RoadAlarm: a Spatial Alarm System on Road Networks

Kisung Lee, Emre Yigitoglu, Ling Liu, Binh Han, Balaji Palanisamy, Calton Pu

DiSL, College of Computing, Georgia Institute of Technology

{kisung.lee, eyigitog, lingliu, binh.han, balaji, calton}@cc.gatech.edu

***Abstract*—Spatial alarms are one of the fundamental functionalities for many LBSs. We argue that spatial alarms should be road network aware as mobile objects travel on spatially constrained road networks or walk paths. In this software system demonstration, we will present the first prototype system of** ROADALARM **- a spatial alarm processing system for moving objects on road networks. The demonstration system of** ROADALARM **focuses on the three unique features of** ROADALARM **system design. First, we will show that the road network distance-based spatial alarm is best modeled using road network distance such as segment length-based and travel time-based distance. Thus, a road network spatial alarm is a star-like subgraph centered at the alarm target. Second, we will show the suite of** ROADALARM **optimization techniques to scale spatial alarm processing by taking into account spatial constraints on road networks and mobility patterns of mobile subscribers. Third, we will show that, by equipping the** ROADALARM **system with an activity monitoring-based control panel, we are able to enable the system administrator and the end users to visualize road network-based spatial alarms, mobility traces of moving objects and dynamically make selection or customization of the** ROADALARM **techniques for spatial alarm processing through graphical user interface. We show that the** ROADALARM **system provides both the general system architecture and the essential building blocks for location-based advertisements and location-based reminders.**

I. INTRODUCTION

Spatial alarms extend the concept of time-based alarms to spatial dimension and remind us when we travel to some predefined location of interest in the future. An example of road network aware spatial alarms is "alert me when I am within 2 miles of the dry clean store at the junction of Druid Hill and Baircliff".

It is important to note that spatial alarms are standing spatial triggers and are very different than continuous spatial range queries. Thus using techniques for processing continuous spatial range queries to evaluate spatial alarms will deliver very poor performance for at least two reasons. First, spatial range queries require continuous evaluation as the focal object of the queries is moving on the road. In contrast, spatial alarms do not need to be evaluated when their mobile subscribers are marching on the road. Instead, spatial alarms only need to be checked when the mobile user approaches the vicinity of her subscribed alarms. Second, spatial range queries are asking for spatial information that is relevant to the current location of the query focal object (user). In contrast, spatial alarms are referring to spatial information that is not directly relevant to the current location of its mobile subscribers. Thus it would be completely wasteful if we apply spatial range query processing algorithms and evaluate all the spatial alarms subscribed by a mobile user as soon as this mobile object moves on the road. This is because many of the subscribed alarms may be far away from the current location of the moving object and thus there is zero probability for them to be triggered before the moving object approaches the alarm target.

Spatial alarms can be categorized as *private*, *shared* or *public* alarms depending on the scope of subscribership of the alarm. Private alarms are relevant to a single subscriber authorized to install or remove the alarm. A subscriber may install an alarm on the neighborhood grocery store reminding her to purchase groceries when she is within a one mile radius of the store. Shared alarms are installed by a subscriber of the alarm and may be shared with a group of users; for example, in the above scenario the subscriber may wish to share the alarm on the dry clean store with other members of her household. Public alarms are relevant to all subscribers in the system; examples of such alarms are warning notifications against hazardous road conditions.

Spatial alarms are personalized location-based triggers installed by mobile users to serve as a reminder of a location of interest to be encountered in their future trips. Unlike continuous spatial queries, spatial alarms do not require immediate processing and periodic reevaluation upon installation. Thus, a critical challenge for efficient processing of spatial alarms is to determine when to evaluate each spatial alarm, while ensuring the demanding requirements of *high accuracy* and *high system scalability*. *High accuracy* ensures no alarms are missed, and *high system scalability* guarantees that the alarm processing system scales to a large number of spatial alarms and growing base of mobile users.

Different approaches may be deployed in order to process spatial alarms. A simple approach is *periodic evaluation* at a high frequency. Each spatial alarm evaluation can be conducted by testing whether the user is entering the spatial region of any of her relevant alarms. High frequency is essential to ensure that none of the alarms are missed. Though periodic evaluation is simple, it can be extremely inefficient due to frequent alarm evaluation and the high rate of irrelevant evaluations.

Furthermore, existing alarm processing approaches [1] are based on the Euclidean distance space model and fail to incorporate road network distance into spatial alarm definition and spatial alarm processing. As a result, they tend to incur unnecessary wakeups and shorter hibernation time at mobile clients and unnecessary computation and alarm checks at the

spatial alarm processing server.

To the best of our knowledge, ROADALARM is the first system that optimizes the spatial alarm processing by taking into account spatial constraints on road networks and mobility patterns of mobile devices. We introduce road network-based spatial alarms using two types of road network distances (segment length-based and travel time-based). Furthermore, we develop a suite of road network aware alarm processing techniques and show that ROADALARM can significantly enhance the efficiency and scalability of spatial alarm processing.

II. ROADALARM SYSTEM ARCHITECTURE

The ROADALARM system is composed of a spatial alarm processing engine and a location server as shown in Fig. 1(a). Concretely, the spatial alarm processing engine communicates with each mobile client to send spatial alarm alerts or **hibernation time**, which is a time interval during which the mobile client does not need to wake up and the processing engine does not need to perform alarm checks for this mobile client. To calculate the hibernation time and check the spatial alarm alerts, the spatial alarm processing engine communicates with the location server to obtain the current road network locations of mobile clients as well as the road network locations of alarm targets for all alarms maintained in its database. The location server uses localization techniques (such as GPS, WiFi or any hybrid localization technology) to keep track of the current positions of moving objects (mobile clients). It also manages locations of moving objects and the locations of static objects (such as gas stations, restaurants, and so on). Mobile clients need to install the thin client of ROADALARM as a mobile application on their devices. Each mobile client will obtain an initial hibernation time at the commit of her alarm installation or subscription. Upon the expiration of its old hibernation time, the mobile client will automatically contact the spatial alarm processing engine to obtain its new hibernation time. We assume that the mobile clients are able to communicate with the engine through wireless data channel. During the hibernation time, the ROADALARM application is hibernated at the client side of the mobile client, and the engine pays zero alarm processing cost for this mobile client.

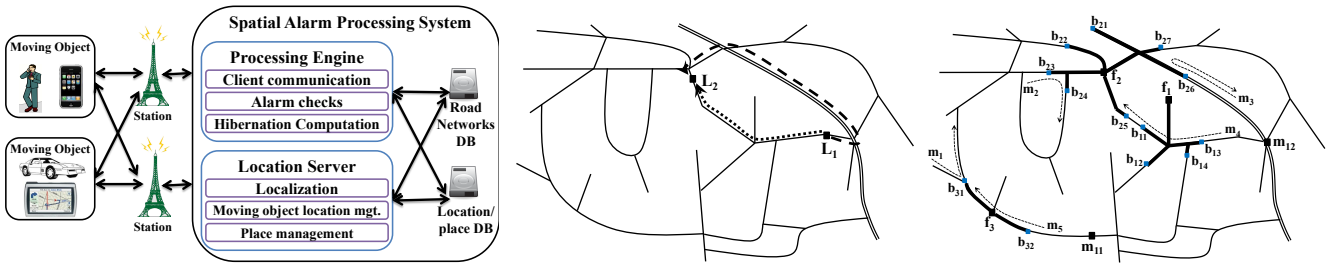
The most commonly used distance measure between two locations on road networks is the summed lengths of constituent segments of a path having the smallest sum among all possible paths connecting the two locations. We call this distance the **segment length-based road network distance** and the path having this distance the **segment length-based shortest path**. However, the segment length-based distance may not provide sufficient and accurate distance information in terms of actual travel time from the current location (L_1) to the destination (L_2), considering that highway road segments are much longer but also with much higher speed limits and thus may have relatively lower travel time compared to some local road segments. Fig. 1(b) shows an example where a mobile client is moving from L_1 to L_2. A double line represents a freeway where speed limit is 65 mph and a single line represents a local road where speed limit is 35 mph. Assume that the path using only local roads (the dotted path in the figure) is the segment length-based shortest path and its distance is 10 miles. The expected travel time from L_1 to L_2 of this shortest path is about 17 minutes ($\frac{10miles}{35mph}$) if the moving object travels at the speed limit. In contrast, the dashed path in the figure is an alternate path whose segment length-based distance is 12 miles, longer than the dotted path, including 10 miles of the freeway and 2 miles of local roads. If the object travels at the speed limit along this dashed path, the expected travel time is about 12.5 minutes ($\frac{10miles}{65mph} + \frac{2miles}{35mph}$), faster than the segment length-based shortest path. To ensure high accuracy and high performance of spatial alarm processing, in ROADALARM we need to consider the **travel time-based distance** as an alternative road network distance measure.

Existing work, to the best of our knowledge, has used a rectangle region to define spatial alarms in terms of Euclidean distance. Even though it is simple to define and handle the rectangular spatial alarms in the Euclidean space, they are not appropriate on road networks since the road network distance is usually much longer than the Euclidean distance and even there could be no path connecting the two road network locations. In the ROADALARM system, we define road network-based spatial alarms in terms of the segment length-based and travel time-based distance. A road network-based spatial alarm is a star-shaped subgraph defined as $SA_{RoadNetwork}(p_f, r, S)$ where p_f is the alarm target or so called the focal point (a road network location) of this alarm, r is the alarm monitoring region, represented by a spatial range (segment length or travel time) from p_f, and S is a set of mobile clients which have subscribed this alarm. Consider Fig. 1(c) that shows three star-shaped alarms with focal points f_1, f_2 and f_3.

A. Spatial Alarm Processing in ROADALARM

The Euclidean distance-based approach is often considered as an intuitive way to implement road network-based spatial alarm processing. To calculate the hibernation time for a moving object m, the Euclidean distance from the current location of m to the nearest spatial alarm and the global maximum speed are used. However, the hibernation time is computed using the Euclidean distance rather than road network distance, thus the hibernation time is unnecessarily short. Consequently, mobile clients need to wake up frequently, making ROADALARM consuming high battery energy.

Another intuitive approach to evaluating road network-based spatial alarms is to use Dijkstra's network expansion algorithm. To find the nearest spatial alarm in terms of the road network distance of a moving object m, shortest paths from the current location of m to all spatial alarms are computed using the Dijkstra's algorithm and then a path having the smallest segment length-based or travel time-based distance is selected. The travel time of the selected path is used as the hibernation time for m. However, the computation cost of this approach is extremely high since they consider all spatial alarms subscribed by m at each wakeup. Furthermore, if an alarm is far away from the current location of m, it is highly costly to compute the shortest path using the



(a) System Architecture (b) Segment length vs Travel time (c) Road Network-based Spatial Alarms

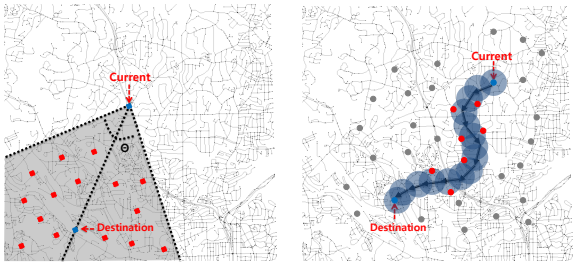
Fig. 1. ROADALARM System

Dijkstra's algorithm because it expands exhaustively too many unnecessary nodes and edges.

To solve the problems of above approaches, in ROADALARM, we use the subscriber-based filter to consider only those alarms that are subscribed by the mobile object m. Similarly, we utilize the concept of *Euclidean lower bound (ELB)* as another type of filter to minimize the number of shortest path computations by filtering out some irrelevant spatial alarms. We call this second type of filter the ELB filter. The concept of *Euclidean lower bound* refers to the fact that the segment length-based distance between two network locations L_1 and L_2 is at least equal to or longer than the Euclidean distance between L_1 and L_2. By combining subscriber-based filter and ELB filter, the ROADALARM basic approach (*BA*) can significantly reduce the shortest path computations required for computing hibernation time for each mobile client.

When a moving object m wakes up due to the expiration of its hibernation time, *BA* first selects spatial alarms subscribed by m through the subscriber-based filtering. Next, an ELB-based filtering method is utilized to find the nearest alarm to the current location (L_m) of m through the Incremental Euclidean Restriction (IER) algorithm. Concretely, instead of computing shortest paths from L_m to every alarm subscribed by m, *BA* computes the new hibernation time of m in five steps: *First*, instead of computing shortest paths from L_m to every alarm of m, *BA* computes the Euclidean distance between L_m and every spatial alarm of m and sort the set of spatial alarms based on their Euclidean distances to L_m in an ascending order. *Second*, let a_{nn} denote the spatial alarm that has the smallest Euclidean distance to L_m. We compute the shortest path between L_m and a_{nn} using segment length-based distance. Let *sldistance*(a_{nn},L_m) denote the segment length-based distance between L_m and a_{nn}. *Third*, we use a binary search algorithm to examine the sorted list of spatial alarms and remove those alarms whose Euclidean distance to L_m is bigger than *sldistance*(a_{nn},L_m). *Fourth*, for each remaining spatial alarm a_j, *BA* computes the shortest path between a_j and L_m respectively. Let *sldistance*(a_j,L_m) denote the segment length-based distance between a_j and L_m. If *sldistance*(a_j,L_m) $<$ *sldistance*(a_{nn},L_m) holds, we assign a_j to be a_{nn}. *Finally*, *BA* finds the nearest spatial alarm a_{nn} to L_m and thus uses the travel time of the segment length-based shortest path to a_{nn} as the hibernation time for m.

When we use the travel time-based distance instead of the segment length-based distance, we cannot directly use the



(a) Using destination vector (b) Using shortest path

Fig. 2. Mobility-aware Optimizations

ELB-based filtering since the Euclidean lower-bound property does not hold for the travel time-based distance. For example, when the Euclidean distance and the segment length-based distance between an alarm and the current location of m are 5 miles and 10 miles respectively, there could be another alarm in which the Euclidean distance and the segment length-based distance are 12 miles and 15 miles respectively, but it has shorter travel time-based distance since there is a freeway connecting the alarm and the current location of m. Therefore, we extend the ELB-based filtering for the travel time-based distance. Instead of using only segment lengths, we define the *travel time-based Euclidean lower-bound* as the travel time multiplied by the global maximum speed limit. For example, if the travel time from the current location of m to a spatial alarm is 1 hour and the global maximum speed limit is 70 mph, the travel time-based ELB is 70 miles (1h x 70mph). We exclude spatial alarms whose Euclidean distance is longer than 70 miles since the moving object m cannot get to those alarms within 1 hour even if it travels at the global maximum speed. The remaining steps for calculating the hibernation time are the same as those in the approach using the segment length-based ELB. In the first prototype of ROADALARM we use the global maximum speed limit for travel time-based ELB in order to ensure the high accuracy of alarm evaluation. It would be interesting to use some less conservative speed or motion metrics to see if we can further reduce the search space needed for computing the hibernation time for mobile objects at the cost of a small and affordable accuracy loss.

B. Mobility-aware Optimizations

Even though the subscriber filter and the ELB filter reduce the number of spatial alarms to be evaluated and thus the computation cost while ensuring zero or a very low and negligible alarm miss rate, the ELB filter is not always effective since there are some cases in which the number of

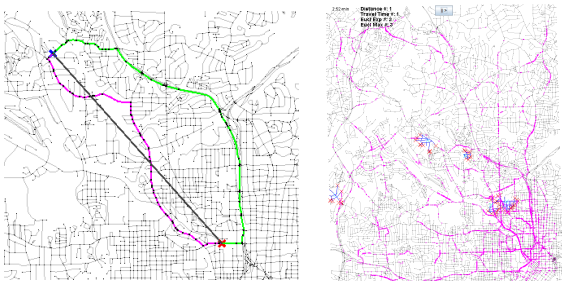
spatial alarms after applying ELB filter remains to be high. To reduce the computation cost more while guaranteeing low alarm miss rate, we devise some mobility-aware filters using the concept of steady motion assumption. This enables us to further remove those irrelevant spatial alarms and thus reduce the search space of hibernation time computation. The steady motion assumption refers to the fact that moving objects on the road will move along its current direction for a certain period of time. It is well suited to the road networks since moving objects can move only on the predefined roads in the spatially constrained space.

In this ROADALARM demonstration, we present two types of steady monition-based mobility-aware filters: *DSM* using the *destination vector* in which the current location and the destination location are used as the initial and terminal point of the vector respectively, and *SPSM* using the shortest path from the current location of a moving object to its current destination. Both assume that the destination location is given by the moving object. In *DSM* we assume that moving objects will move toward their destination within the confidence degree Θ and thus only those spatial alarms which reside in the area defined by the destination vector and Θ will be examined as shown in Fig. 2(a). *DSM* finds the nearest spatial alarm among the selected alarms using *BA* and then uses the travel time, of the shortest path from the current location of a moving object m to the nearest spatial alarm, as the new hibernation for m. However, *DSM* can be inefficient in finding the nearest alarm and computing hibernation time due to the potentially large search space, especially when Θ is large and many alarms are subscribed by moving objects.

The shortest path steady motion (*SPSM*) approach further improves the *DSM* approach by reducing the search space through shortest path-based filtering while maintaining high accuracy. Initially, *SPSM* calculates the shortest path from the current location to the destination for each moving object and then it calculates the hibernation time for this mobile object by considering only those spatial alarms within a boundary distance d from the shortest path as the target alarms (see Fig. 2(b)). The distance d indicates the level of steadiness: if a moving object follows the calculated shortest path, a small value of d capturing the possibilities of short distance detour is sufficient. Such offset distance d can be set as default by the system based on road network characteristics and customized by the mobile users. For each moving object, *SPSM* stores the set of target alarms and their shortest path distance from the moving object. When a moving object m wakes up, *SPSM* retrieves the stored target spatial alarms of m and finds the nearest spatial alarm among the retrieved alarms using *BA*. For further details, we refer to our paper [2].

III. SYSTEM DEMONSTRATION PLAN

We plan to demonstrate the ROADALARM system in three phases. First, we show how end users can install private, shared or public spatial alarms on a road network of their choice. Second, we show how the installed alarms are being processed with different distance functions. In the third phase, we show the performance and effectiveness of our road network aware spatial alarm processing algorithms.



(a) Different types of distances (b) Alarm processing

Fig. 3. ROADALARM GUI screenshots

Concretely, in Phase I, users can select any real road network of their own interest from USGS [3] through a graphical user interface (GUI) and generate mobility traces on the selected road network map using our *gt-mobisim* simulator [4]. In addition, we will have a set of road network maps of US cities and mobility traces prepared for demonstration. Users can directly install private or shared or public alarms on the road network of their choice. The ROADALARM system GUI can be used to visualize not only the road networks and the generated mobility traces but also different types of spatial alarms and our ROADALARM algorithms for processing them. For a given moving object, we show which of her alarms are triggered as the object moves on the road network. In addition, we show how the system is working under a set of moving objects, each subscribed to a set of spatial alarms. Two example screen shots are given in Fig. 3 (northwest Atlanta).

In Phase II, we will show how a location-based advertisement service can be supported by our RoadAlarm system. We will also provide two scenarios to illustrate why spatial alarms are completely different from spatial range queries in terms of both location-based service semantics and processing logic.

In Phase III of the demo, we show two types of performance comparisons: First, we compare ROADALARM with the existing Euclidean-based approach and the conventional network expansion approach. Second we compare the ROADALARM basic approach empowered by subscriber filter and ELB filter (*BA*) with two of the mobility-aware optimizations (*DSM* and *SPSM*). We end our demonstration by showing how spatial alarms can be used as an effective means for location-based advertisements and location-based entertainments.

ACKNOWLEDGMENT

This work is partially sponsored by grants from NSF CISE NetSE program, SaTC program and I/UCRCs program, an IBM faculty award and a grant from Intel ISTC on Cloud Computing.

REFERENCES

[1] M. Doo, L. Liu, N. Narasimhan, and V. Vasudevan, "Efficient indexing structure for scalable processing of spatial alarms," in *GIS*, 2010.
[2] K. Lee, L. Liu, S. Meng, and B. Palanisamy, "Scaling Spatial Alarm Services on Road Networks," in *ICWS*, 2012.
[3] "U.S. Geological Survey," http://www.usgs.gov/.
[4] "gt-mobisim," http://code.google.com/p/gt-mobisim/.