

# Just-in-Time Delivery of Events in Event Notification Service Systems for Mobile Users

Chit Htay Lwin<sup>1</sup>, Hrushikesh Mohanty<sup>1</sup>, and R. K. Ghosh<sup>2</sup>

<sup>1</sup> Department of CIS, University of Hyderabad, Hyderabad, India  
{mc01pc08, hmcs}@uohyd.ernet.in

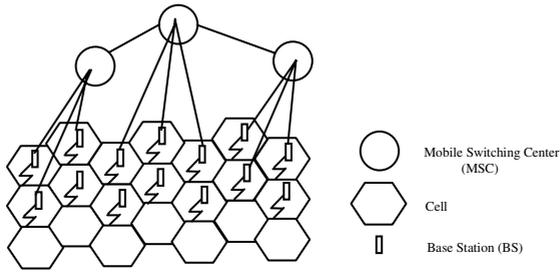
<sup>2</sup> Department of CSE, IIT Kanpur, Kanpur, India  
rkg@cse.iitk.ac.in

**Abstract.** Advances in wireless network technology and the increasing number of handheld mobile devices enable service providers to offer several services to mobile users. Event notification service (ENS) system is a useful mean to push information to selective users and the same service can also be extended to mobile users. But mobility may cause delay in transferring subscriptions from the previous event server to the current event server and establishing new routes of notifications. This delay is not tolerable for many real-life applications, like stock exchanges, auction and traffic condition monitoring. In this paper, we propose an algorithm that ensures the timely delivery of published events to all interested mobile users with the help of regional route map which is a physical route map of a small neighborhood of the locations of mobile hosts.

## 1 Introduction

The development of portable computing devices such as notebook computers, personal digital assistants (PDAs) and 3G cellular phones along with wireless communication technology enable people to carry computational power with them while they are changing their physical locations in space. In an event notification service (ENS) system, producers publish events and consumers subscribe to them. If a consumer subscribes an event, an ENS system delivers all matched notifications that are published by any producer until the consumer cancels the respective subscription. An ENS system provides time decoupling and space decoupling [5]. The loose coupling of producers and consumers is the primary advantage of ENS systems. Although existing push-based technologies described in [7] and WAP push technology may enable mobile users to receive timely information, but they are not scalable as their services are centralized.

The event notification service approach can be used in many applications and many of them such as stock trading systems, auction systems, emergency notifications, banking, reservation systems and traffic condition monitoring systems need to manage deadline bound data items. Time bound applications require more functionality such as deadline constraints that enable scheduling of event delivery. A user may be willing to receive sports and news information with delay by a few minutes. But it may not tolerate delays or out of sequence in quotes for stock prices.



**Fig. 1.** A cellular mobile architecture

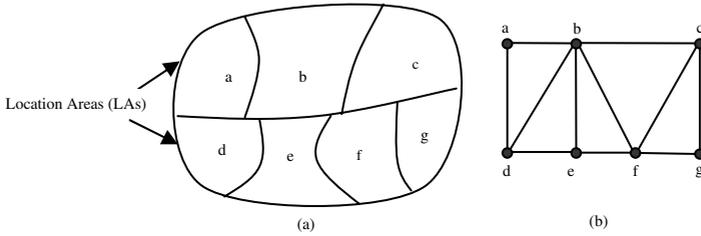
As an event notification service (ENS) system follows multicast communication protocol, it allows a producer to send the same event to many interested consumers with only one publishing operation. It can cope with dynamically changing operational environment where producers and consumers frequently disconnect and reconnect, as the system can operate with loose coupling of producers and consumers. Hence, an ENS system is a good candidate for a variety of e-commerce and m-commerce applications for mobile users. Huang [9] emphasizes on the issues that make an ENS system adaptable to dynamic changes occurring in a mobile computing environment. Several implementations of ENS systems (e.g. SIENA [1], JEDI [3], REBECA [6] and Elvin [12]) provide mobility support for delivery of events to the mobile users. Generally in these ENS systems, events notified to a consumer are stored at its event server (ES) located at the consumer's previous location during disconnection. On reconnection, the stored events are forwarded to the consumer's new location. Obviously mobile users expect uninterrupted service. In order to achieve it, users' subscriptions are not only to be registered at their new locations, but new routes need to be established at their new locations. This causes latency in delivery of events at new locations. In this paper we propose a solution that improves event delivery process for mobile users. The proposed solution ensures just-in-time event delivery i.e., a mobile user can receive notifications as soon as he/she reaches at new location. The rest of the paper is organized as follows. Section 2 describes the system model of mobile network for ENS systems, location management scheme, and, definitions and notations used in this paper. The proposed pre-handoff approach is presented in section 3. Section 4 describes the basic algorithm of location-based pre-handoff approach and section 5 concludes the paper.

## 2 System Model and Definition

### 2.1 System Model

We choose a model used in mobile cellular environment which is a collection of geometric areas called cells (Figure 1). Each cell is serviced by a base station (BS) located at its center [4]. Mobile terminals are connected to the network via the base

stations. Cells can have different sizes: picocells are used in indoor environments; microcells and macrocells are used within cities and rural areas and highways [13].



**Fig. 2.** (a) Regional map with LAs (b) Graph model showing the interconnections of LAs

The network is assumed to be divided into a few big location areas (LA) as described in [8]. The graph model is used for representing interconnections of LAs as shown in Figure 2. Each LA, consisting of many cells (about 50 to 100), is served by a mobile switching center (MSC). An MSC acts as a gateway of the mobile network to the existing wired network. The base stations of a LA are connected to a MSC which stores the location and service information for each registered local mobile host. Tracking of the current location of a mobile host, involves two basic operations, terminal paging and location update and the network determines the mobile host's exact location within the cell granularity. This approach determines the mobile host's exact location within the cell granularity. Many works have been done on location management schemes [4]. But in ENS systems, the routing of events is dependent on event content rather than particular destination address. Hence location updates of mobile hosts only occur within a LA served by their event server.

Usually event servers (ESs) are implemented on separate servers. However, MSCs with enough computing resources can also be used as event servers. In the following sections, we use the term event server (ES) instead of MSC to be in tune with discussion concerning event notification service systems. We assume that all connections are bi-directional. Therefore, the abstract architectural model is an undirected graph.

## 2.2 Subscription and Notification

An ES checks an event against all subscriptions of its routing table to forward it to neighboring ESs or to deliver it to the local consumers. Registration of subscriptions by each consumer at every ES leads to explosion of information as well as traffic. This problem can be managed by using subscriptions covering test as described in [2, 11].

Each ES maintains a routing table (RT) for propagation of subscriptions and notifications. Routing table (RT) is a collection of tuples {<subscription>, <consumers>, <incoming servers>, <outgoing servers>};

i.e.,  $RT := \{ \langle S, C\_ID, IS, OS \rangle \}$ .

In RT, the consumers associated to a subscription S mean the local consumers who subscribe subscription S. Incoming servers of subscription S are the ESs from which subscription S is received, and outgoing servers of subscription S are the ESs to which S is forwarded to. As and when a mobile user ceases interest to an event, it can unsubscribe that event. If a notification N (an event) is received from a producer, local ES matches with all subscriptions stored in RT. The results are:

1. a set of consumers residing in the area covered by local event server, and,
2. a set of event servers to which N needs to be forwarded.

A notification N matches a subscription S if S covers N ( $N \subseteq S$ ). If RT has subscription S matched with N (i.e.  $\exists S \text{ in } RT \mid N \subseteq S$ ), the local ES delivers N to the consumers  $\{(RT.S.C\_ID)\}$  and forwards N to event servers  $\{(RT.S.IS)\}$ .

### 2.3 Location Table and Border Cell Table

Each ES maintains mobile hosts location table (MLT), and local border cell table (BCT) besides the routing table (RT). A MLT is a collection of tuples  $\{\langle \text{mobile host ID} \rangle, \langle \text{cell ID} \rangle\}$ . In an event notification service system, routing paths of events depend on content of events. Hence, location update occurs within a LA served by an ES and each ES updates its mobile host location table (MLT) every time a mobile host crosses the boundary of a cell.

Each ES also maintains a border cell table (BCT), a list of neighboring ESs relative to each border cell. Border cells mean the cells which are near the boundary of other regions. Hence, BCT is a collection of tuples  $\{\langle \text{border cell} \rangle, \langle \text{neighboring event servers} \rangle\}$ , i.e.,  $BCT = \{\langle CB \rangle, \langle NES \rangle\}$ . By ‘.’ extension convention,  $BCT \cdot CB_k \cdot NES$ , denotes the neighboring ESs with respect to the cell  $CB_k$ . The set of neighbouring cells adjacent to a border cell can be found statically from the regional route map described in subsection 2.1. We can use a pre-declared movement pattern of the mobile consumer; and employ certain heuristics [8] to predict the possible cells where a mobile consumer may show up from a border cell of a location area.

## 3 Pre-handoff Approach

The pre-handoff is a proactive strategy which initiates updates of routing and location tables (RT and MLT) of LAs to which a mobile host may visit during its travel. The main idea behind pre-handoff is to create virtual consumers at a set of expected locations of a mobile consumer in order to set up flow of notification before the mobile consumer reaches any of those locations. Loosely speaking, a virtual consumer is created on all ESs to which the mobile host is expected to connect in the “near” future. The set of virtual clients must change when the mobile host moves.

In an event notification service system, all the subscriptions of the users registered are recorded explicitly or implicitly at all ESs. Hence locations of event producers do not have any direct bearing in delivering of notifications and we do not take mobility of event producers in account. In the proposed approach, we assume that events are

delivered in FIFO order with respect to the producer and that no events are duplicated and lost. To ensure ordering of events, [10] has proposed a causal ordering algorithm. Pre-handoff technique can successfully provide just-in-time service if and only if travel time of host from current border cell to that of new ES is more than equal the time for establishing paths of notifications at new ES.

### 3.1 Location-Based Pre-handoff

If an ES of a moving host executes pre-handoff at every neighboring ESs, it causes substantially unnecessary overheads. In order to avoid this problem, the proposed algorithm chooses ESs for pre-handoff based on locations of mobile hosts.

If a mobile consumer keeps moving in the interior cells of a LA, then the event server for the LA will be able to directly deliver the related notifications to the consumer. But if the consumer is moving around the border of a LA, it is likely that he/she may stray into a neighbouring LA. So we need to consider strategies to deliver the notifications related to the mobile consumer moving around the border cells of a LA should it ever stray into a cell under a neighbouring LA. It is done as follows. Each event server executes pre-handoff for mobile consumers which are moving in its border cells. As described in subsection 2.3, each event server maintains a local border cell table (BCT) that stores border cells and their adjacent LAs (i.e., neighboring ESs). So, an ES can use BCT in order to determine the neighboring ESs on which pre-handoff processes have to be executed on behalf of a moving consumer. The proposed mechanism creates virtual consumers (VC) and pre-subscribes at few ESs of possible future locations. When a mobile consumer ( $MH_s$ ) gets connected to a new ES on which a VC is already running, it finds that its subscriptions exist and routing paths are already established. This ensures that  $MH_s$  can receive matched notifications immediately after connecting to the new ES.

Each ES needs to cancel pre-handoff processes of a mobile consumer  $MH_s$  in the following cases:

1. The  $MH_s$  is not in one of its border cells. It has to tackle two possible sub\_cases:
  - $MH_s$  moves from a border cell to a non-border cell within current LA.
  - $MH_s$  moves to a border cell located within another LA.
2. The  $MH_s$  stops moving (i.e., it does not come out of a border cell) although the mobile is in a border cell.

In case 1, if  $MH_s$  has moved from a border cell to an interior cell under the current LA there is no need to execute a pre-handoff. Also if  $MH_s$  already move to a border cell under a new LA, unless pre-handoff is cancelled, it may lead to ping-pong situation. In case 2, the algorithm also needs to cancel pre-handoff processes. We can define time bound  $T$  to determine whether the host is moving or not. If a mobile host stays in a border cell greater than time  $T$ , the algorithm assumes that the mobile host is not moving (i.e.,  $flag\_mov(MH_s) = 0$ ).

When a mobile consumer  $MH_s$  connects to a new ES, it must handle one of the following two cases.

1. The new event server has a virtual consumer of  $MH_s$
2. The new event server does not have a virtual consumer of  $MH_s$ .

In case 1, the new event server need not execute normal handoff process which includes transferring subscriptions and re-subscribing and it can deliver matched notifications to  $MH_s$  immediately after connection. In case 2, the new event server did not create the virtual consumer of  $MH_s$  due to following reasons:

1. The mobile consumer  $MH_s$  was disconnected for long time and it reconnects to another event server. Hence, the new ES does not have a matching VC for  $MH_s$ .
2. The mobile consumer  $MH_s$  was in a border cell for long time (i.e., greater than  $T$ ) and so the pre-handoff process was cancelled.

## 4 Algorithm

A host on reaching a cell belongs to a new LA registers at the ES of LA. Registration is triggered by a border cell of the LA from where  $MH_s$  has migrated after a pre-handoff, or it may be initiated by some cell in a new LA where  $MH_s$  pops up independently. In first case a surrogate of  $MH_s$  is already up under the new ES. So  $MH_s$  just needs to replace its surrogate. In the second case, a normal handoff should be initiated by the event server of LA where  $MH_s$  shows up.

Suppose a mobile consumer  $MH_s$  moves from the location area (LA) of  $ES_i$  to the LA of  $ES_j$ . Before the  $MH_s$  connects to  $ES_j$ , it must pass through a border cell  $CB_k$  of  $ES_i$ . When the border cell  $CB_k$  sends *register* ( $MH_s, ES_j, CB_k$ ) to its event server  $ES_i$ ,  $ES_i$  executes pre-handoff for  $MH_s$ .  $ES_i$  transfers all subscriptions of  $MH_s$  using *Preforward* ( $MH_s, ES_j, S$ ) to a set of some neighboring ESs,  $e \in E$ , (i.e.,  $E = BCT-CB_k-NES$ ) including  $ES_j$ . Each of the ESs after receiving *Preforward* message creates a virtual consumer  $VC\_MH_s$  and forwards the subscriptions of  $MH_s$  to related ESs to establish a new route of notifications for  $MH_s$ . After sending *Preforward* ( $MH_s, ES_j, S$ ) to  $e$ ,  $ES_i$  needs to send *Pre\_Subforward* ( $MH_s, ES_j, S$ ) to  $e$  whenever it receives *Subscribe* ( $S, MH_s$ ) from  $MH_s$ . On receiving *Pre\_Subforward* ( $MH_s, ES_j, S$ ), each ES forwards  $S$  to related neighboring ESs to update routing paths of notifications. For two cases that need to cancel pre-handoff,  $ES_i$  sends *Cancel\_Preforward* to those ESs which have VC of  $MH_s$ . When  $MH_s$  connects to  $ES_j$  which already has  $VC\_MH_s$ ,  $ES_j$  changes  $VC\_MH_s$  as its local consumer and deliver notifications to  $MH_s$  without delay. If  $ES_j$  does not have  $VC\_MH_s$ ,  $ES_j$  initiates normal handoff process as described in [10]. The basic location-based pre-handoff algorithm is described in Figure 3. The data structures used in the algorithm are as follows:

- $cell\_MH_s$  = current cell of mobile consumer  $MH_s$ .
- $SSUBQ\_MH_s$  = queue to store subscriptions sent by local mobile consumer  $MH_s$
- $CB(ES_i) = \{C_1, C_2, \dots, C_k\}$  = a set of all border cells of  $ES_i$ .
- $flag\_mov(MH_s) = 1$  if mobile host  $MH_s$  is moving
- $VC(ES_{ij}) = ES_i$ 's awareness of virtual consumers created at  $ES_j$  for possible movements of hosts from  $ES_i$  to  $ES_j$ .
- $VC(ES_j) =$  all virtual consumers located at a server  $ES_j$ .
- $Local(ES_j) =$  a set of local mobile hosts at  $ES_j$ .

**Let  $MH_s$  moves from the LA of  $ES_s$  to the LA of  $ES_e$ .**  
**Algorithm A1:: Executed at event server  $ES_e$  where  $MH_s$  is residing.**

```

On receiving Register ( $MH_s, ES_s, cell\_MH_s$ ) /* $ES_k$  means the last connected ES */
if ( $MH_s \in Local(ES_e)$ )
    if ( $cell\_MH_s \in CB(ES_e) \wedge flag\_mov(MH_s) = 1$ )
        (a)  $E = BCT \cdot cell\_MH_s \cdot NES$ 
        For all  $ES_c \in E$  do
            if ( $MH_s \notin VC(ES_c)$ )
                (1) Send Pre_handoffRequest ( $ES_c, MH_s$ ) to  $ES_c$ ;
                (2) On receiving Enable_prehandoff ( $ES_c, MH_s$ )
                    Send Preforward ( $MH_s, ES_c, SSUBQ\_MH_s$ ) to  $ES_c$ ;
            endif
        endfor
        (b) On receiving Subscribe ( $S, MH_s$ ) /*receiving S from  $MH_s$ */
            Send Pre_Subforward ( $MH_s, ES_c, S$ ) to  $ES_c$ ;
    else
        Send Cancel_Preforward ( $MH_s, ES_s$ ) to  $ES_c$ ; /*Cancel pre-handoff */
    endif
endif

else
    if ( $VC\_MH_s \in VC(ES_e)$ ) /*VC is already up for  $MH_s$  */
        (1)  $Local(ES_e) = Local(ES_e) \cup MH_s$ ;
        (2)  $VC(ES_e) = VC(ES_e) - VC\_MH_s$ ;
    else
        Execute normal handoff process;
    endif
endif

```

**Algorithm A2:: Executed at event server  $ES_e$ .**

- I. On receiving **Pre\_handoffRequest** ( $ES_c, MH_s$ ),
  - (1)  $VC(ES_e) = VC(ES_e) \cup VC\_MH_s$ ;
  - (2) Send **Enable\_prehandoff** ( $ES_c, MH_s$ ) to  $ES_c$ ;
- II. On receiving **Preforward** ( $MH_s, ES_c, SSUBQ\_MH_s$ ),
  - (1) Update routing table for all subscriptions  $S \in SSUBQ\_MH_s$ ;
- III. On receiving **Pre\_Subforward** ( $MH_s, ES_c, S$ ),
  - (1) Store S in  $SSUBQ\_MH_s$ ;
  - (2) Forward S to neighboring event servers;
- IV. On receiving **Cancel\_Preforward** ( $MH_s, ES_s$ ),
  - (1) Unsubscribe each  $S \in SSUBQ\_MH_s$ ;
  - (2)  $VC(ES_e) = VC(ES_e) - VC\_MH_s$ ;

**Fig. 3.** Basic algorithm for location-based pre-handoff

## 5 Conclusion

Depending on the application domains, event notification service systems differ in architectures and in features they support. In this paper we have presented a pre-handoff approach for just-in-time delivery of events in an event notification service

system for mobile environment. Handoff process may cause some delay in event notification when a mobile user change its location. This delay is due to transfer of subscriptions of the mobile consumer to the new ES and re-subscribing these subscriptions to establish new routes for notifications. The proposed algorithm is designed to reduce this delay by using pre-handoff by predicting a small set of probable destination locations of the mobile host. Currently, we are studying the performance of our algorithm in simulation environment.

In case of location based services (i.e., when notifications are sensitized to location properties), our approach cannot be applied directly because the same subscription may match different events in different locations. Hence, in pre handoff approach when an event server forwards location-based subscriptions of virtual consumers, it is necessary to qualify location attributes of these subscriptions with respect to possible destination locations.

In our approach, pre-handoff is executed only by event servers and event servers re-subscribe subscriptions on behalf of mobile consumers. Hence this approach is suitable for mobile environment with constraints such as low power consumption, low computing power of mobile devices, and low bandwidth of wireless links. The proposed algorithm can be extended to execute pre-handoff at an exact location area to which a mobile consumer will move by using the direction and speed of the moving mobile host or movement pattern of mobile hosts [8].

## References

1. Caporuscio, M., Carzaniga, A., and Wolf, A. L.: An Experience in Evaluating Publish/Subscribe Services in a Wireless Network. In Proceedings of Third International Workshop on Software and Performance, Rome, Italy (July 2002)
2. Caarzaniga, A., Rosenblum, D. S., Wolf, A. L.: Design and Evaluation of a Wide-Area Event Notification Service. *ACM Transactions on Computer Systems* 19(3) (August 2001) 332-383
3. Cugola, G., Di Nitto, E., and Fuggetta, A.: The JEDI Event-based Infrastructure and Its Application to the Development of the OPSS WFMS. *IEEE Transactions on Software Engineering* 9(27) (September 2001) 827-850
4. DasBit, S. and Mitra, S.: Challenges of Computing in Mobile Cellular Environment-a Survey. *Computer Communications*, Elsevier, 26(18) (December 2003) 2090-2105
5. Eugster, P. T., Guerraoui, R., and Damm, C. H.: On Objects and Events. In Proceedings of OOPSLA, Tampa Bay, USA (October 2001)
6. Fiege, L., Gärtner, F. C., Kasten, O., and Zeidler, A.: Supporting Mobility in Content-based Publish/Subscribe Middleware. In Proceedings of IFIP/ACM Middleware 2003 (June 2003)
7. Franklin, M., Zdonik, S.: Data in your face: push technology in perspective. In Proceedings of ACM SIGMOD International Conference on Management of Data, Seattle, WA, (1998)
8. Ghosh, R. K., Rayanchu, S. K., and Mohanty, H.: Location Management by Movement Prediction Using Mobility Patterns and Regional Route Maps. In Proceedings of IWDC'03, Vol. 2918, LNCS, Springer-Verlag, (2003) 153-162

9. Huang, Y., and Garcia-Molina, H.: Publish/Subscribe in a Mobile Environment. In Proceedings of the 2nd ACM International Workshop on Data Engineering for Wireless and Mobile Access, Santa Barbara, CA (May 2001)
10. Lwin, C., H., Mohanty, H., Ghosh, R. K.: Causal Ordering in Event Notification Service Systems for Mobile Users. In Proceedings of ITCC'04, IEEE, Las Vegas, USA (April 2004) 735-740
11. Muhl, G., Fiege, L., Buchmann, A.: Filter Similarities in Content-based Publish/Subscribe Systems. In Proceedings of ARCS'02, LNCS, Springer-Verlag, (2002)
12. Sutton, P., Arkins, R., and Segall, B.: Disconnectedness- Transparent Information Delivery for Mobile and Invisible Computing. In Proceedings of CCGrid'01, IEEE, (May 2001)
13. Wong, V. W.-S., and Leung, V. C. M.: Location Management for Next Generation Personal Communication Networks. IEEE Network, Special Issue on Next Generation Wireless Broadband Networks (2000)