An Approach to Developing Information Dissemination Service for Ubiquitous Computing Applications

Stephen S. Yau and Sheikh I. Ahamed
Computer Science and Engineering Department
Arizona State University
Tempe, AZ 85287, USA
{yau, iq}@asu.edu

Abstract

In many ubiquitous computing (ubicomp) applications, such as smart classrooms, hospitals, construction sites, stadiums, shopping malls, emergency services, law enforcement, convention centers, battlefields, and inhospitable physical environments, information dissemination service is needed for distributing information from one or more devices to a large number of devices. These systems normally possess the characteristics of autonomous decentralized systems (ADS). Information dissemination for ubicomp applications can be achieved by autonomous coordination, but complex interactions and autonomous coordination of these devices depend upon local and neighboring context and/or situation. Due to the mobility of the devices, unknown destinations for information from a source, and low computational power of the devices for routing, it is difficult to provide information dissemination service for ubicomp applications. Reconfigurable context-sensitive middleware (RCSM) has been used for ADS applications in ubicomp environments since it can provide context-sensitivity, situation-awareness, ad hoc communication and ADS properties. In this paper, an approach to developing information dissemination service for ubicomp applications, which can be incorporated in RCSM, is presented.

Keywords: Information dissemination service, ubiquitous computing (ubicomp), situation-awareness, context-sensitivity, Autonomous Decentralized Systems (ADS), Reconfigurable Context-Sensitive Middleware (RCSM).

1 Introduction

Recent rapid progress in inexpensive, short range, and low-power wireless communication hardware and network standards will soon make ubiquitous computing (ubicomp) applications possible. Ubiquitous computing means computing everywhere and making computing and communication essentially transparent to the user [1]. Many ubicomp applications, such as smart classrooms, hospitals, construction sites, stadiums, shopping malls, emergency services, law enforcement, convention centers, battlefields, and inhospitable physical environments, frequently require information dissemination service, which deals with the distribution of information from one or more devices to a large number of devices. The information dissemination of ubicomp applications has the characteristics of context sensitivity, situation awareness, mobility, equality, locality and self-containment. The later three are also characteristics of autonomous decentralized systems (ADS) [2]. Context sensitivity (or context awareness) is the ability of a device to detect its current context and changes in contextual data [3]. Situation awareness is the capability of a device to capture and analyze the relationship among multiple contexts and actions over a period of time [4]. Besides context sensitivity and situation awareness, these devices dynamically form mobile ad hoc networks (MANET) to engage in context-sensitive communications [5].

For information dissemination in any computing environment, broadcasting [6] can be used. It is impractical to use broadcasting to disseminate information in ubicomp environments because every node relays every received message to every neighbor in broadcasting, which requires much computing resources. The devices in ubicomp environments have limited computing power. Event service [7] can be used for information dissemination in distributed computing applications, but is not suitable in ubicomp applications. Because the devices in ubicomp applications are mobile and if a device working as a channel goes out of the vicinity of another device communicating with this device, the entire information dissemination process will collapse. Routing algorithms [8] are not suitable for information dissemination in ubicomp applications either because routing algorithms require much computational power to calculate routing paths and memory to store routing information as tables due to the mobility of the devices. Moreover, a routing algorithm creates a path between a source and known destination, which is likely unknown in an ubicomp application. Although some research has been done in information dissemination for the short-range ad hoc wireless networks, such as Metadata negotiation [9] and diffusion technique [10], situation-aware communications and mobility of the devices are not addressed.

We have shown how Reconfigurable Context-Sensitive Middleware (RCSM) can be used for ADS applications in mobile ad hoc networks and ubicomp environments [3]. In this paper, we will present an approach to developing information dissemination service for ubicomp applications, which can be incorporated as a service of RCSM. We will use shopping mall as an illustrative example.

2 Coordination States and Coordination Rules

Our approach to developing information dissemination service in ubicomp applications uses situations to establish coordination among the devices so that each device will not receive the same information more than once. Our information dissemination service (IDS) is an add-on module to RCSM [3] for disseminating information as shown in Figure 1. In our approach, we use coordination states and coordination rules for information dissemination in ubicomp applications. We will first describe coordination states and coordination rules, and then present our approach in Section 3.

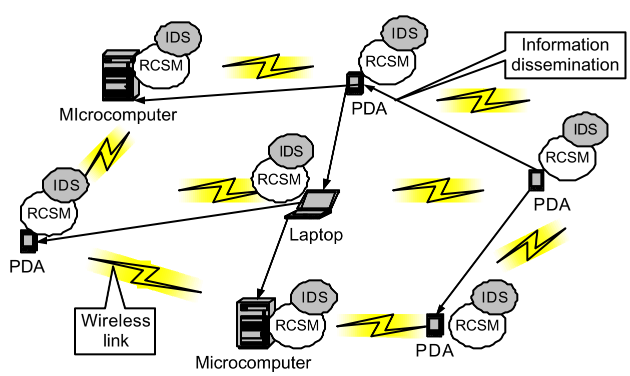

Figure 1: Addition of information dissemination service in a ubicomp environment using RCSM.

Coordination state: It is the condition of a device at any given time to interact with other devices for information dissemination in a ubicomp application.

To illustrate a coordination state, let us consider a future shopping mall, where shoppers will have their own Personal Digital Assistants (PDAs). Shops and shopping mall authority have computing devices, such as desktop computers (DCs), with wireless connections to other computing devices. These computers can also work as relays for information dissemination. The PDAs and DCs collaborate by dynamically forming mobile ad hoc networks in the shopping mall. In a shopping mall, a shopper might want to find all current bargain deals using his PDA when he is in the mall. In this example, the condition of the shopper's PDA is uploading query when the shopper is inside the mall. Uploading query (CSi) is the coordination state of the shopper's PDA. Similarly, downloading query states (CSj) are the coordination states of neighboring DCs of the shopper's PDA since they receive the query from the shopper's PDA.

Coordination rule*:* It is the rule of managing interactions of a device with other devices by changing the coordination state of the device to another coordination state for information dissemination in a ubicomp application. Each coordination rule of a device has a current coordination state and a next coordination state. A coordination rule of a device changes its current coordination state to a next coordination state.

From the above example, a shopper's PDA is in uploading query state (CSi) and the shopping mall's DCs are in downloading query states (CSj). A neighboring DC of the shopper's PDA changes its coordination state to CSi from CSj as soon as it receives the query. Here, CSj is the current coordination state of the neighboring device and CSi is the next coordination state. The other DCs, which are in downloading query state (CSj), will also change their state to CSi after receiving the query. In this way, query information is disseminated. The coordination rule for a neighboring DC is CSj→ CSi, which means that upon receiving coordination state CSi, the DC changes its coordination state to CSi from CSj.

Coordination state match: If a device sends its coordination state to its neighboring device and has a match with the next coordination state of the neighboring device's coordination rule, is called a coordination state match.

In the above shopping mall example, the coordination rule of a neighboring DC is CSj→ CSi, where current coordination state is CSj and next coordination state is CSi. A shopper's PDA sends its coordination state CSi to a neighboring DC. Upon receiving the coordination state CSi, the neighboring DC matches this coordination state CSi to its next coordination state of its coordination rule. Here, this is a coordination state match for CSi and the neighboring DC changes its coordination state from its coordination state from CSj to CSi.

Proceedings of the Sixth International Symposium on Autonomous Decentralized Systems (ISADS'03)
0-7695-1876-1/03 $17.00 © 2003 IEEE

3 Our Approach to Developing Ubicomp Information Dissemination Service

Our approach to developing information dissemination service for ubicomp applications can be depicted in Figure 2.

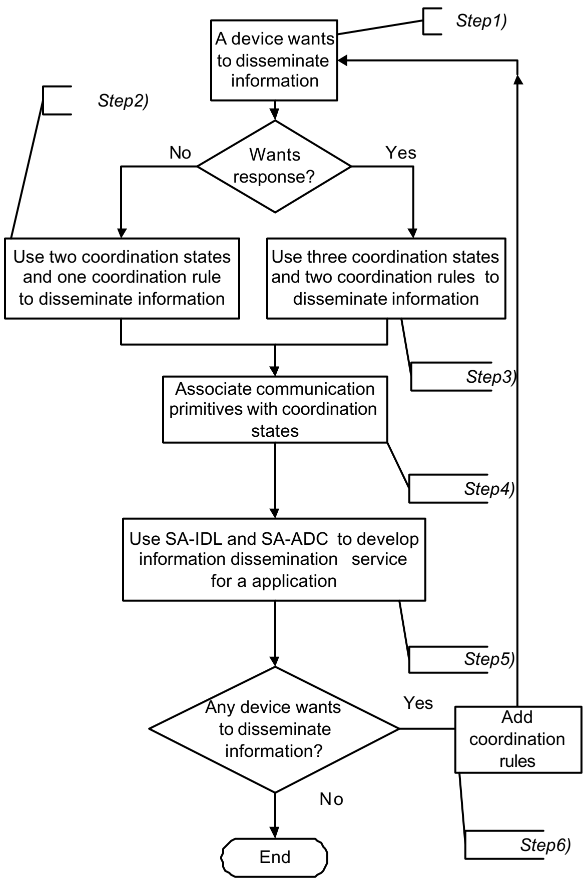

Figure 2: Our approach to providing information dissemination service for a ubicomp application

We will explain each step of our approach as follows:

Step 1) **A device wants to disseminate information.**

Step 2) **Disseminate information without response:** It means that an information source device does not need response from other devices after disseminating information. This mechanism needs two coordination states: uploading coordination state CSi and downloading coordination state CSj. It also needs the coordination rule CSj→CSi, which means that upon receiving the coordination state CSi, a device changes its coordination state from CSj to CSi. To disseminate information, an information source device sends its coordination state to all of its neighbors. Upon receiving the coordination state from an information source device, a neighboring device checks its coordination rule. If it is a coordination state match, the neighboring device requests the information from the information source device. Then the information source device sends information to the neighboring device. The neighboring device receives the information and changes its coordination state to the coordination state of the information source device. This neighboring device will not receive information anymore, since it has the information now and it has already changes it coordination state. In this way, a neighboring device does not receive actual information more than once. Then, this neighboring device sends its coordination state to its neighboring devices and the process is repeated until all the other devices receive the information. The details of this mechanism is described as follows:

a) When an information source device D_s wants to disseminate information, it sends its coordination state CSi to all of its neighbors that are in its communication range. Let us assume that D_r is a neighboring device of D_s. The coordination state of D_r is CSj. D_s sends its coordination state CSi to its neighboring device D_r.

b) A neighboring device D_r checks its coordination rule (CSj→CSi if it receives CSi) after receiving the coordination state CSi from the device D_s. If the coordination state CSi matches the coordination rule of D_r, then D_r asks D_s to send the information and D_r changes its coordination state from CSj to CSi by receiving information from D_s. Now, D_r disseminates information to its neighbors in a similar way. If the coordination state of D_r does not match the coordinate state in the coordination rule, no change will be made in the mechanism of information dissemination.

c) If a device D_i receives coordination states from two or more neighboring devices, D_i requests information from the device, whose coordination state has arrived first at Di.

d) If a device D_i moves, but stays in the same vicinity of other devices (i.e. neighboring devices remain the same), there will be no change in the information dissemination mechanism. If the device D_i leaves the vicinity (i.e. out of range) of other devices, its neighboring device D_j cannot receive information. The D_j changes its state back to CSt-1, where CSt-1 is the previous coordination state of D_j. D_j distributes its coordination states to its neighbors and requests for information. Upon receiving information from a neighboring device, D_i changes its coordinate state.

The above mechanism for information dissemination using coordination states and coordination rule addresses the self-containment, and equality of ADS since each device of ubicomp can manage itself without being directed by or giving direction to others, and every device in ubicomp is self-contained in functions to manage it and coordinate with others.

Step 3) **Disseminate information with response:** It means that an information source device needs response from other devices after disseminating information. This mechanism has two parts: information dissemination without response and reception of response. The first part is similar to information dissemination without response, which needs uploading coordination state CSi, and downloading coordination state CSj. It also need coordination rules CSj→CSi, which means that upon receiving the coordination state CSi, change the coordination state to CSi. This part uses the same mechanism of *Step 2* (disseminating information without response). Reception of response needs uploading response CSk and the coordinate rule CSi→CSk This means that upon receiving the coordination state CSk, the current coordination state of the device changes to CSk. When a device wants to send response, it sends the uploading response CSk to its neighboring devices. The neighboring device receives the coordination state CSk and matches the coordinate state of the coordination rule CSi→CSk. If it is a coordination state match with its coordination state of the neighboring device's rule, the neighboring device requests the response. Upon receiving the coordination state, the neighboring device changes its coordination state. In this way, response is sent to the information source device.

Step 4) **Associate coordination states with communication primitives:** Communication primitives are used to manage when and how to exchange the coordination states between two devices. The communication primitives are *send, receive, wait* and *forward.* The communication primitives are used to assign different coordination states to different devices. Moreover, changes of states are achieved through communication primitives. Each coordination state can be associated with one or more communication primitives. For example, uploading and downloading are associated with the *send* and *receive* primitives as shown in Table l.

Coordination states	Communication primitives
Downloading	Send, receive
Uploading	Send, receive

Table 1: Coordination states and communication primitives

Step 5) **Use SA-IDL and SA-ADC to develop information dissemination service:** We complete the development by generating the code for the situation-aware objects of information dissemination. Situation-aware interface reflects the application-specific situation-awareness requirements on what situations to detect and what actions to take in order to respond to the situations. To specify an application-specific situation-aware interface in a file, we use Situation-Aware Interface Definition Language (SA-IDL) [4]. An Situation-aware Adaptive Object Container (SA-ADC) [4] is automatically generated using the SA-IDL compiler for runtime analysis and detection of application-specific situations. The SA-IDL compiler also generates code of the situation-aware objects whose interfaces are defined in SA-IDL. We use SA-ADC to develop the applications with information dissemination service.

This SA-ADC of information dissemination can be a module of RCSM since it is generated using SA-IDL compiler of RCSM. It can be treated as an add-on module to RCSM for information dissemination.

This step addresses the locality of ADS for information dissemination since each device is able to manage itself and to coordinate with others based on situation. It also addresses situation awareness of information dissemination service.

Step 6) **Add coordination rules for multiple information source devices:** In this step, we assume that sending and receiving coordination states are atomic operations for information dissemination with multiple information sources to avoid conflict and synchronization problems. If there are multiple information dissemination sources, we add the coordination rules of multiple information source devices together. We need two coordination states and one coordination rule for information dissemination without response. Coordination states for the i^{th} information source device are CS1i and CS2i and one coordination rule CS12i→CS1i, where i=1, 2, 3,…, n.

We need three coordination states and two coordination rules for information dissemination with response. Coordination states for the i^{th} information source device are CS1i, CS2i and CS3i and coordination rules are CS2i→CS1i and CS1i→CS3i, where i=1, 2, 3, …, n.

4 An Illustrative Example

Consider the future shopping mall, where shoppers will have their own Personal Digital Assistants (PDAs). Shops and shopping mall authority have computing devices, such as desktop computers (DCs), with wireless connections to other computing devices. These computers can also work as relays for information dissemination. These PDAs and DCs collaborate by dynamically forming mobile ad hoc networks in the shopping mall. In a shopping mall, a shopper wants to know all the bargain deals of a specific item when he is in the shopping mall using his PDA. In this example, the condition of the shopper's PDA is searching query when the shopper is inside the mall. To show information dissemination service, we will take the requirement: A shopper wants to know all the bargain deals of a specific item when he is in the shopping mall using his PDA. In this requirement, the bargain deals of a specific item query will be disseminated.

A conceptual diagram of a future shopping mall is shown in Figure 3. To simplify explanation, let us assume that we have 9 devices in the shopping mall, which can be written as $\{D_1, D_2, D_3, \ldots D_9\}$. Here, D_1 is a PDA and D_2, D_3, $D_4 \ldots D_9$ are DCs. The neighbors of D_1 are D_2, D_3, D_7, which can be expressed as

$$N(D_1) = \{D_2, D_3, D_7\}.$$

Similarly, others DCs can be expressed as

$$N(D_2)=\{ D_1, D_4, D_7\},$$
$$N(D_3)=\{ D_1 , D_5, D_7\},$$
$$\ldots$$
$$N(D_9)=\{ D_5 , D_7 \}.$$

We assume that D_1, D_2, D_3, D_4, D_5, D_7, and D_9 have the middleware RCSM, including the add-on information dissemination service installed on each of these devices. D_6 and D_8 do not have the middleware RCSM, and hence D_6 and D_8, cannot participate in information dissemination.

Here, we will show how our approach provide information source in this example:

Step 1) We have the shopper's PDA D_1 as the single information dissemination source. When the shopper is inside the shopping mall, his PDA triggers the information dissemination.

Step 2) We do not need *Step 2)* in this example since information dissemination needs response.

Step 3) The shopper wants to know all the bargain deals of a specific item when he is in the shopping mall. This distributes the query and also wants to know the deals, which is information dissemination with response. The coordination states for information dissemination with response are: CS1: uploading query, CS2: downloading query, and CS3 uploading deals. The coordination rules are: Rule 1: CS2 → CS1: downloading query to uploading query, Rule 2: CS1 → CS3: uploading query to uploading the deals. The mechanism for information dissemination using coordination states and coordination rules is described below:

a) D_1 sends its coordination state CS1 to all of its neighboring devices D_2, D_3, D_7.

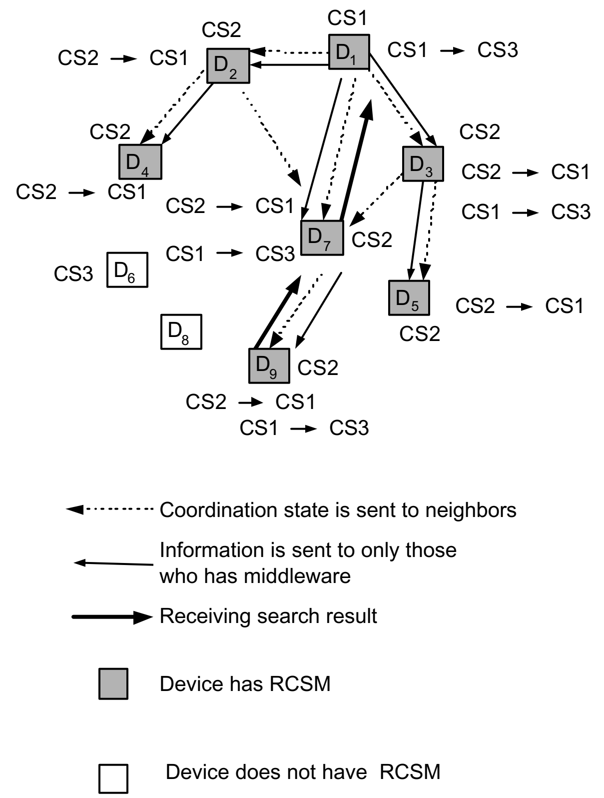

Figure 3: Information dissemination in a shopping mall

b) D_2 receives the coordination state CS1 from D_1 and checks its coordination rule and there is a state match in Rule 1. D_2 requests D_1 to send the query for deals information. D_2 receives information and changes its coordination state to CS_1. Similarly, D_3 and D_7 receive information and this process is repeated to disseminate information. Information is distributed from D_1 to D_2, D_3, D_7 and then D_7 to D_9, D_2 to D_4, and D_3 to D_5.

c) D_7 can receive information D_1, D_2 and D_3. If D_7 receives information from D_1, D_2 or D_3. The path D_1 to D_7 is chosen since D_7 receives coordination state first.

d) In Figure 4, D_2 is moving out of the vicinity of its neighbors and is out of communication range of its neighbors. D_2 distributes its coordination states to its neighbors. D_4 changes its state back to CS1 and receives information from D_7.

At D_9, the query for deals is matched and the deals will be relayed to D_7 using coordination rule (Rule 2). Then, the deals are relayed to D_1 using coordination rule (Rule 2) via D_7.

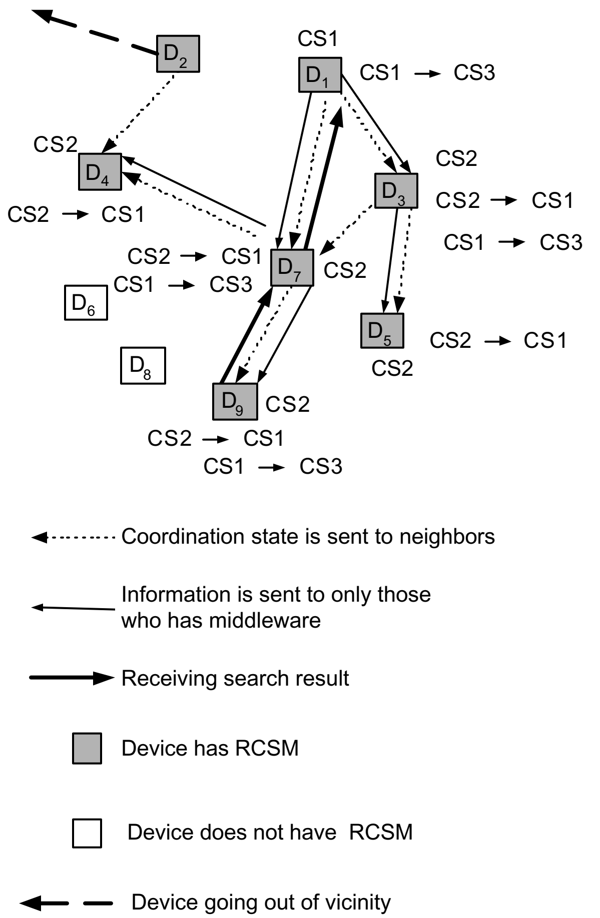

Figure 4: D_2 is moving out of the communication range of its neighbors

Step 4) We have 3 coordination states: downloading query, uploading query and uploading deals. Communication primitives of the above coordination states are given in Table 2:

Coordination states	Communication primitives
Uploading query	Send, receive
Downloading query	Send, receive
Uploading deals	Send, receive

Table 2: Coordination states and communication primitives

Step 5) SA-IDL is shown in Figure 5 for "A shopper wants to know the bargain deals of specific items when he is inside the shopping mall".

We generate SA-ADC using SA-IDL compiler of RCSM and use SA-ADC to develop the application, which use information dissemination service.

```
RCSM_Context context1 {
        string location;   }
ActionTuple action1 {
        string action_name;}
 interface ShoppingMall ['S', 'P', 'M']{
Situation   s1{ //shopper is in Mall currently
        (ForAny  t    in  [0,0])  context1.location  =
        "ShoppingMall"}
Situation s2{ //A shopper was Outside one minute ago
        (ForAny  t  in  [-2,-1])  context1.location  <>
        "ShoppingMall"}
Situation s3  { // A shopper's PDA has not queried for
last 30 minutes
        (ForAny t in [-30, 0] action1.action_name <>
        "FindDeals"}
Situation  s4{ // go query   s1 and s2 and s3 }
        [outgoing] [activate at situation s4 ]
        void  FindDeals() ;}
```

Figure 5: SA-IDL for the shopper's device

Step 6) In this example, we do not need this step since it has only a single information source device D_1.

5 Experimental Results

To show the performance of our approach, we have developed a simulation tool. We can assign different coordination rules and coordination states to various devices, both manually and randomly. To show that our approach works for a large number of devices for a single information source, we have randomly generated coordination states and coordination rules for information dissemination. In our simulation, we have changed the number of devices from 10 to 500 and generated random coordination states and coordination rules and collected experimental data for a single source device. The experimental data is given in Table 3.

In our approach to providing information dissemination service, we need short messages for coordinate states and long messages for information. First, an information source device sends coordination states to its neighboring devices using short messages. If the coordination state matches with the coordination states of the coordination rules of its neighboring devices, the information source device sends actual information to its neighbors. In this way, a neighboring device coordinates with its neighbors and sends information to its neighbors. Hence, a device in our approach does not need to send long messages for all neighbors. It sends long messages only for those neighbors having matched coordination states in their coordination rules. A device needs to send short messages for all of its neighbors.

Number of devices	Information dissemination using our approach		Information dissemination using broadcasting
	Number Short messages	Number Long messages	Number Long messages
10	30	3	30
20	76	4	76
50	430	9	430
100	1463	12	1463
200	5152	15	5152
500	39213	68	39213

Table 3: Experimental data for the information dissemination service using our approach and broadcasting for a single information source.

A device can disseminate information by sending long messages to all neighboring devices using broadcasting. In this case, each device needs to receive long messages more than once and the total number of long messages will be the same as the number of short messages of our approach. In our approach, the number of long messages is much smaller than the case using broadcasting because each device receives information only once. Hence, our approach uses less resources for providing information dissemination service. A performance graph is shown in Figure 6. The heavy line shows the total number of messages and the fine line shows the total number of messages required by using our approach. It is noted that the number of messages sent by all the devices using broadcasting mechanism increases rapidly as the number devices increases, while the number of message sent by all the devices using our approach increases little as the number of devices increases. This means that information dissemination using our approach is much more efficient than that using broadcasting.

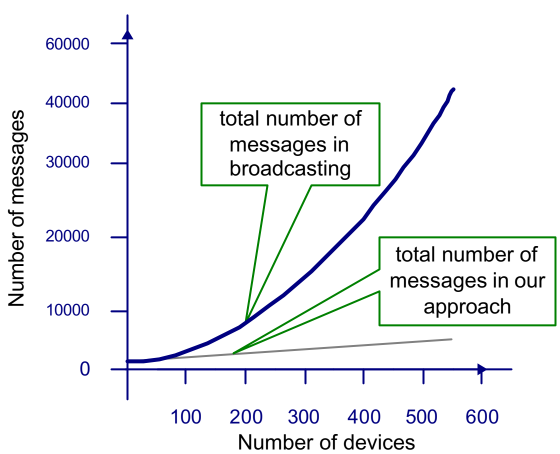

Figure 6: Comparison between broadcasting and information dissemination using our approach for a single information source

6 Discussion

In this paper, we have presented an approach to providing situation-aware information dissemination service for ubicomp applications. We have illustrated our approach using a shopping mall example. We have also shown with experimental data that our approach is much more efficient than broadcasting. In addition, we are implementing our approach in Windows CE environment using Platform Builder 3.0 and Embedded Visual studio 3.0 on CASSIOPEIA E-200 PDA as mobile devices. RCSM will be installed on each PDA and this information dissemination service will be installed as an add-on module to RCSM.

Future research in this area includes the evaluation of our approach for multiple information source devices. We will also explore the possibility of making our approach suitable for real-time distributed mobile ad hoc applications. Security, and other QoS aspects of information dissemination for ubicomp applications and interoperability of the RCSM with other middleware will also be explored.

Acknowledgement

This research is supported in part by National Science Foundation under grant number ANI-0123980. The authors would like to thank Fariaz Karim and Yu Wang for many helpful discussions.

References:

[1] M. Wesier, "Some computer science problems in ubiquitous computing", *Comm. ACM*, Vol. 36, No 7, July 1993, pp. 75-84.

[2] K. Mori, "Autonomous decentralized systems: concept, data field architecture, and future trends", *Proc. Int'l Symp. Autonomous Decentralized Systems (ISADS 1993*), March 1993, pp. 42-44.

[3] S. Yau and F. Karim, "Reconfigurable context-sensitive middleware for ADS applications in mobile ad hoc network environments", *Proc. 5th Int'l Symp. Autonomous Decentralized Systems (ISADS 2001)*, March 2001, pp. 319-326.

[4] S. Yau, Y. Wang and F. Karim, "Development of situation-aware application software for ubiquitous computing environment", *Proc. 26th Int'l Computer and Software*

Applications Conf. (COMPSAC 2002), August 2002, pp. 233-238.

[5] S. S. Yau and F. Karim, "An Adaptive Middleware for Context-Sensitive Communications for Real-Time Applications in Ubiquitous Computing Environments", to be published in *Real-Time Systems*, *The Int'l Journal of Time-Critical Computing Systems*, Kluwer Academic Publishers, Dordrecht, The Netherlands.

[6] B. Williams and T. Camp. "Comparision of Broadcasting Techniques for Mobile Ad Hoc Networks", *Proc. ACM Int'l Symposium on Mobile Ad Hoc Networking and Computing (Mobihoc),* 2002, pp.194-205.

[7] C Ma and J. Bacon, "COBEA: A Corba-based event architecture", *Proc. USENIX COOTS'98*, April 1998, pp. 117-131.

[8] E. Royer, and C. Toh, "A review of current routing protocols for ad hoc mobile wireless networks", *IEEE Personal Communications*, April 1999, pp. 46-55.

[9] W. Heinzelman, J. Kulik and H. Balakrishnan, "Adaptive protocols for information dissemination in wireless sensor networks," *Proc. 5th ACM/IEEE Mobicom Conf. (MobiCom '99)*, August 1999, pp. 174-185.

[10] C. Intanagonwiwat, R. Govindan and D. Estrin, "Directed diffusion: A scalable and robust communication paradigm for sensor networks", *Proc. 6th Annual Int'l Conf. on Mobile Computing and Networks (MobiCOM 2000)*, August 2000, pp. 56-67.

[11] S. Yau, F. Karim, Y. Wang, B. Wang, and S. Gupta "Reconfigurable context-sensitive middleware for pervasive computing", *IEEE Pervasive Computing,* 1(3), July-September 2002, IEEE Computer Society Press, pp. 33-40.

[12] S. Yau and F. Karim, "Context-sensitive middleware for real-time software in ubiquitous computing environments", *Proc. 4th IEEE Int'l Symp. Object Oriented Real-Time Distributed Computing (ISORC 2001*), May 2001, pp. 163-170.