it enters a new one. Information about a mobile phone's current location area is stored in the Home Location Register (HLR). If a call arrives for a certain mobile phone, it is paged on a special channel in all communication cells belonging to the location area stored in the phone's HLR. Different protocols for updating and paging the location information have been discussed, for example a distance, a movement and a time-based protocol in [1] or the LeZi-update protocol in [2]. A dead-reckoning approach is not feasible in this context because of the comparatively large size of the location areas (up to 40 km). In previous work [6], we have compared different non dead-reckoning reporting and querying protocols, including the distance-based reporting protocol, in the context of updating highly accurate location information.

As mentioned above, [12] describes a dead-reckoning protocol for a route that is known in advance, to update a database for mobile objects (see [11]). To determine the distance threshold, at which the source sends a position update, a cost function has been defined. It covers costs for the uncertainty caused by the distance threshold, costs for the deviation between real and actual position and costs for sending the update messages. Different variants of dead-reckoning strategies are compared with the goal of minimizing this cost function. The results show that, under these conditions, it is advantageous to use a protocol that adaptively adjusts the distance threshold (*adaptive dead reckoning, adr*) as compared to a more simple strategy (*speed dead-reckoning, sdr*). To be able to deal with disconnections of the mobile communication link, a variant of these strategies has been designed that continuously decreases the distance threshold, if no updates are sent (*disconnection detection dead-reckoning, dtdr*). How the mobile object determines its (future) route is not discussed in this work.

Matching the position of a mobile object to the links of a map is an important aspect in navigation systems (for an overview see [13]). In the literature, map-matching algorithms using fuzzy-logic or Kalman-filters have been discussed. Curve-based map-matching algorithms consider not only the mobile object's current location for matching, but also the whole geometry (curvature) of the covered way and use it to find a matching position on the map. In our work, we have used a simple map-matching algorithm, which has proven to be sufficient for our purposes. More advanced map-matching algorithms could be used to further improve the identifying of the mobile object's current link and to reduce the updates resulting from a wrong matching.

Originally, the term dead-reckoning stems from the navigation on ships. Here it stands for a method to determine the current position by adding the object's movements, given by its current course, speed, drift etc., to a known original position. Similar methods are used in robotics, aerial photography and car navigation where secondary sensors, for example inertial sensors or an odometer, which do not directly determine a position, are used to support a primary positioning sensor, often GPS (see [10]). As the uncertainties of the secondary sensors add up over time, this method needs to access the primary sensor from time to time to resynchronize. As these methods require constant access to the secondary sensors, they are not suitable to reduce update messages, but could be used to increase the accuracy of the location information of the source.

6. Conclusion and Future Work

In this paper we presented the concept and an overview of possible dead-reckoning protocols for the transmission of location information. We described a new map-based dead-reckoning protocol and evaluated its performance compared to a simple linear prediction and a non dead-reckoning protocol for different characteristic movement patterns of mobile objects.

We showed that employing a simple dead reckoning protocol can already lead to a considerable reduction of the number of update messages necessary to transmit accurate location information. Our simulations also showed that the number of update messages can be further reduced by using a more sophisticated map-based prediction function (overall up to 91%). We therefore believe that dead-reckoning protocols are an important building block for the highly accurate tracking of the location of mobile objects in location-aware applications.

Our work has mainly been aimed at reducing the update messages caused by the mobile object changing its course. Most of the remaining updates are caused by a change of speed, in an extreme example when a car has to stop at a traffic light. Future work will therefore have the goal of addressing this issue. The map-based protocol could, for example, use knowledge about the speed limits for the roads to appropriately change the mobile object's assumed speed. However, in our case this would require an extensive extraction of the speed limits from special information about traffic signs in the car navigation map. It would also be possible to try to capture the acceleration of objects. Because of the relatively short duration of the acceleration phases this would only make sense for a very low requested uncertainty and only if additional factors such as the intended final speed is taken into account.

References

[1] A. Bar-Noy, I. Kessler and M. Sidi: Mobile Users: To Update or not to Update?, in *Wireless Networks*, 1(2), pp. 175-185, 1995.

[2] A. Bhattacharya and S. Das: LeZi-update: An Information-theoretic Approach to Track Mobile Users in PCS Networks, in *Proc. of the 5th Annual ACM/IEEE Int. Conf. on Mobile Computing and Networking (MobiCom '99)*, Seattle, WA, USA, pp. 1-12, 1999.

[3] B. Hofmann-Wellendorf, H. Lichtenegger and J. Collins: *Global Positioning System: Theory and Practice*,, Springer, Vienna, Austria, 1997.

[4] D. Lam, Y. Cui, D. C. Cox and J. Widom: A Location Management Technique to Support Lifelong Numbering in Personal Communications Services, in *ACM Mobile Computing and Communications Review*, 2(1), pp. 27-35, 1998.

[5] A. Leonhardi and K. Rothermel: *Architecture of a Large-scale Location Service*, Technical Report TR-2001-01, Faculty of Computer Science, University of Stuttgart, 2001.

[6] A. Leonhardi and K. Rothermel: A Comparion of Protocols for Updating Location Information, in *Baltzer Cluster Computing Journal,* 4(4), pp. 355-367*,* 2001.

[7] U. Leonhardt: *Supporting Location-Awareness in Open Distributed Systems*, Dissertation, Imperial College of Science, Technology and Medicine, University of London, 1998.

[8] C. Nicu: *Untersuchung von Koppelnavigations-Protokolle für die Übertragung von Positionsinformationen*, Master's thesis No. 1919, Faculty of Computer Science, University of Stuttgart, 2001 (in German).

[9] J. Schiller: *Mobile Communications*, Addison- Wesley, 2000.

[10] J. Skaloud and K. P. Schwarz: Accurate Orientation for Airborne Mapping Systems, in *ISPRS Journal of Photogrammetry and Remote Sensing* 66, 4, pp. 393-402, International Society for Photogrammetry and Remote Sensing, 2000.

[11] O. Wolfson, B. Xu, S. Chamberlain and L. Jiang: Moving Objects Databases: Issues and Solutions, in *Proc. of the 10th Int. Conf. on Scientific and Statistical Database Management (SSDBM '98),* Capri, Italy, pp. 111-122, 1998.

[12] O. Wolfson, A. P. Sistla, S, Chamberlain and Y. Yesha: Updating and Querying Databases that Track Mobile Units, in *Distributed and Parallel Databases Journal,* 7(3), pp. 1-31, 1999.

[13] Y. Zhao: *Vehicle Location and Navigation Systems*, Artech House Publishers, Norwood, MA, USA, 1997.