

Time and Energy Efficient Localization

Wei Cheng^{*}, Jindan Zhu[†], Prasant Mohapatra[†], Jie Wang[‡]

^{*}Department of Computer Science, Virginia Commonwealth University, USA

[†]Department of Computer Science, University of California Davis, USA

[‡]Department of Computer Science, University of Massachusetts Lowell, USA

Email: wcheng3@vcu.edu, jdzh@ucdavis.edu, prasant@cs.ucdavis.edu, wang@cs.uml.edu

Abstract—Time-critical Location Based Service (LBS) applications in mobile ad hoc networks require fast localization. The conventional localization techniques are, unfortunately, unsuitable for such applications, for they neglect the time needed for localization. As a result, time-critical information may become obsolete, and the mobile users such as vehicles may have moved to new locations before the localization procedure is completed. To address this issue, we formulate a notion of *On-Demand Fast Localization* (ODFL) and devise a framework to implement this concept over existing routing protocols in MANETs. We present analytical and simulation results to demonstrate that ODFL can significantly reduce the time solely needed for localization before starting time-critical applications. Moreover, we show that ODFL can also improve location privacy and reduce energy consumptions.

I. INTRODUCTION

The success of an LBS (Location Based Service) application depends on two pipelined procedures: localization of a device and information transmission. The localization problem has been extensively studied under a number of contexts such as wireless sensor networks, MANETs, and 802.11 wireless networks [5], [7], [8], [12], [13]. Under the conventional localization definition, localizing both the source and the destination devices in the same coordinate system is a necessary condition to start the information transmission of an LBS application.

The conventional localization procedure, initiated from a few anchors, propagates the localization information to all users via flooding. As a result, the localization time is bounded below roughly by $T_{min}(r_h, t_1, k)$ defined in the following formula:

$$T_{min}(r_h, t_1, k) = (k \times r_h) \times (t_1 + \Delta_D), \quad (1)$$

where r_h is the radius of the network in the form of the number of hops, t_1 the time needed for localizing one user (including the processes of direct neighborhood discovery, distance measurement, and trilateration), k a parameter related to the positions of anchors, and Δ_D the average transmission delay at each user.

For a typical network setup with three anchors deployed in the center of a network, we have $k = 1$, where $T_{min}(r_h, t_1, 1)$ is the minimum time for transmitting a message from a user in the center to a user

on the border. The localization time $T_{min}(r_h, t_1, 1)$ is also the lower bound for relative localization, where each user starts localization by constructing a Local Coordinate System (LCS) itself. The reason is that the conventional localization definition requires that all the users be localized in the same coordinate system. In other words, the positions of the users under their own LCSs need to be transformed to the global coordinate system, and the position transformations also need to propagate information across the entire network.

Consequently, an LBS application has to wait for at least T_{min} time before transmitting information. While T_{min} may be suitable for time-critical LBS applications in static networks, where the localization procedure only needs to be conducted once, it may not be good for those in MANETs, whose network topology may change dramatically during the T_{min} time frame, preventing the conventional localization procedure from localizing all users in the same coordinate system. As a result, LBS applications could fail because information transmission cannot start, which may also cause excessive amount of energy consumption because the localization process usually keeps trying to localize all the users in a network. Even if the source can eventually obtain a position of the destination, this position may be an old one that is far away from the current position of the destination. Therefore, the conventional localization techniques cannot be directly applied to time-critical LBS applications in MANETs.

Deploying a large number of anchors is a simple solution, with the purpose that every spot in the network can be covered by 3 non-collinear (4 non-coplanar) anchors for 2D (3D) localization. Another solution is to equip every user with a GPS receiver. These two solutions, however, may not be practical for the following reasons: (1) Deploying a large number of anchors may be impossible, or uneconomical even if deployment is not an issue. (2) The GPS satellites may be destroyed or interfered by the opponents in a battle field. (3) Not all the users always turn on their GPS receivers.

It is therefore necessary to devise an infrastructure-free fast localization scheme for MANETs. Moreover, under the conventional localization definition, the in-

formation source has no position privacy, as all users can overhear LBS messages and learn the positions of embedded sources. To address these issues, we formulate a novel concept of *On-Demand Fast Localization* (ODFL) from a new perspective to start LBS applications at the earliest time. In ODFL, we carry out the conventional pipelined procedures of localization and information transmission via two semi-concurrent procedures, so that we can start LBS applications before the destinations are localized. The contributions of this paper are summarized below:

- 1) We initiate a new direction of fast localization by integrating the localization procedure with routing to support time-critical LBS applications in MANETs.
- 2) We investigate the definition of localization and formulate a novel concept of On-Demand Fast Localization, which only involves the users that are necessary for location transformation along the routing path.
- 3) We devise a framework of ODFL that can be implemented over most routing protocols. ODFL can significantly reduce the localization time and the energy consumption.
- 4) We justify, through both analysis and simulations, the superiority of ODFL in terms of shorter localization time, lower energy consumption, higher LBS success ratio, and improved position privacy.

The rest of the paper is organized as follows. In Section II we introduce the network model and present sample applications. In Section III, we formulate the concept of On-Demand Fast Localization and devise an ODFL framework. In Section IV we analyze the performance of the ODFL framework. In Section V we evaluate its performance via simulations. We briefly review the related work in Section VI, and conclude the paper in Section VII.

II. GENERAL MODEL AND APPLICATION EXAMPLES

A. General Model

We consider LBS applications in a general network model, where there exist a small number of base stations and a large number of mobile users. Using multi-hop routing, the mobile users send messages containing their positions to the destinations, which could be base stations or other users. The success of message delivery relies on the adopted routing protocols, the user mobility, and the channel conditions, which will not be discussed in this paper. For our purpose, we simply assume that the source nodes can always send messages successfully to the destinations using an existing routing protocol.

The ultimate goal of localization is to support LBS applications in MANETs. We consider the *success of*

LBS applications as that the destinations can understand the source positions under the same coordinate system after receiving the messages.

B. Application Examples

The general model can be applied to time-critical LBS applications such as military reconnaissance in battle fields and road emergent information sharing in vehicular networks. Specifically, in the military reconnaissance application, scouts (or military robots) are the mobile users who may move at high speeds in a battle field. They share collected information with each other and report it to the command stations (base stations). The transmitted information contains both the military information and the positions from which the information is collected. As scouts may reconnaissance behind enemy lines, fast localization and immediate information transmission are invaluable for their safety and the military decision analysis. Similarly, for vehicular network applications, a vehicle (mobile user) needs to share the road emergent information with other vehicles or road-side stations (base stations) that may be several miles (multi hops) away from it. As the network topology is highly dynamic, it is critical to localize emergent events instantly and send the information out as soon as possible.

III. ON-DEMAND FAST LOCALIZATION

A. Motivation

The fundamental flaw in conventional approach is that it requires the information source and destination to be localized under the same coordinate system before any information transmission can be initiated. This serializing approach could seriously delay the starting time of supported LBS, which as to be discussed later, might cause unacceptable repercussion to quality of service in terms of localization accuracy in a highly dynamic mobile network. Furthermore, conventional approach relies on flooding in the network to disseminate location information. This demands all nodes in the network to participate in the location updating and forwarding process, even when a large portion of them are irrelevant and far away from the actual information transmission route. In a mobile network this process would have to be performed constantly. As a result, the overall communication overhead will be significantly increased while energy efficiency will be decreased.

After all the ultimate objective of an LBS application is to allow the destination to successfully decode the source position (embedded in the messages) in its own coordinate system. This objective can be achieved by creating local coordinate systems (LCS) at each node and performing local coordinate system position transformation between consecutive nodes along the route.

Take scenario illustrated in Fig. 1 as an example, where node 0 needs to transmit a location-aware message to node $2h$ which is $2h$ -hops away. We denote node 0 as source and node $2h$ as destination. Location of source node 0 in this case is also the source location of event. Dashed circle represents communication range of nodes, and nodes resided within one's communication range are one's 1-hop neighbor.

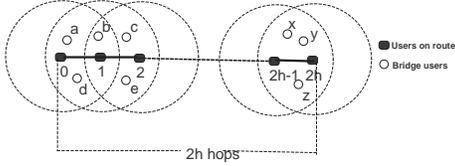


Fig. 1. ODFL example

Without waiting for $T_{min}(h, t_1, 1)$ for location information to propagate from anchors, source can start immediately after 2-hop topology is acquired in $T_{min}(1, t_1, 1)$, by actively localizing neighbors in its own LCS, LCS_0 . LCS is created and represented by choosing three non-collinear interconnected nodes from the neighbors. Other neighbors or the source itself can be subsequently localized in the LCS if necessary ranging information is available. Following the same procedure, the next hop node (node 1) can also create LCS_1 and localize some of its neighbors. Position of localized node, including the source, in one LCS can be transformed into a position in the next LCS, if more than three nodes can be localized in both LCSs. These commonly localized nodes are denoted as *bridge nodes*, and their positions will be embedded in the message together with source position and passed on to next hop to enable the LCS position transformation. In this example, source will localize positions of node a , b , and d , and embedded them as bridge nodes with position of node 0. Upon receiving the message, node 1 can check if node a , b , and d are also localized in LCS_1 . If so source position will be transformed to position in LCS_1 . Such process will continue hop by hop until the message reaches node $2h$, for which the source position is transformed from LCS_{2h-1} to LCS_{2h} . To this end, node $2h$ can understand the position of node 0 under its own LCS: the ultimate objective is achieved.

B. On-Demand Fast Localization

From the example described in the previous section, we know neither user 0 nor user $2h$ can be localized in the same coordinate system after $T_{min}(1, t_1, 1)$ time. Nevertheless, all future steps needed for localization transformation can be embedded into the procedure of information transmission, and the destination can successfully obtain the position of the source in its LCS after receiving the message. We can see that part of the

localization process can be seamlessly integrated into the subsequent information transmissions, incurring limited extra overhead. Based on this observation, we define the concept of *On-Demand Fast Localization* in Def. 3.1.

Definition 3.1: (On-Demand Fast Localization) The localization starts from multi-coordinate systems, and the position transformation is integrated with LBS applications information transmissions.

In particular, ODFL captures the earliest time for starting LBS applications. Each intermediate user on the route can help transforming the information of the position of the source to the one in its next-hop user's coordinate system until the destination successfully obtains the position of the source in its own coordinate system. Based on this per hop location transformation method, LBS applications can start right after the information source localizing itself in its own coordinate system. In other words, information transmissions of LBS applications can start before all the users are localized in the same coordinate system. The start of the second procedure in the original pipeline no longer depends on the success of the first procedure.

C. Framework Design

We devise a framework of ODFL from a different angle of view, which turns the two pipelined procedures into two semi-concurrent procedures. It employs coordinate system transformation on a per hop basis, which consists of three components: (1) LCS construction; (2) embedded information selection; and (3) position transformation.

1) *LCS construction*: The first component of ODFL is to construct a local coordinate system at each user on the information route so that the bridge users can be localized. The procedure of LCS construction is composed of two steps: (1) local information collection and (2) LCS construction.

In Step (1), each user on the routing path first broadcasts an information-request message to its direct neighbors. Each of the user's neighbors replies with the information of its one-hop topology, which can be obtained via round trip communications for neighbor discovery.

For the users on the information transmission route, they collect information in a chain procedure. In other words, each intermediate user initiates its information collection only after receiving the previous user's information request message. Each intermediate user can derive its two-hop topology for LCS construction as shown in Fig. 2. Note that there are no edges between two-hop neighbors.

An LCS can be constructed by finding three mutually connected users (such as users a , b , and X) to form a coordinate system, which is denoted by its *LCS ID* (such as $\langle a, b, X \rangle$). Multiple LCSs may exist

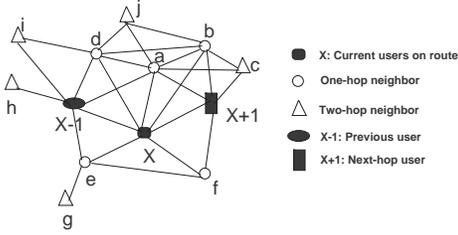


Fig. 2. A user's two-hop topology for LCS construction

in the two-hop topology of the user. The set of users that can be localized in each LCS can be obtained via trilateration in $O(\beta^6)$ time, where β is the maximal node degree in the user's two-hop topology. Then, these LCSs can be merged into a number of candidate LCSs (denoted by CLCSs) according to the condition of coordinate transformation. Therefore, the current user in Fig. 2 can have a few CLCSs and the users that can be localized in these CLCSs after Step (2). Note that some users may be localized in more than one CLCSs, and that any two CLCSs cannot localize more than 2 common users as they are not mutually transformable. For the example in Fig. 2, the current user can have five CLCSs: $\langle a, b, X \rangle$, $\langle d, i, X - 1 \rangle$, $\langle e, X - 1, X \rangle$, $\langle e, f, X \rangle$, and $\langle f, X, X + 1 \rangle$. Users $a, b, c, d, j, X - 1, X$, and $X + 1$ are local CLCS $\langle a, b, X \rangle$; users d, i , and $X - 1$ are localized in $\langle d, i, X - 1 \rangle$; users $e, X - 1$, and X are localized in $\langle e, X - 1, X \rangle$; users e, f , and X are localized in $\langle e, f, X \rangle$; users f, X , and $X + 1$ are localized in $\langle f, X, X + 1 \rangle$.

2) *Embedded information selection*: The objective of this component is to embed appropriate information into the message at each hop to transform source position on a per-hop basis. Suppose that the current users can obtain the corresponding users of the source in their CLCSs. They embed the information as shown in Fig. 3 into the message for helping the next-hop user with position transformation. Based on the embedded information, the next-hop user can transform the source positions to the corresponding ones in its CLCSs when three or more bridge users in a row of Fig. 3 can be localized in one of its CLCSs. Note that the next-hop user may obtain the source's corresponding positions in multiple CLCSs.

Source Position in CLCS1	CLCS1 ID	Bridge users' positions in CLCS1
Source Position in CLCS2	CLCS2 ID	Bridge users' positions in CLCS2
Source Position in CLCS3	CLCS3 ID	Bridge users' positions in CLCS3
.....

Fig. 3. Embedded information format

In general, each user on the information transmission route is ready for the position transformation after LCS

is constructed. As there is no edge between any 2 two-hop neighbors, any of the current user's two-hop neighbors may also appear in the next-hop user's two-hop topology. Moreover, the current user cannot identify the users that cannot be localized by the next-hop user because it does not know the next-hop user's two-hop topology. As a result, every localized user could be a bridge user. In order to maximize the probability of position transformation, we conservatively choose to embed all the users' positions in the CLCSs, where the source and at least three non-collinear bridge users can be localized, into the message.

3) *Position transformation*: Upon receiving the embedded information from the previous user, the current user transforms the position of the source to the corresponding one in its CLCS through solving (x, y) from Eqs. (2), where (x, y) and (x', y') are the coordinates of the source in the current user's CLCS and in the previous user's CLCS, respectively; (x_1, y_1) , (x_2, y_2) , and (x_3, y_3) are the positions of the three bridge users in the current user's CLCS; (x'_1, y'_1) , (x'_2, y'_2) , and (x'_3, y'_3) are the bridge users' corresponding positions in the previous user's CLCS; Note that x and y are the only unknowns in Eqs. (2) as the current user can obtain (x', y') , (x'_1, y'_1) , (x'_2, y'_2) , and (x'_3, y'_3) from the previous user's embedded information, and it can also acquire (x_1, y_1) , (x_2, y_2) , and (x_3, y_3) after its LCS construction procedure. The solution is unique as long as the three bridge users are non-collinear.

$$\begin{cases} (x - x_1)^2 + (y - y_1)^2 = (x' - x'_1)^2 + (y' - y'_1)^2 \\ (x - x_2)^2 + (y - y_2)^2 = (x' - x'_2)^2 + (y' - y'_2)^2 \\ (x - x_3)^2 + (y - y_3)^2 = (x' - x'_3)^2 + (y' - y'_3)^2 \end{cases} \quad (2)$$

Each user reacts based on the events defined in Tab. I. The pseudocode executing at each user is presented in the following:

Event Name	Description
<i>Event_LBS_Ini</i>	Initiate LBS message transmissions.
<i>Event_Rev_Info_Req</i>	Receive an information request.
<i>Event_Rev_Neb_Dis</i>	Receive a neighbor discovery.
<i>Event_Rev_LBS_Pkt</i>	Receive a packet containing LBS.

TABLE I
EVENT DEFINITIONS

Framework of ODFL

Handling(*Event_LBS_Ini*)

- 1: Construct LCS;
- 2: Embed information into the LBS message;

Handling(*Event_Rev_Info_Req*)

- 3: Broadcast a neighbor discovery message;
- 4: Wait for Δ_{t1} time for building its one-hop topology;

5: Send its one-hop topology to the user who sent the information request message;
6: **if** The user itself is the next-hop user **then**
7: Construct LCS;
8: **end if**

Handling(*Event_Rev_Neb_Dis*)

9: Send itself's user ID to the user who sent the neighbor discovery message;

Handling(*Event_Rev_LBS_Pkt*)

10: Wait until its CLCSs have been constructed;
11: **if** the user itself is the destination **then**
12: Remove the CLCSs in which the position of the user itself is not available;
13: **end if**
14: **for** each row in the embedded information (Fig. 3) **do**
15: **for** each CLCS **do**
16: **if** the positions of at least three non-collinear bridge users are available in this CLCS **then**
17: Transform the source position to the corresponding one in this CLCS through solving Eqs. (2);
18: **end if**
19: **end for**
20: **end for**
21: **if** the source position is available in any of the user own CLCSs **then**
22: **if** the user itself is the destination **then**
23: Return LBS success;
24: **else**
25: Embed information into the LBS message;
26: **end if**
27: **else**
28: Return LBS failure;
29: **end if**

Functions:

30: **function** CONSTRUCT LCS
31: Broadcast its information request message;
32: Wait for Δ_{t2} time for building its two-hop topology;
33: Construct CLCS according to Sec. III-C1;
34: **end function**
35: **function** EMBED INFORMATION INTO THE LBS MESSAGE
36: **for** each CLCS **do**
37: **if** the source position in this CLCS is available **then**
38: **if** the positions of at least three non-collinear users are available in this CLCS **then**
39: Embed the CLCS ID, the source position, and the positions of all the users in this CLCS into the message;
40: **end if**
41: **end if**

42: **end for**
43: **if** no information has been embedded **then**
44: Return LBS failure;
45: **else**
46: Transmit the message to its next-hop user;
47: **end if**
48: **end function**

* Δ_{t1} and Δ_{t2} are set according to the real network setup.

* $\Delta_{t2} \approx (\beta + 1) \times \Delta_{t1}$. See Sec. IV-B for detail.

D. Metrics and Impact Factors

The framework of on-demand fast localization is event-driven and fully distributed. Its localization procedure is initiated only on demand. We evaluate its performance on the LBS success ratio, localization time, energy consumption, position privacy, and position accuracy. The performance metrics are defined as follows:

Definition 3.2: (LBS Success Ratio) is the ratio of the number of returned *LBS success* (line 23) to the total number of returns (line 23 + line 28 + line 44).

Definition 3.3: (ODFL Time) is the starting time for transmitting an LBS message at the information source (line 2).

Definition 3.4: (Eventual Localization Time) is the time for the destinations to receive an LBS message and to decode the source position (line 23).

Definition 3.5: (LBS Energy Consumption) is the total energy consumed by successfully transmitting an LBS message to the destinations (line 23).

Definition 3.6: (Source Position Leakage) is the total number of users that are not on the information transmission route but can capture the LBS message in the air and decode the source position.

Definition 3.7: (Source Position Accuracy) is the accuracy of the source position in the destination's CLCS (line 17).

We also define two factors that have impacts on the above performance metrics as follows:

Definition 3.8: (User Velocity) is the moving speed of the mobile user, which determines the frequency and gradient of topology changes.

Definition 3.9: (Transfer time) is the time between the message forwardings (line 46) issued by two consecutive intermediate users (such as the previous user and the current user). In other words, it is the time required by an intermediate user before forwarding a message to its next-hop user. The transfer time determines the extent of bridge user's changes in terms of position and availability.

Note that the transfer time is determined by the channel condition, the system and hardware, and the variance of number of two-hop neighbors at the two consecutive users; but it is not necessarily related to Δ_{t1} and Δ_{t2} , as the next-hop user starts constructing

its LCS (**line 6**) before receiving the application's message. It means that the transfer time could be very small (only including the time spent on receiving the message, transforming the positions, and forwarding the message) if constructing the LCS at two consecutive users takes roughly the same time.

IV. PERFORMANCE ANALYSIS

A. LBS Success Ratio

In the framework of ODFL, the success of LBS depends on the success of source position transformation at each user on the information transmission route. We define the *success of an intermediate user* based on the existence of at least one CLCS that (i) can localize three non-collinear users, which can also be localized in one of its previous users CLCSs where the source position is available; and (ii) can localize three non-collinear users, which can also be localized in one of its next-hop users CLCSs (where the destination position should be available if the next-hop user is the destination). As the previous user and the next-hop user should not be able to directly discover each other under most routing protocols, to guarantee success, we require that the minimum node degree for the intermediate user be 4 (1 connection is from the previous user, 1 from the next-hop user, and 2 from the common neighbors). Fig. 4 presents an example where the intermediate user (X)'s node degree is 4 and an LBS application can succeed if $X-1$ and $X+1$ are the source and the destination, respectively. In this example, the coordinate system $\langle a, b, X \rangle$ is the CLCS that can guarantee the success of LBS.

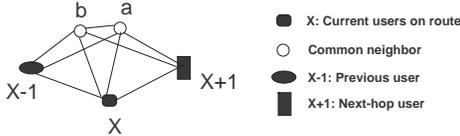


Fig. 4. Minimum topology for success position transformation

In practice, however, the node degree of 4 may not be sufficient to guarantee the success of LBS. The LBS success ratio is also determined by the network density, the user distribution model, the user velocity, and the transfer time. We will evaluate the LBS success ratio via simulations in Sec. V.

B. Localization Time

It follows from the framework design (**line 2 and line 46**) that an LBS application can start its procedure of information transformation approximately after $\Delta_{t2} + \Delta_c$ time, where Δ_{t2} is the time required for constructing a user's two-hop topology and Δ_c is the time for constructing the LCS. Consequently, the

ODFL time can be calculated by

$$\Delta_{t2} + \Delta_c \quad (3)$$

As the maximum node degree is β and Δ_{t1} is the time required for constructing a user's one-hop topology, we have $\Delta_{t2} \approx (\beta + 1) \times \Delta_{t1}$. In other words, the time for constructing a two-hop topology roughly equals the summation of the time required by constructing $\beta + 1$ one-hop topologies, for the user and every direct neighbor of the user all need to discover their direct neighborhoods. For relative localization, where the distance information between any two one-hop users has to be acquired via collecting two-hop topology, $\Delta_{t2} + \Delta_c \approx t_1$, where t_1 is defined by Eq. (1).

Assuming that the source position can be successfully decoded at the destination, the eventual localization time should equal the aggregated time spent at all the users on the information transmission route (**line 2 + line 25 + line 14**). For each intermediate user, the time spent on supporting an LBS message transmission consists of four parts: (1) responses to the previous user's information request; (2) construction of two-hop topology; (3) construction of the LCS; and (4) transmission of the message to the next-hop user. Note that the consumed time of the first part completely overlaps with the time of the previous user's two-hop topology construction. As a result, we do not consider that this time is consumed by the current user. Then each user spends $\Delta_{t2} + \Delta_c + \Delta_D$ time for supporting an LBS message transmission, where Δ_D is the average transmission delay defined in Eq. (1). Note that the time for parts (2), (3), and (4) may also overlap with the time on parts (2) and (3) of the user's next-hop user. Thus, the localization time is bounded above by

$$d_h \times (\Delta_{t2} + \Delta_c + \Delta_D) + (\Delta_{t2} + \Delta_c), \quad (4)$$

where d_h is the number of hops on the routing path. Under the conventional localization definition, theoretically, the earliest time for starting an LBS message transmission is $T_{min}(r_h, t_1, 1) = r_h \times (t_1 + \Delta_D)$ in relative localization. Since the ODFL time is only $\Delta_{t2} + \Delta_c$, the framework of ODFL can reduce the waiting time before starting LBS applications from a linear function of t_1 to a constant that approximates to t_1 . The earliest time for the destination to successfully receive an LBS message is bounded below by $r_h \times (t_1 + \Delta_D) + d_h \times \Delta_D$ under the traditional definition and based on the assumption that the source can guarantee that the destination can be localized in the same coordinate system after $r_h \times (t_1 + \Delta_D)$ time. Note that this lower bound is in general larger than the upper bound of ODFL's localization time when $d_h < r_h$. It is also important to note that the real time required by localization and the improvement of on-demand fast localization depend on the network

topology, the mobility model, the traffic model, and the underlying routing protocol.

C. Position Accuracy

Since nodes are unable to take position snapshot and record movements after event, localization error is an inevitable result of various delays in mobile networks. Assuming a reasonably small ranging delay, localization error in conventional approach mainly consists of two parts: 1) intermediate error accumulated along the transmission route due to delay of finding three localized neighbors for each intermediate nodes to acquire their position with trilateration; 2) source error caused by the delay between the event time and when the source finally localizes itself. Longer overall localization time incurs larger source error.

With ODFL on the other hand, the source error can be significantly reduced or essentially removed, since source can localize itself in LCSs as soon as ranging to neighbors is completed. Intermediate error exists during position transformation at each intermediate node, in terms of the delay in finding and localizing bridge nodes. Localization error is accumulated as source position is translated from one LCS to another. Nonetheless, considering inaccurate ranging would be the same dominating factor for localizing nodes in both ODFL and conventional approaches, this accumulative error in ODFL is at most comparable to the intermediate error in conventional approaches. As a result, ODFL can achieve higher localization accuracy by starting ODFL process as early as possible to minimize source error.

D. Energy Consumption

Energy saving has always been an important concern for mobile applications. Generally, the on-demand fast localization can save the energy consumption caused by LBS applications because it eliminates the unnecessary transmissions carrying only localization information by integrating the transmissions of position transformation into the application's payload transmissions. Moreover, the energy consumption is more controllable as the expected ODFL time is a constant and only the users on the route and their two-hop neighbors will be involved in the localization procedure. On the contrary, the conventional localization techniques incur much higher and unpredictable energy consumption for localization, for the localization procedure may not stop in highly dynamic mobile networks and it may have to involve all users.

We measure the energy consumption by counting the number of transmissions and receptions used to support LBS applications, as they usually incur much higher energy cost than position computations. Under the framework of ODFL, each user on the routing path needs to transmit four types of messages: (1) an

information request; (2) responses to the information requests from its previous user (except the source) and its next-hop user (except the destination); (3) responses to its direct neighbor's neighborhood discovery request; (iv) transmission of the LBS message to its next-hop user (except the destination). Note that it is not necessary to transmit the third type of message because the user's direct neighbors can discover the user after receiving the information request message, and that the user only needs to respond to the information request once, for its next-hop user can also obtain the message. The number of users on the routing path is $N_0 = d_h + 1$. Thus, the total number of transmissions on the routing path is

$$3 \times N_0 - 1 = 3d_h + 2. \quad (5)$$

The users, whose minimum distance to the users on the routing path is one hop, need to transmit additional two types of messages: (5) a neighborhood discovery request and (6) responses to the information requests. The number of such one-hop users is bounded above by $N_{1-hop} = (d_h + 1) \times \beta - 2 \times (d_h - 1) - 2$, for at least two direct neighbors of each intermediate user, one direct neighbor of the source, and one direct neighbor of the destination are on the path. Note that the one-hop user only responds to the information requests once, for all its direct neighbors can receive the message. Consequently, the number of transmissions at all the one-hop users is bounded above by

$$2 \times N_{1-hop} = 2(\beta - 2)d_h + 2\beta. \quad (6)$$

The users, whose minimum distance to the users on the routing path is two hops, only need to (7) respond to the neighborhood discovery request from the one-hop users. The number of such two-hop users is bounded above by $N_{2-hop} = (\beta - 1) \times N_{1-hop}$, for each one-hop user has at least one direct neighbor on the routing path. Note that each two-hop user may need to respond at most β times as all its direct neighbors could be the one-hop users, which may send their neighborhood discovery requests at different times. Therefore, the number of transmissions at all two-hop neighbors is bounded above by

$$\beta \times N_{2-hop} = \beta(\beta - 1)(\beta - 2)d_h + \beta^2(\beta - 1). \quad (7)$$

It follows from Eq. (5), (6), and (7) that the total number of transmissions for supporting an LBS message is bounded above by

$$(\beta^3 - 3\beta^2 + 4\beta - 1)d_h + \beta^3 - \beta^2 + 2\beta + 2. \quad (8)$$

The users in the network transmits a total of 7 types of messages, which can be categorized into three classes: broadcast, multicast, and unicast. In particular, type (1), (5), and (6) belong to broadcast, type (2)(up to 2 receivers) belongs to multicast, and type (4) and

(7) belong to unicast. Accordingly, we can obtain the upper bound of the total number of receptions in (9) as follows:

$$\begin{aligned} & \beta(N_0 + 2N_{1-hop}) + 2N_0 - 2 + (N_0 - 1) + \beta N_{2-hop} \\ = & (\beta^3 - \beta^2 - \beta + 3)d_h + \beta^3 + \beta^2 + \beta. \end{aligned} \quad (9)$$

We can see from time bounds (8) and (9) that the upper bound of the total number of transmissions and receptions under on-demand fast localization is $O(d_h)$ when the maximum node degree β is a constant.

Using the conventional localization technique, all the users in a network need to broadcast their positions at least once. As a result, the localization procedure needs at least N transmissions and $N\beta$ receptions, where N is the number of users in the network. Consequently, the total number of transmissions and receptions for supporting an LBS message is at least $N + d_h$ and $N\beta + d_h$, respectively. It means that the energy consumption under conventional localization techniques is $O(N)$, which is in general larger than $O(d_h)$ under on-demand fast localization.

E. Position Privacy

Using the conventional localization techniques, all the users that can overhear the LBS messages can decode the source position as they are all localized in the same coordinate system. On the contrary, in the framework of ODFL, users that are not on the routing path do not calculate their positions. Even if they try to localize themselves, their positions are only available in their own CLCSs. As a result, they cannot directly decode the embedded source position even if they can capture the messages. Moreover, they may fail to transform the captured source position to the corresponding ones in their CLCSs because of the lack of information for position transformations. Therefore, the framework of ODFL can also reduce source position leakage during the procedures of localization and message transmission.

V. SIMULATIONS

We evaluate the performance of ODFL over AODV [11] using the INETMANET simulation models in OMNet++ 4.2.2. In the simulations, 29 mobile users are randomly deployed in a 1200×1200 (meter) playground, 1 mobile user is initially deployed at location (100, 100) as the information source, and 1 static station is deployed at (1100, 1100) as the destination. The information source periodically sends UDP packets as LBS messages to the destination. The mobile user velocity vary from 3.6km/h to 108km/h. The success ratio and the eventual the localization time are obtained based on the UPD success ratio and the UDP delay, respectively. Fig. 5 reports the simulation

results under the impacts of the user velocity and the transfer time, which vary from 0.01s to 2.2s. Each reported point in figures is an average of 150 instances over 3000s simulation time.

Fig. 5(a) reports the the success ratio of LBS applications under ODFL. We can see that the success ratio can almost reach 1 when the transfer time is less than 0.3s, and that it generally decreases with the increasing of the transfer time and the user velocity. The reason is that the set of two-hop neighbors at the next-hop user may change frequently with a longer transfer time if the users move faster. As a result, the next-hop user has lesser chance to find three non-collinear commons users from the previous embedded information for the position transformation. Similar observations can also be obtained for localization accuracy from Fig. 5(b). Theoretically, the localization error is more sensitive to the neighbor changing because not only the number of bridge users but also their positions can affect the localization result according to Eq. (2). Based on the results, the obtained source positions are only valid when the transfer time is less than 0.3s or the user velocity is less than 25km/h. We can also see why the conventional localization techniques could be an endless process in mobile networks from Fig. 5(b), for the minimum network localization time T_{min} is generally much larger than 0.3s. The energy consumptions of ODFL are reported in Fig.5(c) and Fig.5(d) in terms of the number of transmissions and the number of receptions at each user for each LBS message, respectively. It is interesting to observe that the transfer time has less impact on the energy consumption than the user velocity, and that the energy consumption decreases with the increasing velocity. This is partially due to the smaller success ratio at higher velocity, where the LBS messages are dropped before transmitting to the destination. Fig. 5(e) reports the localization time of ODFL in terms of LBS message transmission delay. We can see that the transfer time dominates the changes of eventual localization time, and that high velocity can help with reducing the delay as the movement of mobile users may reduce the number of hops to the destination. In summary, the ODFL implementation over AODV can successfully support the LBS applications in mobile networks, when the transfer time is less than 0.3s or the user velocity is less than 25km/h, with low energy consumptions.

VI. RELATED WORKS

The localization techniques in the literature can be categorized into two classes: (1) ranging-free localization and (2) ranging-based localization. Generally, ranging-free localization schemes have stronger requirements of the number of anchors and the anchor's capability (such as [6]), and the localization accuracy is low. On the other hand, ranging-based

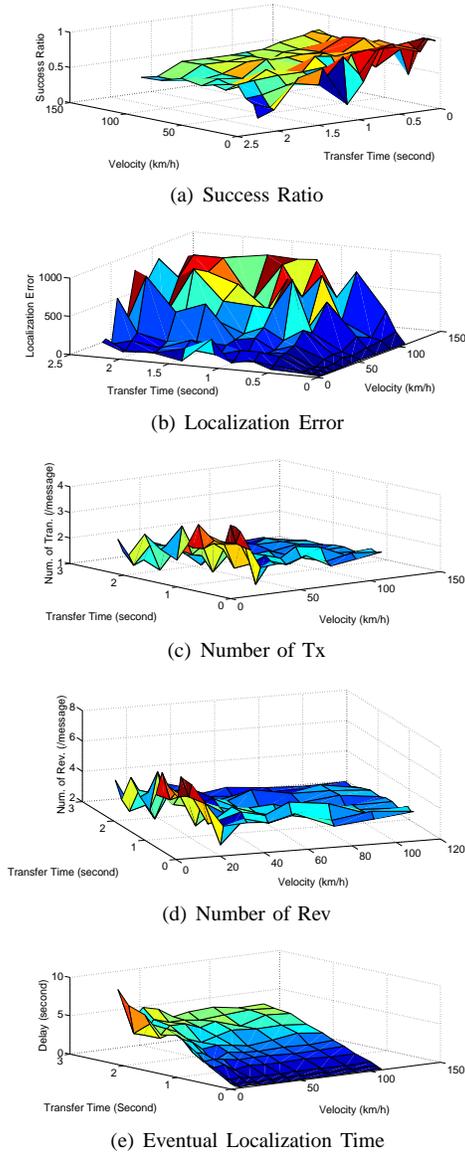


Fig. 5. Simulation results

schemes are generally more accurate and may not require any anchor (such as [2]). When the ranging information is available, the problem of assigning node positions to satisfy the edge-length constraints is NP-hard [1]. The notion of K -lateration, where K is an integer, provides a mathematical foundation for a set of localization schemes (such as [9], [10]) that can complete the localization in polynomial time, and all of its centralized and distributed implementations are equivalent in localizability [3]. However, the polynomial localization time may still be too long for time-critical LBS applications. So far no theoretical or practical studies have addressed this issue in mobile networks. To the best of our knowledge, Cheng et al's paper [4] is the only work that had made the first step toward reducing the localization time, but the results

are obtained based on the assumption that the network is static. This paper is the first study on the on-demand fast localization problem while considering at the same time the underlying routing protocols.

VII. CONCLUSIONS

We present a new concept of ODFL for supporting time-critical LBS applications in mobile ad hoc networks by merging the procedure of localization into routing. The proposed ODFL framework can increase the success ratio of LBS applications, reduce the localization time and energy consumption, and improve the source position privacy. The ODFL implementation over the AODV routing protocol can successfully support LBS applications in MANETs, when the transfer time is less than 0.3s or the user velocity is less than 25km/h.

REFERENCES

- [1] J. Aspnes, T. Eren, D. K. Goldenberg, A. S. Morse, W. Whiteley, Y. R. Yang, B. D. O. Anderson, and P. N. Belhumeur. A theory of network localization. *IEEE Transactions on Mobile Computing*, 5(12):1663–1678, 2006.
- [2] S. Capkun, M. Hamdi, and J. Hubaux. Gps-free positioning in mobile ad-hoc networks. In *HICSS '01*, page 9008, Washington, DC, USA, 2001. IEEE Computer Society.
- [3] W. Cheng, A. Teymorian, L. Ma, X. Cheng, X. Lu, and Z. Lu. Underwater localization in sparse 3d acoustic sensor networks. In *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, pages 236–240, april 2008.
- [4] W. Cheng, N. Zhang, X. Cheng, M. Song, and D. Chen. Time-bounded essential localization for wireless sensor networks. *Networking, IEEE/ACM Transactions on*, 2012.
- [5] J. geun Park, D. Curtis, S. Teller, and J. Ledlie. Implications of device diversity for organic localization. In *INFOCOM '11*, pages 3182–3190, april 2011.
- [6] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher. Range-free localization schemes for large scale sensor networks. In *MobiCom '03*, pages 81–95, New York, NY, USA, 2003. ACM.
- [7] S. Ji, K.-F. Sze, Z. Zhou, A. M.-C. So, and Y. Ye. Beyond convex relaxation: A polynomial-time non-convex optimization approach to network localization. In *The 32th IEEE Conference on Computer Communications (INFOCOM 2013)*, Turin, Italy, April 14-19 2013.
- [8] Y. Kwon, K. Mechitov, S. Sundresh, W. Kim, and G. Agha. Resilient localization for sensor networks in outdoor environments. *ACM Transactions on Sensor Networks*, 7(1), August 2010.
- [9] T. Melodia, D. Pompili, and I. F. Akyldiz. Handling mobility in wireless sensor and actor networks. *IEEE Transactions on Mobile Computing*, 9:160–173, February 2010.
- [10] D. Moore, J. Leonard, D. Rus, and S. Teller. Robust distributed network localization with noisy range measurements. In *SensSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 50–61, New York, NY, USA, 2004. ACM.
- [11] C. E. Perkins and E. M. Royer. Ad-hoc on-demand distance vector routing. In *HotMobile '97*, pages 90–100, 1997.
- [12] S. Yang, P. Dessai, M. Verma, and M. Gerla. Freeloc: Calibration-free crowdsourced indoor localization. In *The 32th IEEE Conference on Computer Communications (INFOCOM 2013)*, Turin, Italy, April 14-19 2013.
- [13] X. Zhu, Q. Li, and G. Chen. Apt: Accurate outdoor pedestrian tracking with smartphones. In *The 32th IEEE Conference on Computer Communications (INFOCOM 2013)*, Turin, Italy, April 14-19 2013.