# A Location-Based Publish/Subscribe Framework for Wireless Sensors and Mobile Phones

[1]He Huang, [1]Edith C.-H. Ngai and [2]Jiangchuan Liu
[1]Department of Information Technology, Uppsala University, Sweden
[2]Department of Computing Science, Simon Fraser University, Canada

***Abstract*—Wireless sensor networks (WSNs) have been widely deployed for environmental monitoring and urban sensing applications. With the advancement of mobile phones, mobile users have increasing demands for sensing data relevant to their locations and activities. It is crucial to support reliable ubiquitous sensing for mobile users to retrieve sensing data of their interests anytime and anywhere. In this paper, we propose a novel location-based publish/subscribe framework for mobile users to subscribe for sensing data by simply specifying the event types and target locations of their interests. The framework is developed on a location-based Distributed Hash Table (DHT) overlay formed by a network of brokers. Event data can be subscribed by mobile users and then multicast to the subscribers through efficient location-based routing. To support client mobility, we further propose a reliable protocol to handle client registration, location look-up and client relocation. It can provide reliable data delivery without causing any data loss even when the users are disconnected or moved to a new place. Extensive simulation are conducted to evaluate the message delay, communication overheads, and the support of client mobility.**

## I. Introduction

A number of innovative location-based services and sensing applications have been implemented, such as environmental monitoring and urban sensing applications [1], [2]. Although wireless sensors could be deployed for monitoring the environment, they are usually small and stationary devices with limited computation capability, battery, and no user interface [3], [4]. Mobile devices such as smart phones, which have larger memory and batteries, could be utilized as mobile terminals to collect sensing data from remote or sparsely deployed sensors [5], [6]. With the advancement of mobile and sensing technology, mobile users will be able to obtain ubiquitous sensing data relevant to their locations and activities anytime and anywhere. They can subscribe to events of interest in a sensing field and receive the corresponding data when the unusual events occur. It is important to provide efficient and reliable subscription and data delivery even though there is disconnection or relocation of the mobile users.

In this paper, we present a novel publish/subscribe framework that enables mobile clients to subscribe for sensing events of interests at specified locations. Whenever a subscribed event is detected by any wireless sensors, notifications with relevant data will be disseminated to the corresponding subscribers. Given a dynamic network with both mobile devices and randomly deployed wireless sensors, our major challenge is to design a reliable network infrastructure that can provide users with efficient subscriptions and data delivery even in mobility. More specifically, our framework will include the following features: (1) The system architecture will be distributed, scalable and able to support heterogeneous devices. (2) The routing strategies will be efficient in terms of message delay and communication overheads. (3) Reliable data delivery will be supported to prevent data loss in disconnections or relocations due to mobility of clients.

To achieve the above goals, we propose a novel publish/subscribe framework for mobile and sensing devices by building a distributed and location-based peer-to-peer (P2P) overlay with a network of brokers. Mobile users can subscribe for sensing events in specified target areas through their associated brokers. The delivery of sensing data will be triggered by the sensors or mobile phones that observe the subscribed events. The sensing data will be forwarded to the corresponding brokers and dispatched to the subscribers. Location-awareness in our publish/subscribe system is facilitated by an addressing scheme based on a deployment space partition strategy. Efficient routing strategies are proposed to reduce the communication overheads for both the brokers and mobile devices. Our system also includes a handover protocol, which can handle client mobility by redirecting event data reliably from the previously associated broker to the new location.

## II. Related Work

Several publish/subscribe frameworks have been proposed for WSNs. TinySIP [7] makes the functionality of the Session Initiator Protocol (SIP) available to WSNs. It supports session semantics, publish/subscribe service and instant messaging. Mires [8] is another publish/subscribe architecture for WSNs. In Mires, sensors only publish readings after the user has subscribed to specific sensor reading. Messages are aggregated in the clusterheads. Subscriptions are issued from the sink node directly connected to the end user terminals. DV/DRP [9] is a publish/subscribe architecture in which subscriptions are made based on the content of the desired messages. Subscriptions are flooded in the network. Intermediate nodes aggregate the subscriptions and forward the data only if there is an interest for the data. However, all of the above systems considers stationary network without any mobile nodes. Moreover, either the advertisements or the subscriptions have to be flooded to the whole network, which may lead to high message overheads.

978-1-4673-0437-5/12/$31.00 ©2012 IEEE 

Structured P2P overlay networks, such as Distributed Hashed Table (DHT) [10], have been suggested for building decentralized distributed applications. Pastry [11] is a scalable, distributed object location and routing substrate for wide-area P2P applications. Each Pastry node can efficiently route message to the node with a node ID that is numerically closest to the key. Scribe [12] is an application-level multicast infrastructure that built upon Pastry [11]. It forms a multicast tree by joining the Pastry routes from each subscriber to a rendezvous point associated with a topic. Since the above approaches route the subscriptions and the data simply based on the key and the nodes IDs, the locations of sensors have not been taken into account. Although location-based publish/subscribe has been studied in [13], it assumes that the publishers and the subscribers are within the communication range. Different from our work, location-based routing between the publishers and subscribers has not been fully investigated.

Client mobility is another key problem in providing reliable publish/subscribe services for WSNs and mobile devices. A publish/subscrib middleware has been presented in [14] to support mobile clients. It relies on an acyclic underlying network topology so that a conjunction node between the old and new routes can always be found. A component called "mobile proxy" is introduced in [15] to explore the possibility of creating a generic solution for different publish/subscribe frameworks. Other than that, the MobileIP protocol [16] has been proposed, which applied techniques like Care-of-Address (CoA), Home Agent (HA) and Foreign Agent (FA) to cope with host mobility issues. Similar to MobileIP, our approach also takes the advantages of on-demand handover to avoid the triangle routing problems. Nevertheless, we explore the location information of the nodes in our P2P overlay and integrate it seamlessly with our location-based publish/subscribe scheme in the application layer.

## III. Network Infrastructure

We consider a network consisted of three kinds of nodes, including brokers, sensors and mobile phones as shown in Figure 1. Mobile phone users can subscribe data from the wireless sensors located at different part of the sensing field through a network of brokers. The brokers are interconnected through various available communication standards, such as WiFi, Bluetooth, ZigBee, GPRS, etc. Any wireless sensors or mobile phones can join the network if they can communicate with any of the brokers. The role of the brokers is to maintain the information and process the requests of the associated mobile users, such as storing client subscriptions, routing event notifications to subscribers, and handling the handover process in support of client mobility. Buffers are allocated in brokers for storing subscriptions, client information and sensing data. There is a logical interconnection of brokers which is based on a structured peer-to-peer (P2P) overlay in the network. This design can support any distributed hash table (DHT) protocols applied longest prefix matching in routing.

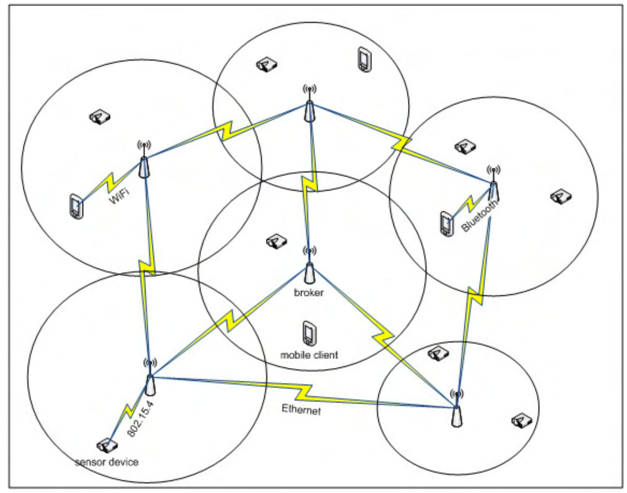


Fig. 1. Network infrastructure with sensors and mobile devices.

## IV. Location-Based Addressing

We present a location-based addressing strategy, in which subscriptions are propagated only to the areas of interest for efficient communications.

### A. Field Partitioning

We consider that the mobile phones and sensors in the system can obtain their locations by GPS [17] or some existing localization techniques [18], [19]. The deployment area is split into partitions managed by the brokers. Figure 2 illustrates the deployment field partitioning with 10 brokers. The join of a new broker always results in a binary splitting of one existing partition. The partition and ID assignment forms a hierarchical structure according to the common prefix of their IDs. Each broker will take the prefix of its partition ID as the prefix of its broker ID. The broker ID will be used as the destination key to route the messages. Since the ID prefixes are assigned according to the locations, it can ensure that the two nodes having adjacent node ID prefixes are geographically close to each other. This location-aware node ID assignment reduces the difference between the logical route in the DHT and the physical path in the underlying network topology, which can reduce message latency in routing.

### B. Boundary List

We include a boundary list to store the information of the partitions in each broker. Boundary list has its size dependent on the level of the node in the hierarchical partition. The $i^{th}$ element in the list stores the splitting boundary of the partition that shares a common prefix in the first $i$ bits of its broker ID. Each element in the boundary list is represented in the format of a triplet $\{$Dimension, Coordinates, Direction$\}$. The first field "Dimension" is either "X" or "Y", specifying the splitting is done based on which dimension. The second field "Coordinates" specifies the boundary location on the splitting dimension. The last field "Direction" indicates on which side of the splitting the broker is located. Possible values are "INC"

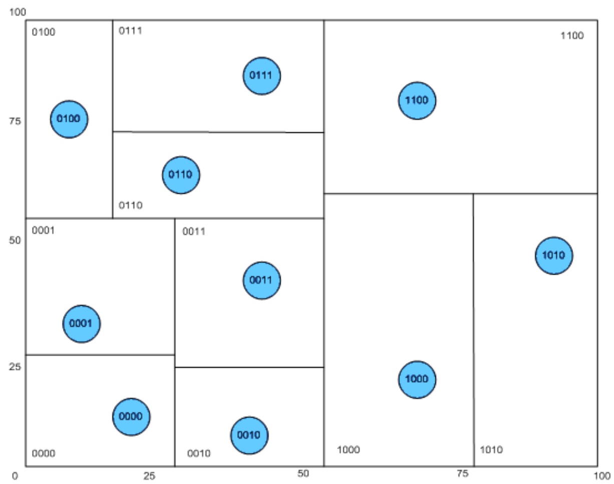


Fig. 2. Deployment space partition with 10 brokers.

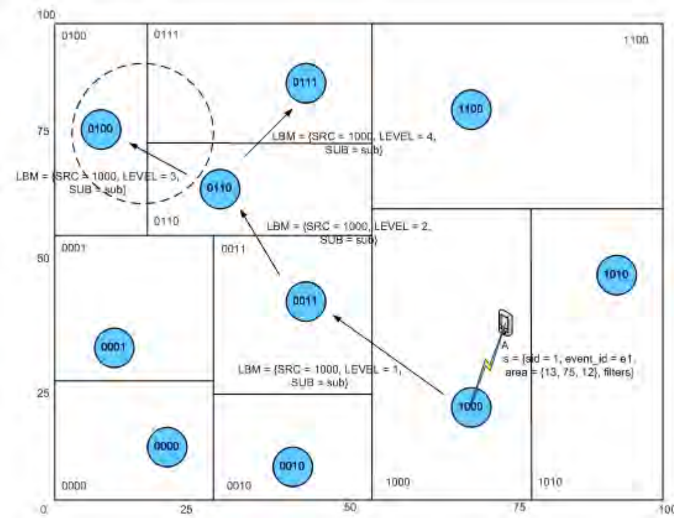


Fig. 3. Subscription installation by LBM routing.

or "DEC", which stand for the increasing or decreasing side on the chosen dimension. For instance, the boundary list of broker B0011 in figure 2 could be {X, 56, DEC}, {Y, 53, DEC}, {X, 27, INC}, {Y, 20, INC}.

### *C. Client Association*

Each client must be associated to a broker in order to subscribe and receive event data. Client information, such as a unique client identification, an IP address and a listening port, will be registered to the broker. A client will send a request message to the broker for association. The broker will allocate buffer for storing the client information and the subscribed events. Afterwards, the broker will send a response message back to the client indicating the end of the client association process. In case of client mobility, this association process is extended to support client reconnection and handover, and to enable message replay capability.

## V. Publish/Subscribe Protocols

In this section we explain the routing protocols for subscription installation, event publication and event data dissemination.

### *A. Subscription Installation*

When a client subscribes for an event of interest, it first registers its interests to the associated broker by sending a "SUB" message {sid, event_id, area, filters}, where "sid" is an unique ID for the subscription, "event_id" is an ID for the event type, "filters" is a set of attribute-value pairs (AVPs) applied to the published event, so that only the events of interest will be disseminated to the corresponding subscribers. The field "area" describes the target area where the subscription takes effect. For simplicity, we model the target area as a circle = {x, y, r} centered at (x,y) with the specified radius r.

The brokers make use of their boundary lists and routing tables to route the "SUB" message to the destinations through location-based multicast (LBM). In LBM, each broker checks its local boundary list to see which side the target area belongs to. A level index is introduced here to prevent the route from forming a loop. The level index $i$ is set to 0 for the source broker. If the target area is covering two or more partitions, the "SUB" message will be sent to the brokers in all of these partitions. More details of the LBM algorithm is presented in Algorithm 1.

**Algorithm 1** LBM subscription installation algorithm

**for** each element $b_j$ in the boundary list of broker B **do**
  $dis_j$ = distance($b_j$ , A.center)
  **if** $|dis_j| < A.radius$ **then**
    $S'$ = copy of (S)
    $p[0 \dots j-1] = B.prefix[0 \dots j-1]$
    $p[j] = NOT B.prefix[j]$
    $p[j+1 \dots length-1] = 0$
    send $S'$ to the node that shares the common prefix $p$ with B in B$'s$ DHT routing table with index number $i = j + 1$
    **if** the target area overlaps with B$'s$ partition **then**
      B stores the subscription
    **end if**
  **end if**
**end for**

Figure 3 illustrates a subscription installation scenario in which user A issues a subscription of event "e1" to the target area {13, 75, 12}, which is geographically a circle covering three partitions, "0100", "0110" and "0111". Given the kd-tree formed on the routes by the brokers, the time complexity measured in number of hops for the LBM subscription installation algorithm is $\log_2 N$, where $N$ is the number of brokers.

### *B. Event Notification to Responsible Brokers*

When any of the subscribed events is observed by the sensors or mobile phones equipped with sensing capabilities, the event notification process will be triggered. Events will be published by sensing devices in the format of e = {eid, loc, timestamp, data}, where "eid" is a unique event ID. The field "timestamp" indicates the time when the event occurred. The field "data" encloses the real sensing data. The field "loc", in the format of loc = {X, Y}, specifies the X,Y coordinates where the event occurred.

When an event is captured by a sensing device, the device will encapsulate the event information in a "PUB" message and forward it to the associated broker. The broker will examine the "loc" field of the message to see if the event occurred in its own partition. If not, the "PUB" message has to be forwarded to the responsible broker of that partition. Note that the associated broker is not necessary the broker responsible for the received "PUB" message due to the inconsistency between the partitions and the wireless communication range.

### C. Data Delivery to Subscribers

When a "PUB" message is dispatched to its responsible broker, the encapsulated event information (i.e. event ID, location, filters) is matched against the subscriptions stored in the broker. For each of the matched subscriptions, the corresponding event data will be disseminated by the broker through the overlay network to the subscribers.

Since the traditional DHT protocol does not support multicast in key-based routing (KBR) [20], [21], we introduce an ID-based multicast (IDM) algorithm in overlay network level to reduce the communication overheads. The IDM algorithm is executed when a broker B receives a notification N with a list of destination IDs as shown in Algorithm 2. As a result, event notification can be dispatched to all the subscribed brokers along the paths that shape a spanning tree. The time complexity in terms of message hop count is $\log_2 N$, which is the same as the average hop count of most ID-based DHT routing protocols.

**Algorithm 2** IDM event notification to subscribers

```
for subscriber ID on the client list in broker B do
    Get next hop h of B's routing states
    if h is not found then
        Send a copy of the notification N(eid, data) to each client c
        in CL
    else
        if h is not in T then
            Add h into T
        end if
        add current {id, CL} in T.h.destination
    end if
    for each h in T do
        Send a copy of the notification N(eid, data, T.h.destination)
    end for
end for
```

## VI. Support of Client Mobility

We further extend the client association process and the event notification protocol to support mobility of clients. Our handover protocols is implemented in the application layer, which allows us to explore the location and publish/subscribe information from the P2P overlay.

### A. Client Registration and Location Lookup

Client location registration is a process for updating the location information of a mobile client whenever a reconnection or a relocation occurs. In a mobile scenario, the connection between the broker and the client could be broken for any length of time depending on the communication range and remaining battery. Hence, it is important for the clients to update their information to the system when they associate to any brokers again. In our design, we utilize the identity of the associated broker to estimate the location of the mobile client. The mapping from the client identity to its associated broker ID, i.e. $cid \rightarrow bid$, is referred as Care-of Address (CoA).

We integrate the CoA with the overlay of brokers. For each client, its CoA information will be stored in the broker whose node ID is numerically closest to the hash value of the client identity. We name this broker as the Home Broker (HB) of that client. Each time a mobile client associates to a broker, the broker will compute a hash value by applying a hash function to the client identity, and forward the client location information to the home broker with a "REG" message. Whenever a broker in the overlay loses the trace of a mobile client, it can fetch the latest location of the client by routing a "LOOKUP" message to the home broker.

### B. Handover and Event Notification Redirection

We design our handover protocol by introducing a new data structure, called "forward pointer", which is maintained in the broker. Forward pointer is a table storing key value pairs of client identifications and the IDs of their newly associated brokers after relocations. Our handover protocol ensures that the buffered data during disconnection or relocation will be replayed to the new location without any data loss.

We present the procedure of a mobile client relocated from broker 1 to broker 2 as an example. At beginning, client A is associated with broker 1, such that the home broker of client A stores A's CoA={A, broker 1}. All matched event data are disseminated to client A through broker 1. At a later time, a disconnection of client A occurs and is detected by broker 1, broker 1 then starts buffering all the incoming event data towards client A to avoid data loss. After awhile, client A sends an association request to broker 2. Broker 2 notices the relocation of client A from its old broker ID from the association message. A handover process is then triggered in broker 2 by sending a handover message to broker 1 after registration in the home broker. Upon receiving the handover message, broker 1 starts to replay all the buffered event data for client A to broker 2. It also informs the new location information of client A to all the brokers from which event data instances have been buffered, so that the new event data will be redirected to the new location.

## VII. Simulations

We implement our publish/subscribe framework in OverSim, which is an overlay and P2P module for the OMNET++ simulator [22]. Two experiments, one for networks with only stationary clients and another for networks with mobile clients, are conducted.

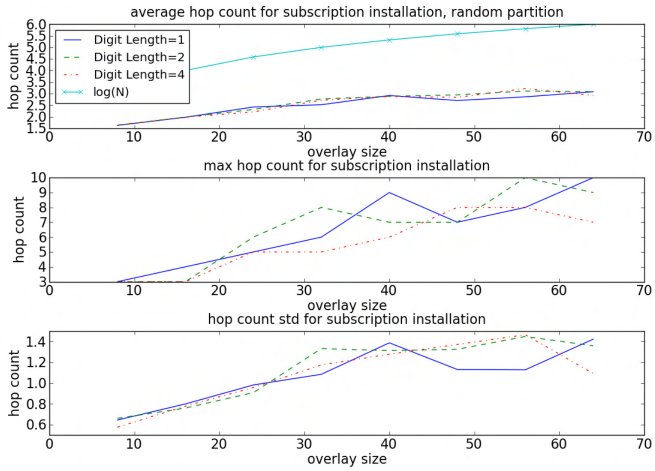


Fig. 4. Average message hop count for LBM routing.

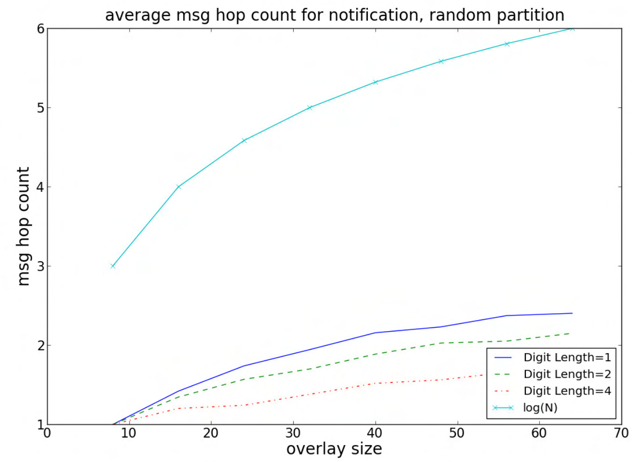


Fig. 5. Average message hop count for IDM routing.

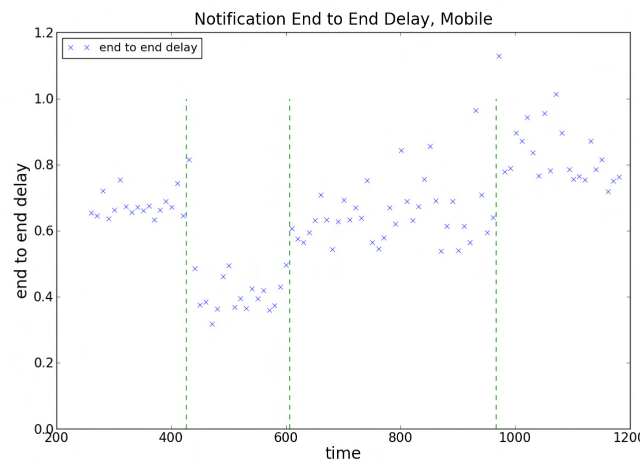


Fig. 6. Message latency on mobile client with short disconnection time.

### A. Experiments with Stationary Clients

We evaluate the performance of the subscription installation process and event data notification process. The experiment is run in an overlay with 8 to 64 brokers and involving only stationary clients in this experiment. It measures the message latency in hop counts.

*1) Subscription Installation:* The top of figure 4 presents the average message latency in hop count for subscription installation. The average hop count is calculated as the sum of message hop counts divided by the total number of destination brokers that have received at least one subscription. The theoretical bound for existing P2P KBR routing protocols ($\log_2 N$) is also plotted in the figure for reference. The maximum hop count of all runs are presented in the middle of figure 4. The bottom of figure 4 gives the standard deviation of the measured hop count. As shown in figure 4, the average hop count increases logarithmically with the size of the broker overlay, but it keeps much lower than $\log_2 N$. We also find that the length of node ID gives no effect to the message latency.

*2) Event Notification:* Figure 5 shows the average message latency in hop count for event notification. The results show that the average hop count increases logarithmically with the size of the broker overlay. Again, the results keep much lower than $\log_2 N$. However, different from the subscription installation protocol, the average hop counts are affected by the length of broker ID. With the increase of ID length, the average hop count decreases for all sizes of broker overlay.

### B. Experiments with Mobile Clients

We evaluate the latency and message overheads of our framework in support of client mobility. There are 32 mobile clients in the network as subscribers, who move randomly in the field following the mass mobility pattern implemented in the INET framework of OMNeT++ [23]. We measure the end-to-end delay as the time difference between the data is published and the data is received by the subscriber.

*1) Event Notification Latency:* Figure 6 shows the end-to-end delay for the events collected by one of the mobile clients from 200s to 1200s. The size of the broker overlay is set to 24. The relocation times of the mobile client are shown as vertical dashed lines. During the extracted time period, the mobile client relocated 3 times and has been associated with 4 different brokers. Since the client always associates to a new broker immediately after it moves out of the previous broker, there is almost no disconnection time for this client. We also observe that the latency for event notifications from the same broker remains almost the same.

We repeat the same experiment with long periods of disconnection at the client. Figure 7 shows that immediately after the first three relocations, several messages have been received at almost the same time. During the disconnection, these messages are buffered in the old broker and then replayed

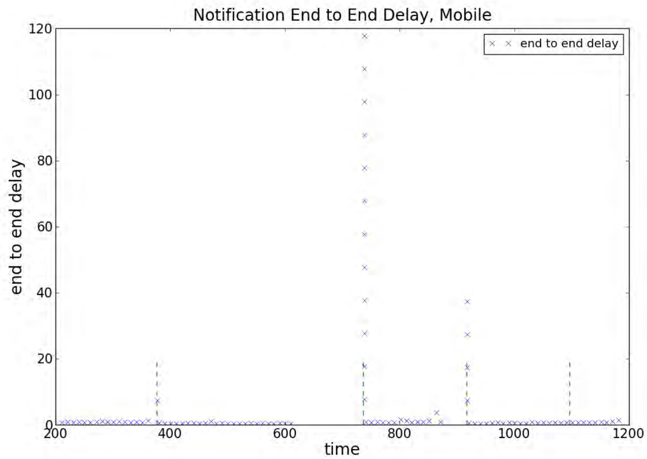


Fig. 7. Message latency on mobile client with long absent time.

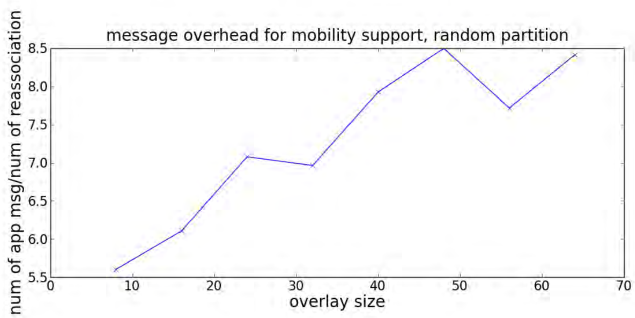


Fig. 8. Message overhead for handover process.

to the new broker as soon as the mobile client reconnects to the broker overlay. There is a 10 seconds gap between each two consecutive buffered messages, which is due to the event publication rate of the system.

*2) Message Overhead:* Figure 8 shows the average message overhead for handover. As the size of the broker overlay grows, the message overhead for handover increases logarithmically. This is because all handover messages are disseminated by the broker overlay through the DHT routing protocol.

## VIII. Conclusions

We proposed a novel location-based publish/subscribe framework that supports mobile users to subscribe and retrieve events of interest in a sensing field. The framework is built on a location-based DHT overlay with a network of brokers. We presented a scalable geographic addressing and routing scheme, which utilizes the locations of the brokers to route the subscriptions to the destinations. Event data collected by the sensors can be forwarded to their associated brokers and multicast to the corresponding subscribers. Client mobility is supported by our framework through a client location registration protocol and a reliable handover protocol to avoid data loss in relocation. Extensive simulation results demonstrate that our scheme can achieve low message latency for both subscriptions and event notifications. It can also handle client mobility successfully by forwarding messages to the new location without causing any data loss.

## References


[1] J. Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy, M. B. Srivastava. *Participatory sensing*, Workshop on World-Sensor-Web at SenSys 2006, Oct 2006.

[2] A. Campbell, S. Eisenman, N. Lane, E. Miluzzo and R. Peterson. *People-centric urban sensing*, in Proc. International Wireless Internet Conference (IEEE WICON), Aug 2006, pp. 2-5.

[3] I. F. Akyildiz and W. Su and T. Sandarasubramaniam. *Wireless sensor networks: a survey*, Computer Networks, vol. 38(5), 2002, pp. 393-422.

[4] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk and J. Anderson. *Wireless sensor networks for habitat monitoring*, in Proc. ACM International Workshop on Wireless Sensor Networks and Applications, Sep 2002, Atlanta, GA, USA.

[5] Z. Ruan, E. C.-H. Ngai, and J. Liu. *Wireless sensor network deployment in mobile phones assisted environment*, in Proc. IEEE International Workshop on Quality of Service, Jun 2010, pp. 1-9.

[6] C. Chen, J. Ma, and K. Yu. *Designing energy-efficient wireless sensor networks with mobile sinks*, In Proc. ACM Conference on Embedded Networked Sensor Systems (SenSys 2006), Boulder, Colorado, USA., October 2006.

[7] S. Krishnamurthy. *TinySIP: Providing seamless access to sensor-based services*, in Proc. International Workshop on Advances in Sensor Networks (IWASN), Jul 2006.

[8] E. Souto, G. Guimares, G. Vasconcelos, M. Vieira, N. Rosa, and C. Ferraz. *A message-oriented middleware for sensor networks*, in Proc. Workshop on Middleware for Pervasive and Adhoc Computing (MPAC), 2004, pp. 127-134.

[9] C. P. Hall, A. Carzaniga, J. Rose, and A. L. Wolf. *A content-based networking protocol for sensor networks*, Department of Computer Science, University of Colorado, Tech. Rep., Aug. 2004.

[10] M. F. Kaashoek and D. R. Karger. *Koorde: A simple degree-optimal distributed hash table*, Peer-to-Peer Systems II, Lecture Notes in Computer Science, 2003, vol. 2735/2003, pp. 98-107.

[11] A. Rowstron and P. Druschel. *Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems*, in Proc. IFIP/ACM Middleware, Heidelberg, Germany, Nov 2001.

[12] M. Castro, P. Druschel, A-M. Kermarrec, and A. Rowstron. *Scribe: A large-scale and decentralised application-level multicast infrastructure*, IEEE Journal on Selected Areas in Communications (JSAC) (Special issue on Network Support for Multicast Communications), vol. 20, no. 8, Oct 2002, pp. 1489-1499.

[13] P. Th. Eugster, B. Garbinato, and A. Holzer. *Location-based Publish/Subscribe*, in Proc. IEEE International Symposium on Network Computing and Applications, 2005.

[14] L. Fiege, F. C. Gartner, O. Kasten, and A. Zeidler. *Supporting mobility in content-based publish/subscribe middleware*, in Proc. ACM/IFIP/USENIX International Middleware Conference, 2003, pp. 103-122.

[15] M. Caporuscio, A. Carzaniga, A. L. Wolf. *Design and evaluation of a support service for mobile, wireless publish/subscribe applications*, IEEE Transactions on Software Engineering, vol. 29, 2003, pp. 1059-1071.

[16] Perkins, *IP Mobility Support for IPv4*, RFC3344, http://tools.ietf.org/html/rfc3344.

[17] E. D. Kaplan and C. Hegarty, *Understanding GPS: Principles and Applications*, 2nd edition, Artech House Publishers, 2005.

[18] C. Savarese, J. M. Rabaey and K. Langendoen, *Robust positioning algorithms for distributed ad-hoc wireless sensor networks*, in USENIX ATEC, 2002, pp. 317-327.

[19] B. H. Cheng, L. Vandenberghe and K. Yao. *Distributed algorithm for node localization in wireless ad-hoc networks*, ACM Trans. Sensor Networks, vol.6, issue 1, Article 8, Dec 2009, pp. 1-20.

[20] A. Mislove and P. Druschel. *Providing administrative control and autonomy in structured peer-to-peer overlays*, Peer-to-Peer Systems III Lecture Notes in Computer Science, vol. 3279, 2005, pp. 162-172.

[21] I. Baumgart and S. Mies. *S/Kademlia: A practicable approach towards secure key-based routing*, in Proc. International Conference of Parallel and Distributed Systems, Dec 2007, pp.1-8.

[22] I. Baumgart, B. Heep, and S. Krause. *OverSim: A flexible overlay network simulation framework*, in Proc. IEEE Global Internet Symposium in conjunction with IEEE INFOCOM, Anchorage, AK, USA, May 2007, p. 79-84.

[23] A. Varga. *The OMNeT++ discrete event simulation system*, in Proc. European Simulation Multiconference (ESM), Prague, Czech Republic, Jun 2001.