Research Article

Phoenix: A Collaborative Location-Based Notification System for Mobile Networks

Yongping Xiong,¹ Yubo Deng,² Wendong Wang,¹ and Jian Ma¹

¹ State Key Lab. of Networking and Switching Tech., Beijing University of Posts and Telecom., Beijing 100876, China

² Institute of Sensing Technology and Business, Beijing University of Posts and Telecom., Beijing 100876, China

Correspondence should be addressed to Yongping Xiong; ypxiong@bupt.edu.cn

Received 28 November 2013; Revised 29 January 2014; Accepted 18 February 2014; Published 24 March 2014

Academic Editor: Yang Shi

Copyright © 2014 Yongping Xiong et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Location-based notification (LBN) aims to alert the users in a target area to their interested information. With a wide range of applications, LBN has been gaining more and more attraction among wireless users and service providers. The mainstream centralized solution based on cellular networks may incur high service cost. In this paper, we present an innovative scheme called Phoenix, which does not rely on any infrastructure, to implement location-based notification service. In our design, devices (users) across the target area form a dynamic peer-to-peer network, where a user can be a message source, a message carrier, or a message subscriber. When a user meets the message carrier, he will get a copy of the message. Phoenix keeps messages of interest being circulated in the target area; hence users are being notified. To achieve desired notification performance, Phoenix adaptively controls when a user should take the carrier role and help disseminating a message in order to keep the message “alive,” given the fact that message carriers may leave the target area and drop the message. Extensive simulations have been conducted to show the efficacy of Phoenix notification system.

1. Introduction

In recent years, we have seen rapid growth in the area of location-based services (LBS). Different from LBS based on pull model requiring users to send request to a server, location-based notification (LBN) [1], say LBS based push model, actively alerts users with their interested information when they are in a specific geographical area. This model facilitates retrieving information for mobile users and brings a better user experience. It has been gaining more and more attraction among wireless users and service providers.

LBN has a wide range of applications in automatic parking places searching, traffic jam warning, electronic advertising, and so on. A typical scenario is proximity-based electronic advertisement which is shown in Figure 1, where a shopping mall (or the source) is publishing messages about the time-limited sales promotion and desires to notify all the potential clients in the certain proximity (e.g., 1 km) of the store before the sales activity ends.

The mainstream LBN solution relies on the infrastructure networks (such as cellular networks). In the solution,

the position of a user determined by his mobile device with the GPS module or by the mobile network infrastructure is published continuously to a central tracking server. Then the servers push the matched information to the user according to his registered interests when they enter the specific destination area. However, the centralized solution has a significant limitation; say the service cost is expensive because users' positions need to be published continuously to the server which consumes large amounts of bandwidth. Meanwhile, the available bandwidth per user in a crowded area would be severely limited [2] for cellular technology like UMTS. Thus, the solution in an ad hoc manner may avoid the above drawback and act as a beneficial supplement of the centralized solution to provide LBN services.

In this paper, we develop a system called *Phoenix* to provide the location-based notification service to push information to subscribers. In Phoenix, mobile users across the target area form a mobile peer-to-peer network, where a user may take three possible roles for each message: *source*, *carrier*, and *subscriber*. The source generates the original notification messages with the propagation parameters such

FIGURE 1: Location-based electronic ad application.

as destination propagation region. It initializes the message dissemination procedure by sending it to those who move into the communication range of the source. Those receivers willing to carry and disseminate the message replicas are called carriers. With the movement of message carriers through the area, the message replicas are transmitted to the encountered users according to certain rules, which produce new carriers. When leaving the specified target area, carriers delete the corresponding messages and stop taking the carrier role for those. Based on the above procedure, Phoenix keeps the messages “immortal” over the target area through the peer-to-peer opportunistic message swapping. The users interested in a message are subscribers which specify their interests in the local devices. When a subscriber meets any of the message carriers, it downloads the interested message from the carrier. The source and subscriber roles are predetermined, and carrier role is determined dynamically by Phoenix.

A critical performance metric of a location-based notification service is *notification ratio* which denotes the percentage of subscribers who receive the messages when passing through the area. Additionally, due to the constrained resources in mobile devices, *message overhead* is another important design concern. In this paper, the design goal of Phoenix is to satisfy the user-specified notification ratio while minimizing the transmission overhead of the system. To achieve this goal, we derive the analytic expression of notification ratio of Phoenix and observe that the notification ratio relies on two parameters: the percentage of carriers and the number of mobile users in the destination zone. Then we propose an adaptive message propagation scheme which computes periodically the desired percentage of carriers according to the estimated number of mobile users in the area and then adjusts the message propagation procedure to achieve that percentage.

In this paper, we carried out simulations with the mobility model capturing the realistic characteristics of human mobility and real-world city map to evaluate the system performance. The results show that Phoenix achieves the desired notification ratio with the low message overhead. Moreover, Phoenix is robust to tolerate a nontrivial percentage (20%) of selfish users in the network.

The main advantages of Phoenix include the following: (1) the service cost is very low because the users collaborate to implement the location-based notification and it does not rely on any infrastructure; (2) it is flexible because the notification system is tunable to achieve the tradeoff between the user-specified notification ratio and system overhead.

The rest of this paper is organized as follows. Section 2 reviews the related work. Section 3 describes the system architecture of Phoenix. Section 4 presents the design of the core component in Phoenix. Section 5 discusses the design considerations of Phoenix. Section 6 presents the evaluation results and Section 7 concludes the paper.

2. Related Work

There are plenty of works related to location-based system in the mobile networks. Thus, only the most relevant works are discussed due to the space limit in this paper.

2.1. Location Dependent Data Service in Mobile Networks. A lot of researchers study the location-based services in mobile networks. Chen et al. [3] proposed a method to provide information relevant to location and users subscription using a centralized architecture which works in cellular networks. The centralized solution is far from our work. STEAM [4], LPS [5], and L-ToPSS [6] are the location-based publish/subscribe systems in wireless ad hoc network.

These systems deliver messages only to the interested users which are inside a destination region exactly at publishing time. Autonomous gossiping [7] spreads messages selectively based on vulnerability of nodes instead of flooding the network. By specifying the nodes' profile, it can be used to provide location-based services. Leontiadis and Mascolo [8] presented an opportunistic spatiotemporal dissemination system for vehicular networks. It generates the message replicas dynamically according to the collected subscriptions. However, all of these efforts fail to consider keeping the message circulated in the destination region to notify all interested users across the area.

To the best of our knowledge, the most related work is [9] in terms of maintaining a message over a geographical area to notify the interested users. In order to implement the persistent dissemination, it employs three methods including fixed infostation, ad hoc dissemination, and opportunistic dissemination. However, it did not consider satisfying the expected notification performance.

2.2. Location Aware Routing/Casting in Ad Hoc Networks. This type of research tries to deliver the message to those destination nodes identified/limited by their location in the ad hoc networks. Geocast is the earliest concept to deliver the message to the set of nodes within the specified geographical area in the mobile ad hoc networks (MANET). As a more relevant work, Abiding Geocast [10] aims to deliver time-stable message in a geographical area. It employs periodic epidemic spreading to disseminate the message to a group of users in an area. Comparing with Phoenix, its overhead of route establishment and maintenance caused by the continuous changes in network topology is very expensive. Several researches [11–13] aim to deliver messages to multiple target users in the specific area. Their basic idea is to utilize the variant of constrained epidemic protocols to constrain the propagation of a message within the given area specified by the source. By contrast, researchers [14–16] define the notion of “relevance” to enable the routing layer to self-identify the geographical areas in which the messages should be delivered. In contrast to these works, Phoenix offers much richer semantics in which publishers and subscribers are decoupled as the former define the notification area while the latter express their interests.

3. System Architecture

The proposed system architecture of Phoenix is shown in Figure 2, which consists of four components described as follows.

Application Interface. Application interface provides application developers with two classes of uniform interfaces. Users may exploit *subscribe/unsubscribe* primitive to register/unregister their interested information. All interests are stored in the local repository and not required to pass to other users or central servers. Besides, source users may use *notify* primitive to publish messages. The operation takes four parameters as follows. *MsgArea* is the destination

FIGURE 2: The architecture of Phoenix.

geographical area that a message should reach. Typically, the circular destination area in the aforementioned e-ad example may be denoted as a triple $\langle x, y, r \rangle$ where (x, y) and r are, respectively, the coordinate of the circle center and the radius of the circle. Note that the area may be independent of the current position of the message source. $MsgNR \in (0, 1]$ is the desired notification ratio of the message. This makes Phoenix flexible to support the multiple applications with different notification service requirements. A message is deleted from the current carrier when it lives longer than $MsgDuration$. Here, the absolute time is unnecessary to avoid the strict time synchronization. $MsgBody$ means the message content being propagated.

Peer Detector. Peer detector is responsible for discovering neighboring users in the mutual radio range and executing the handshaking process. After identifying the new neighbor peer by receiving the periodic beacon message, it exchanges further the control information including the list of local interests and the IDs of all carried messages. The information is used by the following message matcher and message controller.

Message Matcher. Message matcher controls which messages should be filtered out and which messages should be transmitted to the encountered users according to their interests. In Phoenix, the content of a message (*MsgBody*) is expressed as a collection of attribute values. For example, a sales message may be represented as `<type = "sales push", brand = "NIKE", discount = "20%", location = "X Market", data = description of the shoes>`. The data item can be text or multimedia such as pictures and videos. The interest is expressed as a combination of these attributes pattern, for example, `<type="...", brand="...">`. When a carried message has the corresponding attribute conditions stated in an interest, a match occurs and the message is transmitted to the peer.

Message Controller. Message controller manages the message propagation to adjust the system performance. Individual message carriers decide whether a carried message should be deleted or copied to the encountered peer which does not hold the corresponding replica. If the message is epidemically

sent to all encountered users, the system achieves the highest notification ratio. However, such system suffers from the high message overhead and the waste of resources and vice versa. Considering its key impact on the system performance, we focus on the design of message controller, in which two main design issues of message controller include the following.

- (i) How to satisfy the notification performance requirements specified by the message?
- (ii) How can we control the message overhead to improve the scalability?

Bearing the design issues in mind, we propose the algorithm of the Message Controller detailed in the next section.

4. Design of Message Controller

In this section, we firstly model the notification ratio and analyze several main impact parameters, then develop an adaptive message propagation algorithm to achieve the design goal.

4.1. Notification Ratio. As stated in Section 3, a subscriber will be notified successfully if it encounters any message carrier. Thus, the notification ratio P is the probability of meeting at least one carrier when a subscriber moves through the target area. Supposing that the message carriers are always distributed uniformly in the zone, the probability that an encountered user is a carrier at a time equals the percentage of carriers to all users in the area. Then the notification ratio is computed as

$$P = 1 - (1 - \rho)^{\beta \times n \times t_s}, \quad (1)$$

where ρ is the percentage of carriers (carrier percentage for short), n is the number of mobile users in the area, t_s is the average time spent staying in the area, and β is the pairwise meeting rate (meeting times in a time unit) between two users. It is shown [17] to follow nearly an exponential distribution under common mobility models (such as the random waypoint or random direction). Also, β can be estimated approximately as follows:

$$\beta = \frac{2wdE[V^*]}{A}, \quad (2)$$

where w is a constant specific to the mobility model (the details can be referred to [17]), d is the transmission range, $E[V^*]$ is the average relative speed between two users, and A is the area of the target region.

4.1.1. Correlation between n and ρ . For most realistic applications, it is reasonable to assume that users' moving speed and transmission range keep constant, while they stay in the area. Thus, we focus on exploring the other two main parameters affecting notification ratio: number of users in the area n and carrier percentage ρ .

n is time varying and not controllable because users autonomously enter and exit the target area continuously.

FIGURE 3: Variation of the number of mobile users in a realistic area.

FIGURE 4: Number of users versus carrier percentage.

However, the statistics of a realistic mobility traces exhibit interesting characteristic that n keeps roughly stable during the peak traffic time. These traces are obtained from a multiagent microscopic traffic simulator (MMTS) which is capable of simulating public and private traffic over real regional road maps of Switzerland with a high level of realism [18]. Figure 3 shows the number of mobile users in around 18 hours of a day in a 3000 m $\times$ 3000 m central region of the Zurich city. During around 3 hours in two peak hours, the number of users varies in a small range (800–1000 users and 220–300 users for the morning and evening peak time, resp.).

We study the correlation between n and ρ under the given notification ratio in a typical scenario, where users with transmission range of 50 m move at the speed of 1 m/s in the circular area with 1 km radius. The parameter w for random waypoint is set to 1.37 according to [17]. Figure 4 depicts their correlation when the desired notification ratios are 0.7, 0.8, and 0.9, respectively. It can be seen that ρ is less sensitive to n

as n increases and exceeds a certain value. For example, ρ may be around 0.13 for the notification ratio of 0.9 when n varies from 150 to 200. From the view point of applications, LBN service is meaningful and commercially feasible only when there are enough users in the destination region. It means that rough estimation of n is enough to determine the value of ρ in order to achieve the desired notification ratio.

4.2. Adaptive Message Propagation Algorithm. The basic idea of our message propagation algorithm is to adjust the carrier percentage according to current number of users in the area to achieve the desired notification ratio. Specifically, the adaptive algorithm takes the following three steps. (1) Estimate continually the number of mobile users in the target area. We assume that the communication range and moving speed of all nodes keep roughly constant. When a user receives a message and becomes its carrier, it uses the target area indicated in the message to compute the pairwise meeting rate β according to (2). Each user keeps a variable x to record the number of the encountered users in a time unit. We have $x = \beta \times n$ according to the definition of β in Section 4.1, so n can be figured out directly. (2) Compute the carrier percentage ρ corresponding to the specified notification ratio ($MsgNR$) in a message according to (1). (3) Adjust the message propagation procedure and drive the system to reach a steady state with the calculated ρ for a message to achieve the desired notification ratio. We introduce a key algorithm parameter $\mathcal{M}$ to control the message propagation procedure, which will be elaborated in the following.

Achieving the desired carrier percentage ρ in a dynamic peer-to-peer network is not trivial. Suppose that the carrier percentage equals ρ at current moment; it will fall down since new users enter and existing carriers leave the area from time to time. In order to keep dynamically the desired ρ , we propose a message propagation procedure which produces new message carriers at the same rate as the leaving rate of existing carriers. Without loss of generality, suppose that current carrier is denoted as i . It runs procedure (see Algorithm 1) while meeting user j , where $\mathcal{D}$ represents the set of the messages carried by i in line 1. When leaving the destination region, the corresponding messages are deleted from the carrier in lines 2–5. In lines 6–15, i keeps a variable C_m for a carried message m , which denotes the number of consecutive noncarriers for message m of i met. When carrier i meets $\mathcal{M}$ noncarrier users consecutively, the carrier generates a message replica for the $\mathcal{M}$ th users and assign it as a new carrier.

$\mathcal{M}$ controls the carrier generation rate. The larger its value is, the more slowly the message carrier is generated. When $\mathcal{M}$ is set to 1, the algorithm produces the message copies at the maximum rate, which is equivalent to the epidemic spreading. Therefore, the key issue is transformed to select the appropriate $\mathcal{M}$ to achieve the desired ρ .

4.3. Parameter Selecting. Next, we will derive the value of $\mathcal{M}$. Let us assume that the target area of message m is a circular area with the radius r , so its area $A = \pi r^2$. n users reside in

Require: receive the carried message identifiers from j

```

(1) for all  $m \in \mathcal{D}$  do
(2)   if  $i$  leave the target area of  $m$  then
(3)      $\mathcal{D} \leftarrow \mathcal{D} - m$ 
(4)   else if  $j$  carries  $m$  then
(5)      $C_m \leftarrow 0$ 
(6)   else
(7)     if  $C_m = \mathcal{M} - 1$  then
(8)       Generating and transferring a new copy of
        $m$  to  $j$  and making  $j$  a carrier
(9)      $C_m \leftarrow 0$ 
(10)  else
(11)     $C_m \leftarrow C_m + 1$ 
(12)  end if
(13) end if
(14) end for

```

ALGORITHM 1: Message propagation procedure.

TABLE 1: Summary of major notations.

r	Radius of target area
A	Area of target area
n	Number of users in the target area
v	Average moving speed
C_m	Variable maintained for message m
d	Transmission range
w	Mobility model parameter
β	Pairwise meeting rate
θ	Movement direction angle to leave the area
ρ	Desired carrier percentage
λ	Rate of users leaving target area
r_p	Rate of producing message copies
r_d	Rate of carrier leaving target area
P_s	Expected notification ratio

the area and move according to an (*iid*) model at the average speed v . These assumptions are used to derive the theoretical results, but our solution does not rely on them, which will be verified in the evaluation results. Nodal transmission range is denoted by d . The symbols and notations used in the model are summarized in Table 1.

4.3.1. Carrier Leaving Rate. We first compute the rate at which users leave the target area. As shown in Figure 5, the users which will leave probably the area at the next moment must be those located in the shaded annular area defined by a radius $r - v$ and a radius r at the current moment. Any user u located in the shaded area at the current moment will leave the target area at the next moment only if it chooses the movement

FIGURE 5: The calculation of users leaving rate.

direction within the angle θ . Under uniform mobility model, users' leaving rate is calculated by

$$\begin{aligned} \lambda &= \int_{r-v}^r \frac{n}{A} 2\pi x dx \frac{\theta}{2\pi} \\ &= \frac{2n}{A} \int_{r-v}^r \left(\pi - a \cos \left(\frac{v^2 + x^2 - r^2}{2vx} \right) \right) x dx. \end{aligned} \quad (3)$$

A portion of users in the area are carriers, so the carrier leaving rate is calculated by

$$r_d = \lambda \rho = \frac{2n\rho}{A} \int_{r-v}^r \left(\pi - a \cos \left(\frac{v^2 + x^2 - r^2}{2vx} \right) \right) x dx. \quad (4)$$

4.3.2. Carrier Generating Rate. We use variable C_m to track the number of consecutive noncarriers the carrier has met. Upon meeting $\mathcal{M}$ consecutive noncarriers, the carrier generates a message replica and then resets C_m to 0. Thus, the variable can take any value from 0 to $\mathcal{M} - 1$. It can be represented in a Markov process illustrated in Figure 6. For any contact at steady state, the probability of meeting a noncarrier and increasing C_m is $1 - \rho$. The probability of meeting a carrier and resetting C_m is ρ .

Thus, the transit probability matrix of the Markov process is represented by

$$\mathbf{P} = \begin{pmatrix} \rho & 1-\rho & 0 & \dots & 0 \\ \rho & 0 & 1-\rho & \dots & 0 \\ \rho & 0 & 0 & 1-\rho & \dots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & 0 & \dots & 0 \end{pmatrix}. \quad (5)$$

The matrix is ergodic and has a steady distribution $\alpha = \alpha P$. Thus, α is an eigenvector of P with eigenvalue of exactly 1. Here we are interested in the probability of residing the $\mathcal{M} - 1$ state, which equals

$$\alpha_{\mathcal{M}-1} = \frac{\rho(1-\rho)^{\mathcal{M}-1}}{1 - (1-\rho)^{\mathcal{M}}}. \quad (6)$$

FIGURE 6: Markov process for message production.

Therefore, the expected rate at which the message carriers are generated is

$$r_p = (\rho n) n\beta (1-\rho) \alpha_{\mathcal{M}-1}, \quad (7)$$

where β is the pairwise meeting rate.

4.3.3. Computation of $\mathcal{M}$. Let the carrier generating rate equal the carrier leaving rate; say $r_p = r_d$ and then substitute (6); we obtain the close-form solution of $\mathcal{M}$:

$$\begin{aligned} &(\rho n) n\beta (1-\rho) \alpha_{\mathcal{M}-1} \\ &= \frac{2n\rho}{A} \int_{r-v}^r \left(\pi - a \cos \left(\frac{v^2 + x^2 - r^2}{2vx} \right) \right) x dx \Rightarrow \mathcal{M} \\ &= -\log_{1-\rho} \\ &\quad \times \left(1 + \rho \frac{nA\beta}{2 \int_{r-v}^r (\pi - a \cos ((v^2 + x^2 - r^2)/2vx)) x dx} \right). \end{aligned} \quad (8)$$

We carry out a simulation to evaluate the accuracy of this model. In the designed scenario with the same network parameters in the Section 4, all users' mobility follows random waypoint model. We firstly compute the $\mathcal{M}$ corresponding to the desired ρ according to (8). Then we run a simulation program to measure the realistic carrier percentage when parameter $\mathcal{M}$ varies. As shown in Figure 7, the closeness between the theoretical and measured results verifies the accuracy of the model. Besides, it shows further that smaller $\mathcal{M}$ should be set to achieve the higher carrier percentage.

4.4. Message Overhead Analysis. Message overhead is caused mainly by exchanging the information messages (messages for short) and control messages including the interest list and the carried message list. In most applications, the size of control messages is much smaller than the information messages, so the overhead analysis focuses on the latter, that is, the overhead caused by maintaining the desired carrier percentage. According to our algorithm, the message replica should be generated continuously to make new carriers in order to achieve the expected notification performance. Specifically, each carrier produces and transmits a message replica only when it meets $\mathcal{M}$ noncarrier users consecutively. Thus, the transmission overhead in a time unit equals $(\rho n) n\beta (1-\rho) \alpha_{\mathcal{M}-1} \times s_m$ where s_m is the size of the information message.

FIGURE 7: Comparison between theoretical and measured results.

5. Discussions

5.1. System Feasibility. The theoretical analysis of our paper assumes that all users are willing to be message carriers. In reality, users may be selfish or not collaborative. The following strategies will make our system feasible in real applications. First, we use an incentive scheme based on economic model or game theory [19, 20]. Let us take the electronic ads as an example. An available business solution is to let the mall advertisers add a certain amount of money-saving e-coupon messages to the normal ads messages in order to attract volunteer users. Second, the service is almost free of charge except the low battery and storage resources consumption which are verified in the following evaluation. Third, the feasibility is further improved due to its robustness to tolerate a percentage of selfish users, which will be illustrated in the next section.

5.2. Distributed Design. In our design, the message notification system does not rely on the centralized server to exchange information. All users are autonomous and able to control their own data and mobility. The proposed self-controlled and adaptive approach to determine the carriers makes the system flexible and scalable. However, the performance of the fully distributed system degrades probably at several extreme situations, for example, if the number of users in the target area is so small that existing carriers have no opportunities to send a message replica to the encountered users before they leave the area. The message probably disappears in the area and the future entering subscribers fail to be notified. Thus, in the real applications, the user may place at least a fixed message source in the target area in order to guarantee the performance when the number of users is too small.

5.3. Optimization. In current solution, subscribers receive the message of interest only from the encountered carriers.

The performance can be further optimized if subscribers that have received the message are permitted to act as additional *carriers* for the message in the circulation. In addition, when a carrier is transmitting a message replica to another user and assign it as a new carrier, other users in the transmission range can receive the message and store it locally. The two types of users do not need to behave like the real carriers to exchange information to keep the dynamic percentage. In contrast, they play a role of the moving message cache to help in delivering messages to the encountered subscribers, which improve the notification performance. If no subscriber is encountered, the optimization methods do not cause additional network overhead.

6. Performance Evaluation

In this section, we evaluate the performance of Phoenix through simulation in the realistic city map as well as the power-law mobility model. We first study the spatial distribution of the carrier users and carrier percentage distribution, and then we explore the impact of users' number on notification performance. Finally, we also examine the effect of selfish users on system performance.

6.1. Simulation Setup. We implemented Phoenix in the opportunistic network environment (ONE) [21] simulator. To simulate the realistic scenarios, we need large scale data. However, the large scale realistic mobility trace based on absolute position is not available. To the best of our knowledge, the publicly available large scale human trace is the Dartmouth College trace collected through the Wi-Fi access points covering the entire campus. It includes the mobility traces of 6300 users for a long period. However, the movement of all users is only limited on the campus and the position of each user is fairly coarse because the trace of each user is represented by a series of the location of access points associated with the user. Besides, an access point covers several hundred meters distance and users are not always connected to the closest access point. Thus, the trace is not suitable to study the performance of location-based notification systems. In the evaluation, we make use of a realistic city map in which the users move along roads and walkways. Meanwhile, their movement follows a power-law mobility model which is elaborated below.

The realistic city area that we selected is 8500 m × 7500 m Helsinki city's central area which is illustrated in Figure 8. Two hundred points of interests (POIs) are deployed randomly on the map to model popular locations, for example, shops, restaurants, and tourist attractions. We use the similar algorithm proposed in the literature [22] to generate the mobility trace. The experiment results from [22] have shown that the generated mobility trace contains those similar statistical features including the truncated power-law flights, pause time, and intercontact time, which are observed in the real gathered human mobility trace over various scales of time and space.

In this mobility model, each user is deployed randomly at a POI. For a user located at the current POI i ,

FIGURE 8: The target area on the Helsinki city's central area.

the probability that a next POI j is chosen is computed as $(1/d_{ij}^\phi)/(\sum_{k \in V-V'} 1/d_{ik}^\phi)$ where d_{ij} is the Euclidean distance from i to j , ϕ is a fixed constant within 0 to infinity, and V and V' are, respectively, the set of all the POIs and visited POIs so far. Based on this probability, the next POI is chosen from unvisited POIs. Specifically, if ϕ is infinite, a user always chooses the nearest unvisited POI and if it is zero, a user randomly chooses the destination point. Then the user calculates the shortest path to the next POI using Dijkstra's algorithm and moves to the destination following the calculated path. Then we use a truncated power-law pause time distribution with parameter ϕ to decide the amount time to stay at each POI. The selected target area is located in the downtown area of Helsinki with a radius of 1 km, as shown in Figure 8. The message source is fixed in the center point of the area with coordinates (5200, 5000). The source generates 10 kinds of information messages to notify the interested users passing through the area. There are different proportions of users subscribing to these information messages. The information message packet size is 1024 bytes and has the same lower bound of notification ratio 80%. The simulation lasts for 5 hours (18000 seconds), and results are averaged over the achieved notification performance of all messages. Unless otherwise stated, we use the default simulation parameters listed in Table 2.

6.2. Spatial Distribution of Carriers. The design of Phoenix relies on the premise that the probability of a user encountering a message carrier is equal to the global carrier percentage. Its validity depends on whether the carrier users distribute uniformly throughout the network. Thus, we firstly explore whether the condition is met under the power-law mobility model. Figure 9 illustrates the spatial distribution of carrier users in the target area through the entire simulation. The red users are carrier users; the gray users are noncarriers. As expected, the message carriers are nearly uniformly distributed throughout the target area for the most of the simulation duration.

TABLE 2: Default simulation parameters.

Parameter	Value
Size of network area	$8500 \times 7500 \text{ m}^2$
Radius of target area	1000 m
Number of POIs	200
Transmission range	150 m
Total number of users	1000
Control message size	80 bytes
Information message size	1024 bytes
Average nodal speed	1 m/s
Lower bound of notification ratio	80%
Parameter ϕ	1.5
Power exponent for pause time ϕ	1

6.3. Time Stability of Carrier Percentage. After the system reaches the steady state, we need to maintain the carrier percentage around an expected level in order to achieve the desired notification performance. Figure 10 depicts the variation of carrier percentage with time in the scenario with different number of users when $\mathcal{M} = 2$ and $\mathcal{M} = 5$, respectively.

It can be observed that the carrier percentage keep nearly stable after a short initialization phase. As expected, different carrier percentage levels are achieved for the different $\mathcal{M}$. Besides, the higher the number of users is, the more stable the kept carrier percentage is. It is because the number of users residing in the target area is little and considerably fluctuates in the low density scenario. Thus, the generating and leaving of carriers result in the drastic change of carrier percentage.

6.4. The Achieved Notification Ratio and Message Overhead. We also compare the notification performance in terms of achieved notification ratio and message overhead between Phoenix and Epidemic which is a simple version of our algorithm where $\mathcal{M} = 1$ all the time. The expected lower bound of notification ratio is set to 80%. The number of mobile users in the scenario varies from 500 to 2500 at a step of 250, and our adaptive message propagation algorithm recomputes the desired carrier percentage according to (1) and then adjusts the parameter $\mathcal{M}$. We measure the realistic notification ratios for each user density. Besides, we also use the total number of the messages transmitted to keep the desired carrier percentage by all carriers as the message overhead.

Figure 11(a) compares the achieved notification ratio between Phoenix and Epidemic. When the number of users is less than 1000, they have the same performance because our adaptive algorithm chooses $\mathcal{M} = 1$ for low density users. In this cast, they cannot satisfy the expected lower bound 80% of the notification ratio. It is because notification ratio relies on both carrier percentage and other factors such as user number and transmission range. In the highly sparse network, subscribers can probably not be notified because of failing to meet any message carrier although the carrier percentage is 100%. Therefore, the enough user density is necessary to ensure the desired performance. When

FIGURE 9: Spatial distribution of message carriers.

there are over 1000 users in the area, Phoenix achieves the better agreement with the desired performance requirement than Epidemic because of its adaptive message propagation scheme.

As demonstrated in Figure 11(b), our algorithm brings much lower message overhead than Epidemic scheme. Moreover, the more the number of users is, the better our algorithm performs. It can be explained below. In our algorithm, the rate of message generation is adjusted adaptively to the optimal level by observing the variation of the user density in the area, which means more stable transmission overhead and better scalability. In contrast, Epidemic makes a message replica for each encountered user, resulting in that the message overhead increases rapidly as the user density rises. When the number of mobile users in the area becomes large enough, Epidemic might even fail to work because of the transmission collision.

6.5. Against Selfish Users. We evaluate the robustness of the system against selfish users, which only receive the messages and refuse to spread the received messages. The proportion of selfish users varies from 10% to 50% in two scenarios with 1000 and 2000 users, respectively. We record the achieved notification ratios of 10 runs with different random seeds.

As shown in Figure 12, it is observed that the achieved notification ratio falls with the increase of the selfish users. When the proportion of selfish users is smaller than 20%, it hardly has the obvious effect on the success ratio, it hardly place the obvious impact on the success ratio which always stay around the expected 80%. However, the notification performance decreases significantly when the selfish user proportion reaches 40% and basically fails to satisfy the desired performance. It is because the number of the encountered users by the subscribers passing through the target

FIGURE 10: Variation of carrier percentage for the desired notification ratio 80%.

FIGURE 11: Phoenix versus Epidemic.

area is approximately constant in a given network. The more the selfish users are, the lower the probability of meeting a carrier for a subscriber accordingly is. The results illustrate that Phoenix is robust to the certain proportion of selfish users.

7. Conclusion

In this paper, we design a peer-to-peer location-based notification system, Phoenix. We focus on the design of the key component message controller. Message controller aims to satisfy the expected lower bound of notification ratio

while minimizing the transmission overhead. For this target, we firstly model the notification ratio and then propose an adaptive algorithm which adjusts the message propagation procedure to achieve the desired notification performance by observing the number of mobile users in the destination region. We also discuss some other design problems such as the system feasibility, distributed design, and performance optimization. Finally, we evaluate the system performance in the simulation with a realistic city map and power-law movement model. The results show that Phoenix can satisfy the user-specified notification performance with the low transmission overhead. Moreover, Phoenix is robust to the certain proportion of selfish users in the network.

FIGURE 12: Impact of selfish users when the user specified is 80%.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (Grant no. 61202436 and Grant no. 61271041) and State Key Development Program for Basic Research of China (Grant no. 2011CB302902).

References

- [1] J. P. Munson and V. K. Gupta, "Location-based notification as a general-purpose service," in *Proceedings of the 2nd International Workshop on Mobile Commerce (WMC '02)*, pp. 40–44, September 2002.
- [2] A. Chief, "Operators aren't prepared for onslaught of data traffic," 2009, <http://tinyurl.com/at-t-chief>.
- [3] X. Chen, Y. Chen, and F. Rao, "An efficient spatial publish/subscribe system for intelligent location-based services," in *Proceedings of the 2nd International Workshop on Distributed Event-Based Systems*, pp. 1–6, September 2003.
- [4] R. Meier and V. Cahill, "STEAM: event-based middleware for wireless ad hoc networks," in *Proceedings of the 22nd International Conference on Distributed Computing Systems Workshops*, pp. 639–644, July 2002.
- [5] P. T. Eugster, B. Garbinato, and A. Holzer, "Location-based publish/subscribe," in *Proceedings of the 4th IEEE International Symposium on Network Computing and Applications (NCA '05)*, pp. 279–282, Cambridge, Mass, USA, July 2005.
- [6] I. Burcea and H. Jacobsen, "L-ToPSS-push-oriented location-based services," in *Proceedings of the 4th VLDB Workshop on Technologies for E-Services (TES '03)*, 2003.
- [7] A. Datta, S. Quarteroni, and K. Aberer, "Autonomous gossiping: a self-organizing epidemic algorithm for selective information dissemination in wireless mobile ad-hoc networks," in *Semantics of a Networked World. Semantics for Grid Databases*, Lecture Notes in Computer Science, pp. 126–143, Springer, Berlin, Germany, 2004.
- [8] I. Leontiadis and C. Mascolo, "Opportunistic spatio-temporal dissemination system for vehicular networks," in *Proceedings of the 1st International MobiSys Workshop on Mobile Opportunistic Networking*, pp. 39–46, July 2007.
- [9] I. Leontiadis, P. Costa, and C. Mascolo, "Persistent content-based information dissemination in hybrid vehicular networks," in *Proceedings of the IEEE International Conference on Pervasive Computing and Communications (PerCom '09)*, pp. 1–10, Galveston, Tex, USA, March 2009.
- [10] C. Maihöfer, T. Leinmüller, and E. Schoch, "Abiding geocast: time-stable geocast for ad hoc networks," in *Proceedings of the 2nd ACM International Workshop on Vehicular Ad Hoc Networks*, pp. 20–29, April 2005.
- [11] S. Eichler, C. Schroth, T. Kosch, and M. Strassberger, "Strategies for context-adaptive message dissemination in vehicular ad hoc networks," in *Proceedings of the 3rd Annual International Conference on Mobile and Ubiquitous Systems*, pp. 1–9, San Jose, Calif, USA, July 2006.
- [12] P. Costa, D. Frey, M. Migliavacca, and L. Mottola, "Towards lightweight information dissemination in inter-vehicular networks," in *Proceedings of the 3rd ACM International Workshop on Vehicular Ad Hoc Networks (VANET '06)*, pp. 20–29, April 2006.
- [13] G. Korkmaz, F. Özgüner, E. Ekici, and Ü. Özgüner, "Urban multi-hop broadcast protocol for inter-vehicle communication systems," in *Proceedings of the 1st ACM International Workshop on Vehicular Ad Hoc Networks (VANET '04)*, pp. 76–85, Philadelphia, Pa, USA, October 2004.
- [14] C. Adler, R. Eigner, C. Schroth, and M. Strassberger, "Context-adaptive information dissemination in vanets—maximizing the global benefit," in *Proceedings of the 5th IASTED International*

- Conference on Communication Systems and Networks (CSN '06)*, pp. 7–12, Palma de Mallorca, Spain, August 2006.
- [15] M. Caliskan, D. Graupner, and M. Mauve, “Decentralized discovery of free parking places,” in *Proceedings of the 3rd International Workshop on Vehicular Ad Hoc Networks (VANET '06)*, pp. 30–39, September 2006.
 - [16] T. Kosch, C. Schwingenschlogl, and L. Ai, “Information dissemination in multihop inter-vehicle networks,” in *Proceedings of the IEEE 5th International Conference on Intelligent Transportation Systems*, pp. 685–690, August 2002.
 - [17] R. Groenevelt, P. Nain, and G. Koole, “The message delay in mobile ad hoc networks,” *Performance Evaluation*, vol. 62, no. 1–4, pp. 210–228, 2005.
 - [18] V. Naumov, R. Baumann, and T. Gross, “An evaluation of inter-vehicle ad hoc networks based on realistic vehicular traces,” in *Proceedings of the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC '06)*, pp. 22–25, May 2006.
 - [19] L. Buttyán and J.-P. Hubaux, “Stimulating cooperation in self-organizing mobile ad hoc networks,” *Mobile Networks and Applications*, vol. 8, no. 5, pp. 579–592, 2003.
 - [20] P. Antoniadis, C. Courcoubetis, and R. Mason, “Comparing economic incentives in peer-to-peer networks,” *Computer Networks*, vol. 46, no. 1, pp. 133–146, 2004.
 - [21] A. Keranen, J. Ott, and T. Karkkainen, “The ONE simulator for DTN protocol evaluation,” in *Proceedings of the 2nd International Conference on Simulation Tools and Techniques*, pp. 1–10, April 2009.
 - [22] K. Lee, S. Hong, S. J. Kim, I. Rhee, and S. Chong, “SLAW: a mobility model for human walks,” in *Proceedings of the 28th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '09)*, pp. 855–863, Rio de Janeiro, Brazil, April 2009.