

# DRIP: A Dynamic Voronoi Regions-Based Publish/Subscribe Protocol in Mobile Networks

Quan Yuan and Jie Wu

CSE Department, Florida Atlantic University, Boca Raton, FL 33431, USA

Email: {qyuan@, jie@cse.}fau.edu

**Abstract**—The publish/subscribe (pub/sub for short) paradigm is used to deliver events from a source to interested clients in an asynchronous way. Recently, extending a pub/sub system in wireless networks has become a promising topic. However, most existing works focus on pub/sub systems in infrastructured wireless networks. To adapt pub/sub systems to mobile ad hoc networks, we propose DRIP, a dynamic Voronoi region-based pub/sub protocol. In our design, the network is dynamically divided into several Voronoi regions after choosing proper nodes as broker nodes. Each broker node is used to collect subscriptions and detected events, as well as efficiently notify subscribers with matched events in its Voronoi region. Other nodes join their nearest broker nodes to submit subscriptions, publish events, and wait for notifications of their requested events. Broker nodes cooperate with each other for sharing subscriptions and useful events. Our proposal includes two major components: a Voronoi regions construction protocol, and a delivery mechanism that implements the pub/sub paradigm. The effectiveness of DRIP is demonstrated through comprehensive simulation studies.

## I. INTRODUCTION

It is challenging to combine the publish/subscribe (pub/sub for short) system [3] with mobile networks. The pub/sub system is used to connect the distributed information providers and consumers in an asynchronous way, where there are subscribers, publishers and brokers. The subscribers show their interest in certain events by submitting predefined subscriptions; the publishers issue newly detected events to the system; the brokers, which are generally custom servers, collect those subscriptions and events, match them and notify the subscribers of the matched events. In this way, it supports loosely coupled interaction in distributed environments, such as event notification systems and activity monitoring services.

Current pub/sub systems in mobile networks can be characterized into two types: infrastructured wireless network-based [4], [10] and mobile ad hoc network-based. The existing research mainly focus on the former one, which assumes that the backbone network is still wired and mobile users connect to the network by the wireless access points distributed along the backbone network. In contrast, we believe that the latter one applied in mobile ad hoc networks (ad hoc networks for short) is also a crucial research point in the effort to develop a reliable and efficient routing protocol to support pub/sub paradigm in mobile networks. Specially, it introduces the following challenges: 1) in mobile ad hoc networks, every node contributes equally and there are no custom device acting as brokers. Traditional pub/sub technologies are not applicable; 2) due to the mobility of the whole network,

mobility management becomes more complicated to deliver the delayed events to the right subscribers; 3) since it is difficult for nodes to self-organize into a strong structure, the multicast performance may decrease in mobile networks. Thus, efficient and reliable message delivery poses a greater challenge.

In this paper, we propose a dynamic Voronoi regions-based pub/sub scheme that adapts pub/sub systems to ad hoc networks. In particular, the network is dynamically divided into several Voronoi regions after selecting a fixed number of nodes as broker nodes. Each broker node is used to collect subscriptions and detected events, and also to notify subscribers with coinciding events in its Voronoi region. Other nodes join their nearest broker nodes to submit subscriptions, publish events, and wait for notifications of their requested events. Broker nodes cooperate with each other to share subscriptions and useful events. Our design addresses the following questions: 1) How to choose broker nodes, and how many broker nodes are needed? 2) How to dynamically divide the network into Voronoi regions? 3) How to implement the paradigm based on the constructed dynamic Voronoi regions?

The remainder is organized as follows. In Section II, we discuss some related work. Section III presents the overview of our design. Section IV describes the schemes to divide the network into Voronoi regions. Section V gives the mechanisms to implement the pub/sub process, and Section VI provides simulation results. We conclude our work in Section VII.

## II. RELATED WORK

Pub/sub systems in mobile networks introduce new challenges [6] because the link between any two nodes is not as stable as in fixed networks. To adapt pub/sub systems to mobile environments, several designs are proposed. Burcea et al. [2] propose a home-broker based protocol where each mobile client is assigned a home broker which maintains the subscriptions for the client. When the mobile client changes its home broker, a handoff protocol is utilized for event delivery between two brokers. Similarly, several other mobility management protocols [5], [8] are studied. Unfortunately, all of these protocols are based on infrastructured wireless networks, where the backbone network is still wired and only end users are mobile. In [6], Huang and Molina discuss a model with event sources, event brokers and event displays, and modify it to be feasible in mobile networks. However, this method is based on the assumption that broker servers

can always be organized into a multicast tree. An optimal pub/sub tree is proposed in [7] for routing events from the source to all interested recipients. Since each node knows its successors' subscriptions, flooding events can be avoided. However, the scalability of this approach is limited. As the network scale is growing or nodes' subscriptions change over time, tree maintenance incurs costs that are too high to be practical in the reality. In this paper, we propose a non-tree-based protocol to improve the scalability.

### III. SYSTEM MODEL

In DRIP, nodes in the network fall in two categories: *non-broker* and *broker*. Non-broker nodes work as subscribers and publishers. Broker nodes are special nodes that compose a small percentage of the whole network. In addition to being able to subscribe and publish events like non-broker nodes, broker nodes are also capable of collecting subscriptions and events from non-broker nodes, matching events, and notifying the interested subscribers with the matched events. To be strict, we assume that: a) each node in the ad hoc network has a unique identifier; b) initially, every node only knows its neighbors information and no pre-established routing structure is available; c) the delay of a message between two nodes is in direct ratio to their distance. The complete DRIP scheme consists of two components: organization rules to construct the dynamic Voronoi regions, and message delivery mechanisms implementing the pub/sub paradigm based on those Voronoi regions.

#### IV. THE DYNAMIC VORONOI REGIONS ARCHITECTURE

In this section, we present the organization rules which dynamically divide the whole ad hoc network into Voronoi regions. First we introduce the rule to generate Voronoi regions. Then we explain the mobility adaption of that rule.

##### A. Voronoi Regions Generation

The partition of the network into Voronoi regions is conducted in two phases. First, several nodes are selected as broker nodes in the network. Then, non-broker nodes pick the nearest broker node to register, and Voronoi regions are then formed in the network as a result. The process of selecting brokers is similar to cluster head selection. However, instead of utilizing lowest ID or maximum degree as the selection criterion, broker node selection is based on detecting ability. We assume that each node has a detecting ability  $\xi$ . Node  $i$ 's detecting ability  $\xi_i$  can be estimated by nodes' physical conditions, e.g., memory capacity and detecting power. Nodes with larger  $\xi$  can detect more events. We choose  $k$  nodes as broker nodes, denoted by set  $S_B = \{B_i : \xi_{B_i} \text{ is one of the top } k \text{ largest in the network and } 1 \leq i \leq k\}$ . The choice of  $k$  will be discussed in the next section. Selecting broker nodes based on detecting ability has two properties. First, assigning nodes with larger  $\xi$  to be broker nodes can reduce network throughput. Otherwise, the events detected by these nodes need to be transferred to other nodes first. Second, considering that broker nodes are used to collect subscriptions, they can delete

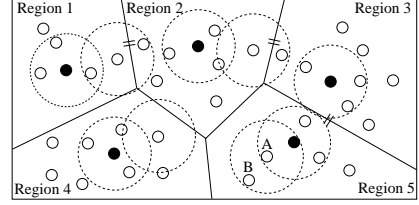


Fig. 1. An example to illustrate Voronoi regions of the network.

unnecessary events locally according to these subscriptions. Having nodes with larger  $\xi$  as brokers improves the possibility of deletion, which conserves memory and network bandwidth.

The implementation of Voronoi regions generation is a bootstrapping process. Initially, there are no broker nodes in the network, and all nodes broadcast their existing information including their detecting ability  $\xi$  and node ID. Then, the top  $k$  nodes with the largest  $\xi$  in the network will be marked as the broker nodes. To identify the Voronoi region it maintains, each broker node chooses a unique region ID. Notice that the broker node is called the local broker node for the non-broker nodes in its Voronoi region. After that, those broker nodes broadcast their node ID and region ID message. Non-broker nodes will find the nearest broker nodes to register based on the delay of the broadcasted messages they received. Nodes in the same Voronoi region share the same region ID, and non-broker nodes may be multiple hops away from their local brokers. An example of Voronoi regions is shown in Figure 1. The network is divided into 5 Voronoi regions. Non-broker node  $B$  registers to its local broker node through node  $A$ .

##### B. Mobility Adaption

Due to nodes' mobile nature, the Voronoi regions may change over time. Thus, a dynamic partition strategy for mobility adaption is necessary. In detail, because of the ad hoc environment, that strategy should satisfy the following requirements: a) it should be conducted in a distributed way; b) it assures that each non-broker node should always attach to a broker node at any time; c) it assures that the local broker node of a non-broker node should always be its nearest broker node in the network.

In DRIP, each broker node periodically broadcasts its existing information message at a low frequency, which consists of its node ID, region ID, sequence number, source time-stamp, and delay time-stamp. The source time-stamp records the time-stamp to broadcast the message and the delay time-stamp records when a non-broker node receives the broadcasted message. Then the delay  $Delay_j^{B_i}$  between a broker node  $B_i$  and a non-broker node  $j$  can be acquired by subtraction. Here, we assume that the clocks in all peers can be synchronized by current techniques to an acceptable accuracy. Every non-broker node always keeps the delay information of its top  $l$  ( $l < k$ ) nearest broker nodes, denoted by set  $S_D = \{D_1, D_2, \dots, D_l\}$ . When receiving a broadcasted message, the non-broker node would first check the delay value  $Delay_j^{B_i}$  of the incoming message. If  $Delay_j^{B_i} \geq Delay_j^{D_z}$  and  $1 \leq z \leq l$ , the non-broker node

would discard that message. Otherwise, the information of the broker node, which has the  $\max(Delay_j^{D^z})$  and  $1 \leq z \leq l$ , would be replaced by the originator of the incoming message, and that message would be forwarded to the non-broker node's neighbors. Every non-broker node always chooses the broker with the  $\min(Delay_j^{D^z})$  in the set  $S_D$  to attach to. Note that although we let broker nodes broadcast their existing information, it will not cause any broadcast-storm because eventually every broadcasted message would stop at some non-broker node. Also, since a broker node broadcasts messages periodically, non-broker nodes receiving these messages can communicate with the broker node by a lightweight AODV protocol.

## V. PUB/SUB PROCESS

### A. Subscription, Publication and Notification

The three operations in a pub/sub system are subscription, publication and notification. We describe how to implement these three operations as follows.

1) *Subscription*: In DRIP, a subscriber submits its subscriptions to its local broker node. Then, the local broker node spreads the subscriptions to other brokers by the broker nodes communication scheme. A subscription message consists of conditions of the requested events, subscriber's node ID, and several broker nodes' IDs. We will discuss the third item in Section V-B.

Since nodes are mobile and the Voronoi regions change over time, a mobility management issue occurs. That is, when a subscriber changes its local broker node, its subscriptions on the previous local broker node should be transferred to the new local broker node. Our solution is that every non-broker node makes a local copy of its subscriptions. When a non-broker node decides to change the local broker, it sends a *Quit* message to its previous local broker, which would delete the corresponding subscriptions. Then, the non-broker node registers to its new local broker with the copy of its subscriptions.

2) *Publication*: Both non-broker nodes and broker nodes can detect events. Having detected new events, nodes publish them to their local broker nodes. An event publication message is composed of the content of the event and the time when the event is published. To reduce the memory overhead, broker nodes delete events when there is not enough space. The deletion is conducted based on two conditions. One is subscription-related - that is - broker nodes will delete the events that are less related to the existing subscriptions. The other is publishing time; newly published events always have higher priority to stay.

3) *Notification*: Broker nodes collect the subscriptions and events from other nodes. Once a broker node receives a newly-published event, it sends the event to the interested nodes according to their subscriptions. If the interested nodes are the non-broker nodes in its Voronoi region, the broker node will flood the event only in its Voronoi region. Otherwise, it sends the event to other brokers in a multiple-copy-multiple-destination way, which is introduced in Section V-B2. Non-

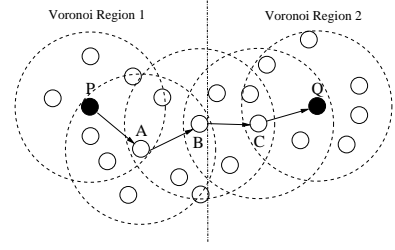


Fig. 2. An example to illustrate communication between two broker nodes. Broker nodes are painted black. The broken line circles stand for node's transmission range.

broker nodes get the matched events from their local broker nodes.

### B. Cooperation among broker nodes

Broker nodes need to cooperate with each other to share subscriptions and deliver useful events in DRIP. We first discuss the way that broker nodes communicate with each other. Then, we present the detailed cooperation strategy.

1) *Communication Method*: Since nodes only have 1-hop information, broker nodes cannot communicate with each other directly unless they are neighbors. Thus, we use a *bridge node* for broker node communication. A bridge node should satisfy three requirements: 1) it is a non-broker node; 2) for a pair of broker nodes, say  $M$  and  $N$ , they are the top two nearest broker nodes of the bridge node; 3) if  $M$  is the bridge node's local broker, the bridge node should have a neighbor whose local broker is  $N$ . A bridge node periodically updates its local broker on its status. As in Figure 2, the local broker of non-broker node  $B$  is  $P$ .  $Q$  is  $B$ 's second nearest broker node.  $B$ 's neighbor node  $C$  is in  $Q$ 's Voronoi region. Thus, node  $B$  is the bridge node for  $P$  and  $Q$ , denoted by  $B = Bri(P, Q)$ . Note that  $Bri(P, Q) \neq Bri(Q, P)$ , because they attach a different broker node. When the broker node  $P$  needs to contact the broker node  $Q$ , it first finds one bridge node  $B$  and contacts it. Then  $B$  relays the message to its 1-hop foreign neighbor  $C$ . As a result, the message can be delivered to the destination  $Q$  from  $C$ , as shown in Figure 2. Note that even though two Voronoi regions are close to each other, bridge nodes may not exist, which implies that the two corresponding broker nodes cannot communicate.

2) *Sharing Subscriptions and Events*: We utilize a pull-based approach to share events among broker nodes. That is, broker nodes only send the matched events to other interested broker nodes according to their own collected subscriptions. For a specified event, the interested broker nodes are the broker nodes whose Voronoi regions the subscribers may be in. On the contrary, broker nodes exchange their subscriptions when they can communicate. This is because each broker node is likely to have some matched events. To acquire those events, subscriptions need to be delivered to all of the broker nodes. Compared to events, subscription packets are smaller. Thus, our approach can reduce network throughput.

To send the matched events to the right interested broker nodes under the dynamic environment, we design a *multi-copy-multi-destination* scheme for sharing events among broker nodes. That is, the source broker node spreads the requested events to its  $m$  nearby broker nodes, denoted by  $\{M_1, M_2, \dots, M_m\}$ , through bridge nodes. Then, after some period, these  $m$  broker nodes would contact all of the interested broker nodes due to nodes' mobility, and transfer the events to them. In this way, subscribers can get the events from one of those interested broker nodes. We estimate the likelihood that a subscriber attaches to a broker node according to the delay. Specially, after a short period, a subscriber is likely in the Voronoi region of one of its nearest broker nodes. Recall that each non-broker node keeps the status information of its  $l$  nearest broker nodes. Note that  $l + m < k$ . Thus, if the events can be sent to those  $l$  broker nodes, the subscriber can get them with a large probability. For example, in Figure 1, the top two nearest broker nodes of node  $A$  are  $B_2, B_3$ . If a broker node has  $A$ 's interested events, it would deliver them to  $B_2$  and  $B_3$ . And  $A$  can get those events from one of the two brokers in a short period with a large probability. Note that the nearest broker nodes information would change every time  $A$  submits its subscriptions.

### C. Analytical Study

In this section, we analyze the delay of multi-copy-multi-destination scheme, denoted by  $T_{mcmd}$ . We show a logarithmic relation among the delay  $T_{mcmd}$ , the number of broker nodes  $k$ , the number of nearest broker nodes  $l$  for delivery destinations, and the number of copies  $m$  in the network can be found. This implies a rational choice of  $k$  when setting up Voronoi regions in the network. In our design, all nodes perform independent random walks on an  $\sqrt{N} * \sqrt{N}$  grid. Two broker nodes can communicate only when the bridge nodes exist, which implies their Voronoi regions meet. Also, we assume that a broker node can contact only one broker node at a time, and spreads copies of the event message to its nearby  $m$  broker nodes quickly. Let  $T_{D_i}$  denote the time until one of the  $m$  copies finds the destination broker node  $D_i$  ( $1 \leq i \leq l$ ).  $T_P^Q$  is the hitting time until the Voronoi regions of broker node  $P$  and  $Q$  meet, which is an independent identical exponential distribution with average  $1/\lambda$  [9]. Note that the time until copies find the destination broker node is I.I.D. Then, the probability of  $T_{mcmd}$  is  $P(T_{mcmd} < t) = P(T_{D_1} < t, T_{D_2} < t, \dots, T_{D_l} < t) = \prod_{i=1}^l P(T_{D_i} < t) = (P(T_{D_1} < t))^l$ . Because  $P(T_{D_1} < t)$  is the minimum of  $m$  independent identical exponentially distribution, which is  $1 - e^{-\lambda mt}$ . Thus,

$$P(T_{mcmd} < t) = (1 - e^{-\lambda mt})^l. \quad (1)$$

Approximating  $(1 - e^{-\lambda mt})^l$  with its Taylor Series terms up to second order, we can derive  $P(T_{mcmd} < t) \approx 1 - le^{-\lambda mt} + \frac{l(l-1)}{2}e^{-2\lambda mt}$ . With that, we can calculate the expected value of  $T_{mcmd}$ ,  $ET_{mcmd} = \frac{l^2+3l}{4m\lambda}$ . Suppose broker nodes are uniformly distributed in the network, two broker nodes can communicate through bridge nodes in a range

of  $2\sqrt{\frac{N}{\pi k}}$ . According to [9], we get that  $1/\lambda$  is equal to  $0.5N(0.34 \log N - 2 + \frac{\alpha}{2\alpha-1})$ , where  $\alpha$  is  $2\sqrt{\frac{N}{\pi k}}$ . Thus,

$$ET_{mcmd} = \frac{l^2 + 3l}{8m} N(0.34 \log N - 2 + \frac{\alpha}{2\alpha - 1}), \quad (2)$$

with  $\alpha$  being  $2\sqrt{\frac{N}{\pi k}}$ . Because the grid size parameter  $N$  is fixed,  $ET_{mcmd}$  is determined by the number of broker nodes  $k$ , the number of nearest broker nodes  $l$  for delivery destination, and the number of copies  $m$ . When the copies of event messages increases, the number of broker nodes decreases, and the number of destination broker nodes declines,  $ET_{mcmd}$  is decreasing. Note that the reasonably larger  $l$  is, the more likely it will be for the subscriber to receive the event messages. Moreover, in order to spread  $m$  quickly,  $m$  should be kept at a rationally small value and  $m + l < k$ . Once  $l$  and  $m$  are determined, given a requested  $ET_{mcmd}$ , we can derive a proper value of the number of broker nodes  $k$  in the network with Eq.(2).

## VI. SIMULATIONS

### A. Simulation Overview

We implemented DRIP and tested it under a variety of simulated scenarios and loads with the JiST/SWANS v1.0.6 [1]. For performance reference, we also developed a simple framework, called SF, in which nodes are divided into Voronoi regions, but events are delivered among broker nodes in an epidemic way. To evaluate the quality of DRIP, we are mainly interested in two properties: *broker coverage* and *delivery delay*. Broker coverage measures the number of broker nodes that have the copy of a specified event when its subscriber gets it. Delivery delay indicates the delay when a specified event is delivered to its subscribers.

1) *Environment Parameters*: We simulate the mobile ad hoc networks with 200 nodes, which are uniformly distributed in a 1,000m  $\times$  1,000m field. The movement of nodes follows the random waypoint model. The nodes' moving speeds are uniformly chosen from 10 to 20 m/s and the pause time is uniformly distributed between 2 and 4 seconds. The maximal transmission range of a node is set to 120m radius. The IEEE 802.11 protocol is used as the MAC layer protocol.

2) *Subscription and Publications*: Subscriptions and publications are simulated by the Number Interval (NI) model in [7]. In specific, an event publication is represented by a integer value within the real interval [1, 100]. A subscription is represented by a range within [1, 100]. An event matches a subscription only when the event's value falls into the subscription's range. Events are generated by a Poisson process with intensity 0.8. Moreover, we determine whether a node in the network detects the event based on its detecting abilities. Nodes' detecting abilities in the network follow a power-law distribution. To make it practical, the probability that a node has subscriptions is 50%. The generation of subscriptions is conducted by picking a center and a range. The center value is randomly chosen between 1 and 100, while the length of

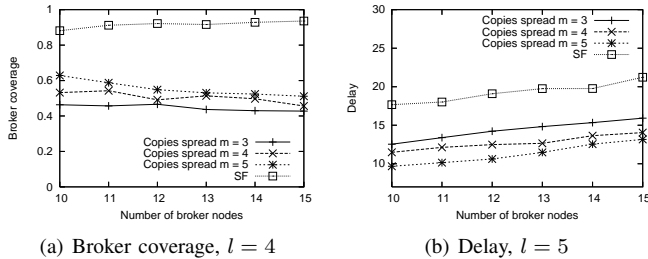


Fig. 3. Impact of  $m$  in 200 nodes network. The  $x$ -axis shows the number of broker nodes in the network.

the range is selected randomly based on a normal distribution with mean 5 and variance 2.

## B. Results

In this section, we present the results of our simulations. Events are generated as time goes by. Those events are delivered to the matched subscribers. We run each simulation for 500 time units, and report the average for all measurements.

1) *Broker Coverage*: Broker coverage is given as the ratio of the number of the broker nodes getting the delivered events to all the broker nodes in the network, which demonstrates two properties. One is that when the broker coverage is low, network memory utilization improves, which means less copies of events in the network. The other is the network throughput, because low broker coverage indicates less transmission. Therefore, the less broker coverage it has, the better the performance is. We first fix the number of interested brokers  $l$ , vary the number of brokers  $k$  and the number of copies  $m$  in the network, and measure the corresponding broker coverage. Figure 3(a) plots the broker coverage in the three networks with  $l = 4$ . We see as  $m$  increases, broker coverage is growing too. This indicates that although we can raise the possibility of delivering matching events to their subscribers by enlarging  $m$ , more network transmission and memory is required because more broker nodes in the network need to store the copy of the same event. Moreover, if we have  $m$  and  $l$  fixed, and let  $k$  grow, the broker coverage is generally reduced. This is because, when sending an event to its subscriber, even in the worst case, only  $m + l$  brokers are required to have a copy of that event instead of all the brokers. Thus, when  $k$  is raised, broker coverage is decreased.

2) *Delivery Delay*: The delivery delay is measured by the simulation time in JiST/SWANS. Figure 3(b) records the delivery delay with the different number of copies  $m$ .  $l$  is set to 5. We see that our scheme costs less time than the SF scheme. The reason is that SF needs to deliver every single event to all of the broker nodes, while in our scheme only a subset of the broker nodes are required to have a copy of the delivered events. As expected, given a fixed number of broker nodes  $k$  in the network, delivery delay decreases as  $m$  is increasing. This is because more copies can enlarge the opportunity to deliver the matched events to their subscribers. In addition, when  $m$  is fixed, delivery delay goes up with  $k$ . The intuitive reason for this is that, when  $k$  increases, there are more Voronoi regions

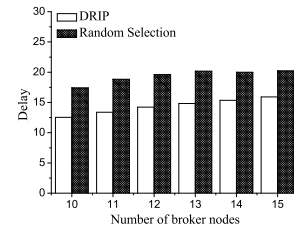


Fig. 4. Comparison of delivery delay with different broker node selection scheme. The  $x$ -axis shows the number of broker nodes in the network.

in the network, resulting in an increase in the time it takes to locate the region where the subscriber is.

Then, we evaluate the impact of broker nodes selection scheme on the delivery delay. DRIP assigns nodes with the largest detecting ability as broker nodes. For comparison, we also develop a random selection scheme, which randomly picks  $k$  nodes as the broker nodes in the beginning. In Figure 4, we see that DRIP incurs less delivery delay in the network. This is because by marking nodes with the largest detecting ability as broker nodes, DRIP saves a step that most of the detected events need to be transferred to the broker nodes first.

## VII. CONCLUSION

In this paper, we propose DRIP, a dynamic Voronoi region-based pub/sub scheme, which adapts the pub/sub paradigm to mobile ad hoc networks. With our design, the network is dynamically divided into several Voronoi regions by selecting proper nodes as broker nodes. The contribution of this paper lies in determining how to choose broker nodes, how many broker nodes we need, and how to implement a pub/sub process based on the dynamic Voronoi regions. Also, comprehensive simulations prove the performance of our design.

## REFERENCES

- [1] JiST/SWANS. <http://jist.ece.cs.cornell.edu>.
- [2] I. Burcea, H. Jacobsen, E. Lara, V. Muthusamy, and M. Petrovic. Disconnected operation in publish/subscribe middleware. In *Proceedings of MDM*, 2004.
- [3] P. Eugster, P. Felber, R. Guerraoui, and A.-M. Kermarrec. The many faces of publish/subscribe. In *Technical Report DSC ID:2000104, EPFL*, 2001.
- [4] U. Farooq, E. W. Parsons, and S. Majumdar. Performance of publish/subscribe middleware in mobile wireless networks. In *Proceedings of WOSP*, 2004.
- [5] L. Fiege, Felix C. Gartner, O. Kasten, and A. Zeidler. Supporting mobility in content-based publish/subscribe middleware. In *Proceedings of Middleware*, 2003.
- [6] Y. Huang and H. Garcia-Molina. Publish/subscribe in a mobile environment. In *Proceedings of the Second ACM International Workshop on Data Engineering for Wireless and Mobile Access*, 2001.
- [7] Y. Huang and H. Garcia-Molina. Publish/subscribe tree construction in wireless ad-hoc networks. In *Proceedings of MDM*, 2003.
- [8] L. Podnar and I. Lovrek. Supporting mobility with persistent notifications in publish/subscribe systems. In *Proceedings of DEBS*, 2004.
- [9] T. Spyropoulos, K. Psounis, and C. S. Raghavendra. Spray and wait: an efficient routing scheme for intermittently connected mobile networks. In *Proceedings of ACM SIGCOMM*, 2005.
- [10] J. Wang, J. Cao, and J. Li. Supporting mobile clients in publish/subscribe systems. In *Proceedings of ICDCSW*, 2005.