NAIST-IS-DD0861207

Doctoral Dissertation

Methods for Efficient Service Replication and Information Gathering in Mobile Networks

Asaad Ahmed Gad El-Rab Ahmed

March , 2012

Department of Information Processing
Graduate School of Information Science
Nara Institute of Science and Technology

A Doctoral Dissertation
submitted to Graduate School of Information Science,
Nara Institute of Science and Technology
in partial fulfillment of the requirements for the degree of
Doctor of ENGINEERING

Asaad Ahmed Gad El-Rab Ahmed

Thesis Committee:

Professor Minoru Ito	(Supervisor)
Professor Hiroyuki Seki	(Co-supervisor)
Professor Keiichi Yasumoto	(Co-supervisor)

Methods for Efficient Service Replication and Information Gathering in Mobile Networks[*]

Asaad Ahmed Gad El-Rab Ahmed

Abstract

Advances of wireless technologies and increasing use of mobile devices have brought the best solution to allow mobile users to communicate with other users. Mobile networks can be classified in two major categories: infrastructure networks like cellular networks and infrastructureless networks like mobile ad-hoc networks (MANETs). While cellular networks are characterized by having fixed and wired gateways, which are responsible for routing messages, MANETs offer quick and easy network deployment in situations where it is required to provide mobile users with a temporary infrastructure. One of the challenging tasks in such environments is how to improve the information sharing service availability between users in MANETs. For example, users want to share situations of the city by temperature, humidity, and noise on any geographical area. Replicating a service at some nodes distributed across the network is an effective strategy. However, service replication can considerably impact the system energy consumption. Since mobile devices have limited battery resources, a dynamic and efficient service replication is necessary to support such environments. Information gathering in a specified geographical area is another challenging task. Exploiting people as a part of the information gathering infrastructure, introduces a new paradigm called *People-Centric Sensing* (PCS) or *participatory sensing*. In PCS, the sensing coverage depends on the uncontrollable mobility of people, therefore it is difficult to achieve full coverage of the target *area of interest* (AoI). Consequently, we propose a concept of (α, T)-coverage of AoI where each point in AoI is sensed by at least one mobile node with the probability of at least α during time period T. Our goal is to achieve (α, T)-coverage of a given AoI by a minimal set of mobile nodes.

[*] Doctoral Dissertation, Department of Information Processing, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-DD0861207, March , 2012.

In the first part of this thesis, we propose a distributed replication scheme called Highly Distributed Adaptive Service Replication (HDAR), aiming to improve service availability in MANETs with reasonable energy consumption across the network. In order to dynamically place service replicas in appropriate nodes, HDAR divides the whole network into disjoint zones with diameters at most 2 hops, selects a node with minimum moving speed in each zone as a zone head, and constructs a virtual backbone network connecting all zone heads. HDAR replicates a service dynamically for some of the zones depending on the service demand level in each zone and the tradeoff between the communication and replication energy consumption costs. In addition, to control the number of service replicas, HADR lets neighboring servers exchange with each other information on their covering zones. Through simulations, we confirmed that our approach can achieve up to 5% higher service availability than existing methods with reasonable energy consumption which was reduced up to 53%.

In the second part of this thesis, to solve the (α, T)-coverage problem, we propose two algorithms: *inter-location* and *inter-meeting-time* algorithms, to meet a coverage ratio α in time period T. The proposed algorithms estimate the probability of locations in the AoI being visited by each mobile sensor node in T, and selects a minimal number of nodes inside the AoI. The inter-location algorithm considers the distance between the nodes while the inter-meeting-time algorithm regards the expected time until any two of the nodes will meet at a location. To meet the required coverage in the case of an insufficient number of nodes existing inside the AoI, we propose an extended algorithm which takes into account not only nodes existing inside the AoI, but also nodes outside the AoI. In addition, for more accurate coverage, we propose an updating mechanism which aims to remove useless nodes and add some extra nodes that contribute more to AoI coverage. The conducted simulations show that the proposed algorithms achieve (α, T)-coverage and the number of selected nodes is much smaller than the number of candidates nodes in the AoI. The reduction ratio of the number of selected nodes is up to 76%, 79%, 85%, and 80% for a variety of values of α, T, node density and AoI size, respectively, compared with the number of all nodes in AoI.

Keywords:

MANET, Service replication, Service availability, Energy consumption, People-centric sensing, Probabilistic coverage, Markov chain.

Contents

List of Figures

List of Tables

1. Introduction

With the advances of wireless technologies and the increasing use of wireless networks, satellites, and portable devices like laptops, PDA, and mobile phones, a trend to support computing while moving has emerged. This trend is known as *mobile computing* or *nomadic computing* [1]. It is also referred to as anytime and anywhere computing. Since a user may not maintain a fixed position in such environments, the mobile and wireless networking support allowing mobile users to communicate with other users (fixed or mobile) becomes crucial. Mobile networks are being used in diverse areas such as travel, education, stock trading, military, package delivery, disaster recovery, business and e-commerce, telecommunications and personal communications, real-time control systems, remote operation of appliances, medical emergency care, and in accessing the Internet applications. Mobile networks are quickly becoming the networks of choice, due to not only their large bandwidth, but the flexibility and freedom they offer [2]. Mobile networks are also experiencing significant progress in the form of wireless local area networks (WLANs) [3], satellite-based networks [4], Wireless Local Loops (WLL) [5], mobile IP [6], and wireless Asynchronous Transfer Mode (ATM) networks [7, 8]. As expected in [9], mobile and wireless technology has made wireless devices remarkably convenient and affordable. Wireless networking is specifically appropriate for situations wherein installation of physical media is not feasible and which require on-the-spot access to information. Wireless networking makes it possible to have access to both voice and data [10]. Also, the movement of mobile wireless technologies in higher education has become an interested research area [11–14]. With the advantages of mobility, mobile wireless technologies help improve efficiency and effectiveness in teaching and learning [15].

Mobile networks can be classified in two major categories: infrastructure networks including cellular networks (Fig. 1) and infrastructureless networks including mobile ad-hoc networks (MANETs)(Fig. 2). While cellular networks are characterized by having fixed and wired gateways (base stations), which are responsible for routing messages, MANETs offer quick and easy network deployment in situations where it is not possible otherwise and they can be used to provide mobile users with a temporary infrastructure to use services in the absence of fixed infrastructure. Nodes in MANETs are free to move and organize themselves in an arbitrary fashion.

One of the challenging tasks in such environments is how to improve the informa-

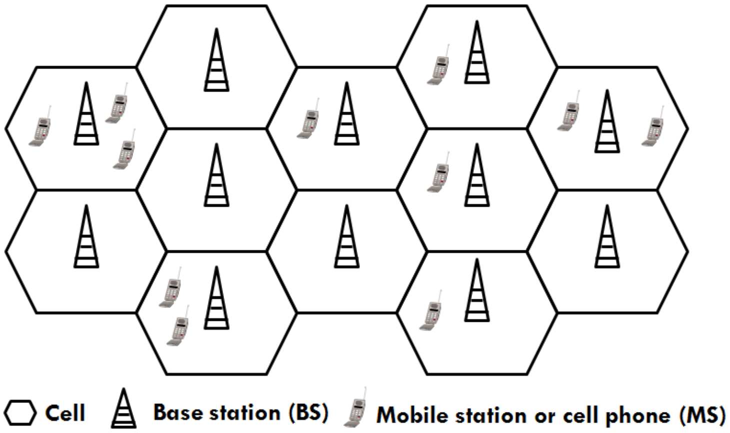


Fig. 1: Cellular Networks

tion sharing service availability between users in MANETs. For example, users want to share situations of the city by temperature, humidity, and noise on any geographical area. Replicating a service at some nodes distributed across the network is an effective strategy. However, service replication can considerably impact the system energy consumption. Since mobile devices have limited battery resources, a dynamic and efficient service replication is necessary to support such environments.

Meanwhile, recently the demand for realtime environmental information about specific regions in urban areas has been increasing for various purposes such as surveillance, navigation, and event detection. People moving inside an urban area offer the possibility of covering a given *area of interest* (AoI) at low cost. This information gathering in a specified geographical area is another challenging task. Exploiting people as a part of the information gathering infrastructure, introduces a new paradigm called *People-Centric Sensing* (PCS) or *participatory sensing*. In PCS, the sensing coverage depends on the uncontrollable mobility of people, therefore it is difficult to achieve full coverage of the target *area of interest* (AoI). Consequently, we propose a concept of (α, T)-coverage of AoI where each point in AoI is sensed by at least one mobile node with the probability of at least α during time period T. Our goal is to achieve (α, T)-coverage of a given AoI by a minimal set of mobile nodes.

In the first part of this thesis, we propose two heuristic distributed replication schemes aiming to improve service availability in MANETs with reasonable energy

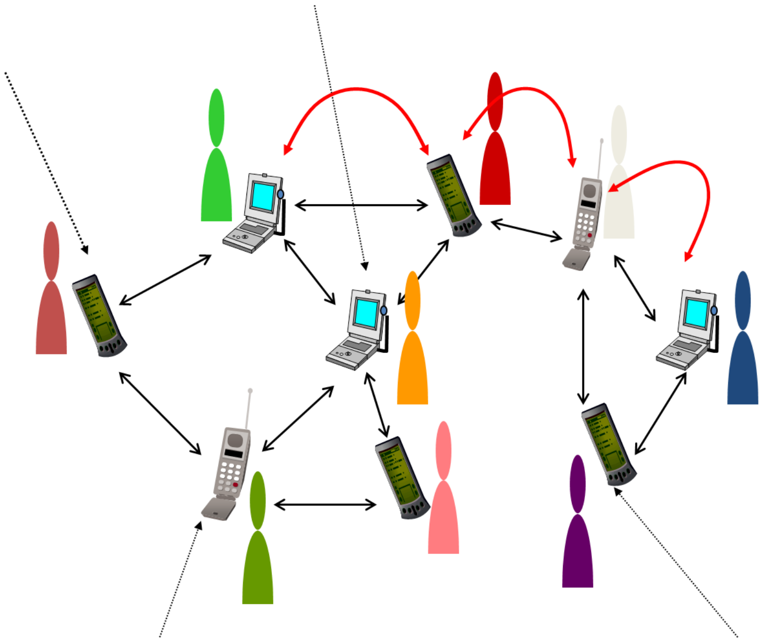


Fig. 2: Mobile ad-hoc networks, MANETs

consumption across the network, which dynamically place service replicas in appropriate nodes depending on the service demand level in a zone (i.e., a group of nodes) and the tradeoff between the communication and replication energy consumption costs. In the second part of this thesis, to solve the (α, T)-coverage problem, we propose three algorithms to meet a coverage ratio α in time period T. The proposed algorithms estimate the probability of locations in the AoI being visited by each mobile sensor node in T, and selects a minimal number of nodes inside the AoI.

First, in Chapter 2, we propose two heuristic algorithms but fully distributed service replication algorithm called DAR and HDAR to solve the service availability problem. In order to dynamically place service replicas in appropriate nodes, DAR and HDAR divide the whole network into disjoint zones with diameters at most 2 hops, select a node with minimum moving speed in each zone as a zone head, and construct a virtual backbone network connecting all zone heads. DAR replicates a service dynamically to some of the zones depending on the service demand level in each zone. In DAR, each server executes the replication mechanism independently of other servers and does not take into account the balance among the covering zones of servers. In addition, the path length between a client and a server depends on the network size where the path length increases as network size increases. In HDAR, to eliminate this dependability and make the balance among zones, in addition to the service demand level in each zone, the replication mechanism considers the tradeoff between the communication and replication energy consumption costs and lets neighboring servers exchange with each other information on their covering zones. In order to evaluate the effectiveness of DAR and HDAR, we conducted simulations for a variety of values of network size, node velocity, service demand level, and replication interval. The conducted simulations show that our approaches can achieve up to 5% higher service availability than existing methods with reasonable energy consumption which was reduced up to 53%.

Second, in Chapter 3, we propose three probabilistic coverage algorithms to solve the (α, T)-coverage problem, named *Inter-Location Based (ILB)*, *Inter-Meeting Time Based (IMTB)*, and *Extended Algorithm without Thresholds (EWOT)* based on the initial locations of nodes existing inside and near AoI when a query is initiated. For more accurate coverage, we propose an update mechanism for the ILB and IMTB algorithms that aims to adapt the number of selected nodes based on the latest location of nodes. This update mechanism is executed every specified time interval during the time period T. We assume that all algorithms are executed by the server s in a centralized fashion. The conducted simulations show that the proposed algorithms achieve (α, T)-coverage and the number of selected nodes is much smaller than the number of candidates nodes in the AoI. The reduction ratio of the number of selected nodes is up to 76%, 79%, 85%, and 80% for a variety of values of α, T, node density and AoI size, respectively, compared with the number of all nodes in AoI.

2. Methods for Efficient Service Replication in MANETs

2.1 Introduction

MANET applications can offer various services and resources to users such as multimedia information service, file-sharing service, database retrieval service, location service, etc. In such applications, users need to detect, share, and invoke the services and resources in a flexible manner. To build a model for those applications, we need a way to organize and maintain service objects based on *Service Oriented Architecture* (SOA) [16].

Consider the following application scenario (as shown in Fig. 3): pedestrians with mobile terminals in a city have some sensors. They want to share situations of the city by temperature, humidity, human density, noise, illuminance, etc. on any geographical point in the target area. This can be realized with SOA as follows: all mobile nodes sense data and send it to a service provider node. The service provider node constructs a database with the collected data and provides a retrieval service. A mobile node sends a query to the service provider node with the location where the node wants to know the situation. In this scenario, since WLAN does not cover the whole city and cellular networks may be overloaded with high density areas or results in high cost (e.g. people may not want to pay for data upload), MANETs can be used as alternative networks to realize this SOA.

The basic components to realize SOA are services (objects and providers), clients, and a service discovery mechanism. Due to dynamic nature of MANETs, a service provider may be either temporary or even permanently unavailable for the client nodes, because the provider node leaves the network and/or battery of the provider node is depleted due to communication and computation loads. From the client's point of view, a service must be available regardless of these reasons. Also, mobile devices have a limited amount of battery, so the energy consumption for retrieving service across the network has to be minimized. Service replication is an effective strategy to satisfy these goals.

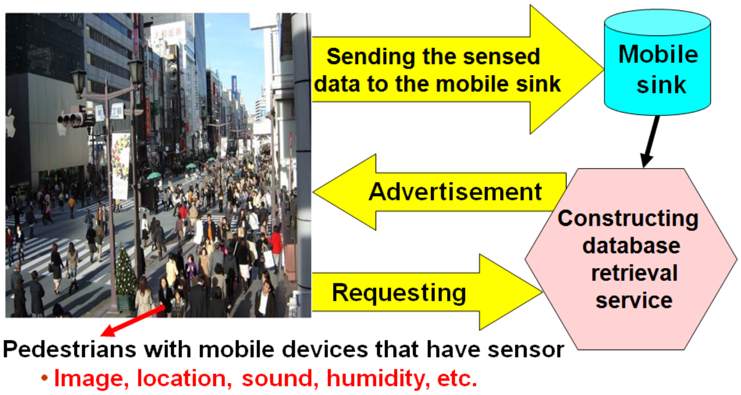

Sending the sensed data to the mobile sink
Mobile sink
Advertisement
Constructing database retrieval service
Requesting
Pedestrians with mobile devices that have sensor
• Image, location, sound, humidity, etc.

Fig. 3: Application scenario for service availability in MANETs

2.2 Related work

There have been (proposed) several methods that replicate a service to some new host nodes based on particular strategies such as a whole knowledge of the network [17,18], network partitioning [19–23], network density [24], or client's request rate [25]. In [22, 23], Derhab et al. proposed a method utilizing replication and merging mechanisms based on estimation of the link quality and partition prediction by using the TORA [26] and some partition detection mechanisms. REDMAN middleware [24] aimed to support resource replication in dense MANETs. By using a simple gossip-based strategy, each node randomly decides whether to host a replica on its storage and successively forwards the resources to be replicated to one of its neighbors.

In [17, 18], the replication process depends on the whole knowledge of the network where nodes should require the information about the other nodes in the network. By using link quality to predict network partitioning, the original service is replicated to the node with high battery lifetime in the partition. In [19], each client monitors the set of disjoint paths between itself and the server and computes a certain metric. If this metric falls below a certain threshold then a potential partition is identified and server replication is initiated. In [20, 21], a partition prediction model was proposed based on grouping of nodes according to their position and speed. Every client sends its coordinates and velocity to the server. Having this global knowledge, the server can predict future partitions and a new server is replicated accordingly. In [25], service distribution protocol *(SDP)* was proposed based on clients' and providers' interests.

In this algorithm, there are two mechanisms: (1) service replication where the service is replicated to a new node if its interest exceeds some predefined threshold called *replication threshold*; and (2) service hibernation where the service is hibernated from the service provider if its interest is less than or equal to some predefined threshold called *hibernation threshold*.

Most of the aforementioned approaches focus mainly on service availability improvement in the case that network partitioning is predicted. When a partition is going to happen, the requested service is replicated in advance and connectivity to it can be guaranteed. The other approaches use another strategy such as gossip-based strategy [24] or fixed threshold such as client's interest [25]. However, all of them focus on how to increase the service availability without taking into account the energy consumption. Thus, we need to design a dynamic replication strategy to select a limited number of nodes to act as service providers that balance energy consumption and the service availability without full knowledge of the network. In addition, the path length between a client and a server is another important metric, because if the path between the client and the server is too long, the client may not access the service and the energy consumption will increase. So, we need to minimize the path length between the client and the server. Here, we use the same definition of service availability as in [22, 25], that is, the ratio of the number of service replies received to the number of service requests sent.

In this chapter, we propose two heuristics algorithms but fully distributed service replication algorithm called DAR and HDAR to solve the service availability problem in MANETs. DAR and HDAR are based on a zone structure where the whole network is divided into disjoint zones. The zone-structure is often used to solve the scalability issue in large MANETs [29–31]. The main two phases in zone-based structure protocols are zone formation and zone maintenance. Zone formation refers to how to build a zone structure for a MANET at the very beginning. Zone maintenance defines how to update the zone structure according to the underlying network topology change during the operation.

A lot of zone-based approaches require the elected zone heads with specific attributes in the zone formation phase. For example, in the lowest ID approach (LID) [30], a mobile node can act as a zone head only if it has the lowest ID among all neighbors. In the highest connectivity approach (HCC) [31], a mobile node can act as a zone head only if it finds out that it has higher node degree than all neighbors. Instead, our proposed algorithms form zones and selects zone heads according to a mobility metric because the zone-based structure requires stable zones even in the presence of node mobility.

In the rest of this chapter, we first describe assumptions, models, and definitions in Section 2.3, then we introduce our proposed algorithms in Sections 2.4 and 2.5. In Section 2.6, we introduce the performance evaluation to evaluate the effectiveness of DAR and HDAR by comparing its performance with the existing method SDP [25].

2.3 Assumptions, Models, and Definitions on Target MANETs

In this section, we describe assumptions, models, and definitions which are needed to formulate the problem to improve service availability when realizing SOA in MANETs.

2.3.1 Network Model

Hereafter, we use discrete time and represent the current time by positive integer variable t. A network at time t which consists of a set of servers and a set of clients is modeled as an undirected graph $G(t) = (V(t), E(t))$, where $V(t)$ is the set of nodes at time t and $E(t) \subseteq V(t) \times V(t)$ is the set of links among the nodes in $V(t)$ at time t. We assume that all nodes are cooperative and there is no any selfish node in $V(t)$. We denote the set of servers and the set of clients at time t as $S(t)$ and $C(t)$, respectively. Here, $V(t) = S(t) \cup C(t)$ and all nodes in $V(t)$ communicate through omni-directional antennas with some nodes in their transmission range denoted by T_R. T_R is assumed to be a disk with a certain radius centered at the sender node. We assume that each node $v \in V(t)$ has a unique ID and a *Received Signal Strength Indicator* (RSSI) capability, and knows its moving speed with some means. A link $(u, v) \in E(t)$ exists iff u and v are within the transmission range of each other at time t. We assume that $G(t)$ is connected for any t. This means that for any pair of nodes u and v, when u and v are not in the transmission range of each other, they can communicate with each other in a *multi-hop* manner through other nodes in the network. We denote the number of hops in the shortest path between any two nodes $u, v \in V(t)$ by $d(u, v, t)$. We assume that the shortest path between any two nodes is known through the lower layer routing protocol.

2.3.2 Service Model

We assume that each client knows the set of available servers and it can contact with a server by using closest (shortest hop) server selection scheme such as [27]. Each client sends a service request to the closest server with its ID and service parameters which depend on the type of application. In order to maintain loose data consistency among servers, we assume that each server periodically sends an update request including new data to other servers every long time period (e.g. every 1 hour). So, the cost of updating is not considered here. We assume that only one packet is needed to send the service request or the service reply message between the client and the server. We denote the service request path length from the client c to the server s at time t and the service reply path length from the server s to the client c at time t as $d(c, s, t)$ and $d(s, c, t)$, respectively. We assume that the energy amounts required to transmit and receive one packet along any link $(u, v) \in E(t)$ are fixed and denoted by E_{tr} and E_{rx}, respectively. We assume that the server s executes the replication process to select new service replicas every specified time interval called *replication interval* denoted by RI. We denote the number of service requests received from the client $c \in C(t)$ at server s per unit time at time t by $sq(c, s, t)$. We assume that s knows the value of $sq(c, s, t)$ locally by counting the number of service request messages received from c and the value of $sq(c, s, t)$ is initialized at the beginning of every replication interval RI.

2.3.3 Replication cost

Service availability is improved by replicating a service to a set of nodes across the network. However, replication itself imposes additional energy consumption for the nodes that transmit and receive the service object (i.e., program and data). Let $R_s(t)$ denote the set of new replica nodes which was determined by $s \in S(t)$ at time t. We assume that k packets are used to send the service object. The amount of consumed energy for the set of new replica nodes $R_s(t)$ at time t is denoted by $RepCost(s, R_s(t), t)$ and defined as:

$$RepCost(s, R_s(t), t) = k \times (E_{tr} + E_{rx}) \times \sum_{r \in R_s(t)} d(s, r, t) \qquad (1)$$

2.3.4 Communication cost

The service request or the service reply from/to the client c to/from the server s is transmitted through a one-hop or multi-hop path. During transmission, each node in the path consumes energy for receiving and transmitting a message. We model a communication cost for the client $c \in C(t)$ by the energy consumption to communicate with the server s along the service request path $d(c, s, t)$ and the service reply path $d(s, c, t)$. The communication cost to communicate with the server s during RI from time t is denoted by $ComCost(s, C(t), t)$ and defined as:

$$ComCost(s, C(t), t) = (E_{tr} + E_{rx}) \times RI \times \sum_{c \in C(t)} sq(c, s, t)[d(c, s, t) + d(s, c, t)]z(c, s) \tag{2}$$

where $z(c, s) = 1$ if the client c accesses the server s, otherwise $z(c, s) = 0$.

2.3.5 Definition of Service Availability

Given a network $G(t)$, and the number of service requests $sq(c, s, t)$ from each client $c \in C(t)$ to each server $s \in S(t)$ at time t, the total amount of energy consumption for communication and replication $Cost(S(t), C(t), t)$ at time t is defined as:

$$Cost(S(t), C(t), t) = \sum_{s \in S(t)} [\sum_{s' \in R_s(t)} ComCost(s', C(t), t) + RepCost(s, R_s(t), t)] \tag{3}$$

Our objective is to find a set of replica nodes $R_s(t)$ for each server $s \in S(t)$ which guarantees a certain level of service availability and minimizes the total amount of energy consumption for communication and replication $Cost(S(t), C(t), t)$. So, the objective function is defined as follows.

$$\textbf{Minimize} \;\; Cost(S(t), C(t), t) \tag{4}$$

subject to

$$\sum_{s \in S(t)} ComCost(s, C(t), t) - \sum_{s \in S(t)} \sum_{s' \in R_s(t)} ComCost(s', C(t), t) > RepCost(s, R_s(t), t) \quad (5)$$

Constraint (5) indicates the difference between the communication cost of the current set of servers and that of the new set of servers (after replication) is larger than the replication cost.

This problem is the combinatory optimization problem and is closely related to Uncapacitated Facility Location Problem (UFLP) which is known to be NP-hard [28].

2.4 DAR: Distributed Adaptive Service Replication Algorithm

2.4.1 Basic Idea

A major goal of constructing zones in DAR is to organize all zone heads into a virtual backbone network to simplify the system control, decrease message overhead, and manage the service replication mechanism. DAR forms zones and selects zone heads according to a mobility metric because the zone-based structure requires stable zones even in the presence of node mobility. Thus, DAR divides the whole network into disjoint zones, selects a node with minimum moving speed in each zone as a zone head, and constructs a virtual backbone network connecting all zone heads. By using our zone structure, DAR provides dynamic replication that determines the number and location of new service replicas. DAR aims to replicate a service dynamically to some of zones depending on the service demand level in each zone.

Our goals are as follows: (a) minimizing energy consumption depending on server workload (e.g. communication and computation loads) and communication cost between clients and the server and (b) improving the service availability. Hereafter, a zone is called an *active zone* if there exists at least one node in the zone which requests a service during a specified time interval, and a zone is called a *passive zone* if there exists no node in the zone which requests a service during a specified time interval.

2.4.2 Zone Formation and Maintenance

In our zone structure, there are the following four states for each node.

Zone_Undecided: *A node which does not belong to any zone.*

Zone_Head: *A node which has become a zone head.*

Zone_Member: *A node which belongs to a zone.*

Zone_MemberGeteway: *A node which is a zone member and a neighbor to at least one member of another zone.*

Now, we will describe the formation and maintenance processes in the following algorithm.

First, all nodes periodically send and receive "Hello" message to/from their neighbors (in 1-hop). Each hello message contains: node id, node speed, node residual energy, node capacity, node state, and other needed parameters. Secondly, all nodes start in $Zone\_Undecided$ state. Finally, each node changes its state as follows: (i) if a node has the lowest speed amongst all its neighbors, its state becomes $Zone\_Head$, otherwise its state becomes $Zone\_Member$, (ii) if two neighboring nodes in a $Zone\_Undecided$ state have the same lowest speed, the node with the lowest ID will be selected as zone head and the node's state becomes a $Zone\_Head$, and the other node's state is changed to $Zone\_Member$, (iii) if a $Zone\_Member$ node is a neighbor to at least one member of another zone, the node's state becomes $Zone\_MemberGateway$, (iv) if two nodes with state $Zone\_Head$ move into each other's radio range, then the node with the lowest speed remains in the state $Zone\_Head$ and the other node's state is changed to $Zone\_Member$, and (v) if a $Zone\_Undecided$ node becomes a neighbor of more than one node with state $Zone\_Head$, it uses the RSSI of hello messages received from these $Zone\_Head$ nodes; selects the stronger RSSI $Zone\_Head$ node and becomes a $Zone\_Member$ node in its zone.

This algorithm leads to the formation of disjoint zones which are at most 2-hops in diameter. Fig. 4 shows the result of applying the above algorithm, where each node is represented by (*ID*, *speed*) pair.

2.4.3 The DAR Replication Mechanism

In DAR, if each client asks its zone head zh to forward the service request message (after that), zh adds to the message the number of nodes in its zone ($n_{zone_{zh}}$) and forwards the message. In addition, each server s can serve multiple zones, so it maintains an $Aggregated\ Request\ Table$ (ART) which stores the zone identifier $zone_{zh}$, the number of received service requests $sr(s, zone_{zh})$, and the number of nodes $n_{zone_{zh}}$

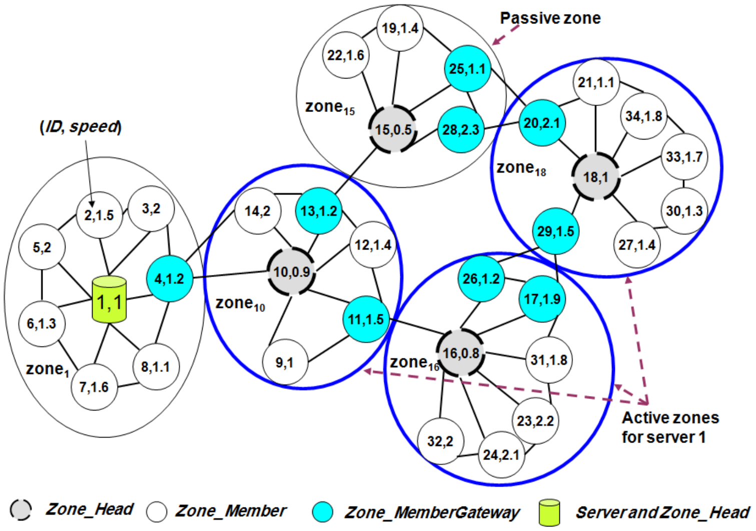


Fig. 4: Zone formation, sever, and active zones.

for every active zone $zone_{zh}$ which accesses s, as shown in Table 1. Due to mobility, multiple servers may exist in the same zone, so the zone head zh uses the registration information to build and update a set of servers $SS_{zone_{zh}}$ which exist inside its zone $zone_{zh}$. As described in the service model in Section 2.3.2, the server s can know received service demand information from the received service request messages. In DAR, replication mechanism depends on two factors: (1) the number of received service requests from all clients in each active zone $sr(s, zone_{zh})$ and (2) the number of nodes in this active zone $n_{zone_{zh}}$. These two factors determine the *degree of interest* of every active zone $zone_{zh}$ which is denoted by $DoI(s, zone_{zh})$. Based on $DoI(s, zone_{zh})$, server s selects new replica zones. Now, we will show how DAR determines the value of $DoI(s, zone_{zh})$.

Table 1: Aggregated Request Table, ART in DAR.

zone Id $zone_{zh}$	**number of received service requests** $sr(s, zone_{zh})$	**number of nodes** $n_{zone_{zh}}$
$zone_{10}$	6	6
$zone_{16}$	5	7
$zone_{18}$	10	8

2.4.3.1 Service Demand Level of Active Zone First, server s determines the average number of nodes for all active zones which is denoted by an_s and defined as:

$$an_s = \frac{\sum\limits_{zone_{zh} \in ART} n_{zone_{zh}}}{nz_s} \tag{6}$$

where nz_s is the total number of active zones (number of records in ART) that access s.

Secondly, we define the average number of received service requests, $avsr(s, zone_{zh})$, for each active $zone_{zh}$ which accesses s as follows:

$$avsr(s, zone_{zh}) = \frac{sr(s, zone_{zh})}{an_s} \tag{7}$$

Finally, by using (7), $DoI(s, zone_{zh})$ is defined as:

$$DoI(s, zone_{zh}) = \begin{cases} \text{high} & \text{if } avsr(s, zone_{zh}) \geq 1, \\ \text{low} & \text{if } avsr(s, zone_{zh}) < 1 \end{cases} \tag{8}$$

2.4.3.2 DAR Replication Rule Based on $DoI(s, zone_{zh})$, we propose a DAR replication mechanism. In this mechanism, every server s selects multiple active zones without replicas to host new replicas once the replication process is triggered. These active zones with replicas are determined based on the following rule:

"*Replicate a service hosted at a server s to an active zone $zone_{zh}$ if $DoI(s, zone_{zh})$ is high.*"

Algorithm 3 shows DAR mechanism at servers. Every active server s periodically executes this algorithm based on its ART contents to select new replica zones. In line 1 to 6, the algorithm calculates the number of active zones nz_s and the total number of nodes in these active zones. In line 7, the algorithm calculates the average number of nodes, an_s. In lines 8 to 16, the algorithm determines active zones which will host new service replicas as follows: (i) it calculates the average number of received service requests, $avsr(s, zone_{zh})$, by using the number of received service requests, $sr(s, zone_{zh})$, for each active zone $zone_{zh}$ as shown in line 9; and (ii) based on $avsr(s, zone_{zh})$ values, the algorithm determines which active zones have a high degree of interest as shown in if statement in line 10. As a result, if $avsr(s, zone_{zh})$ is larger than or equal to 1, then $DoI(s, zone_{zh})$ of this active zone $zone_{zh}$ is high

Algorithm 1 DAR

/*Replication mechanism at server s*/

```
$nz_s = 0$
$totalNodes = 0$
for each $zone_{zh} \in ART$ do
    $totalNodes = totalNodes + n_{zone_{zh}}$
    $nz_s = nz_s + 1$
end for
$an_s = totalNodes/nz_s$
for each $zone_{zh} \in ART$ do
    $avsr(s, zone_{zh}) = sr(s, zone_{zh})/an_s$
    if $avsr(s, zone_{zh}) \geq 1$ then
        $DoI(s, zone_{zh}) = high$
        $replicateService(s, zone_{zh})$
    else
        $DoI(s, zh) = low$
    end if
end for
```

Algorithm 2

/*Termination mechanism at zone head zh*/

```
if $|SS_{zone_{zh}}| > 1$ then
    $bs = bestServer(SS_{zone_{zh}})$
    for each $s \in SS_{zone_{zh}}$ do
        if $s \neq bs$ then
            $shutdown(s)$
        end if
    end for
end if
```

and the service is replicated to this active zone as shown in lines 11 and 12. Otherwise, $DoI(s, zone_{zh})$ of this active zone is low and the service is not replicated to it as shown in line 14.

Algorithm 2 shows the termination mechanism at zone head zh which is described as follows: in line 1, the zone head zh checks the number of servers in its zone. Here, $|SS_{zone_{zh}}|$ denotes the number of servers in zone $zone_{zh}$. If there are more than one server, the zone head zh selects the best server based on server's resources (e.g. residual energy, available memory, number of neighbors, etc.) as shown in line 2. In lines 3 to 5, the zone head zh terminates the other servers.

2.5 HDAR: Highly Distributed Adaptive Service Replication Algorithm

2.5.1 Basic Idea

Similarly to our previous method DAR, HDAR divides the whole network into disjoint zones. A major goal of constructing zones is to organize all zone heads into a virtual backbone network to simplify the system control, decrease message overhead, and manage the service replication mechanism. HDAR forms zones and selects zone heads according to a mobility metric because the zone-based structure requires stable zones even in the presence of node mobility. Thus, HDAR selects a node with minimum moving speed in each zone as a zone head, and constructs a virtual backbone network connecting all zone heads. By using our zone structure, HDAR provides dynamic replication that determines the number and location of new service replicas. HDAR aims to replicate a service dynamically to some of zones depending on (i) the service demand level and (ii) the tradeoff between communization and replication costs for each zone. Table 2 shows the protocol design and behavior differences between DAR and HDAR.

The zone formation and maintenance is similar to the formation process in our previous method DAR.

2.5.2 The HDAR Replication Mechanism

In HDAR, each client asks its zone head zh to forward the service request message. After that zh adds to the message the number of nodes in its zone, $n_{zone_{zh}}(t)$, and

Protocol	**Zone-based**	**Replication mechanism parameters**	
	architecture	**Service demand level of a zone**	**Communication and replication cost tradeoff estimation**
DAR	yes	yes	no
HDAR	yes	yes	yes

Table 2: HDAR and DAR Comparison.

Table 3: Aggregated Request Table, ART. in HDAR

zone Id $zone_{zh}$	**number of received service requests** $sq(s, zone_{zh}, t)$	**number of nodes** $n_{zone_{zh}}(t)$	**number of hops** $d(s, zone_{zh}, t)$
$zone_{10}$	6	6	2
$zone_{16}$	3	7	4
$zone_{18}$	10	8	7

forwards the message toward the nearest server. In HDAR, each server s can serve multiple zones, so it maintains an $Aggregated\ Request\ Table$ (ART) which stores the zone identifier $zone_{zh}$, the number of received service requests $sq(s, zone_{zh}, t)$, the number of nodes $n_{zone_{zh}}(t)$, and the number of hops $d(s, zone_{zh}, t)$ for every active zone $zone_{zh}$ which accesses s, as shown in Table 3. As described in the service model in Section 2.3.2, the server s can know received service demand information from the received service request messages.

In HDAR, the servers exchange some information with each other to know the current service demand in the entire network. This information will help to choose an appropriate number of new service replicas. In such a dynamic network, it is not suitable to exchange information among all servers because of the wastage of network resources. To meet this challenge in a suitable manner, HDAR introduces r-level coverage confirmation mechanism where r is a control parameter which determines the level of a neighboring server, (which will be described later in section 2.5.2.1).

In HDAR, the replication mechanism depends on three factors: (i) the number of received service requests from all clients in each active zone $sq(s, zone_{zh}, t)$, (ii) the number of nodes in this active zone $n_{zone_{zh}}(t)$, and (iii) the tradeoff between communication and replication costs for this active zone. The first two factors with the received confirmation messages information determine the *degree of interest* of every active

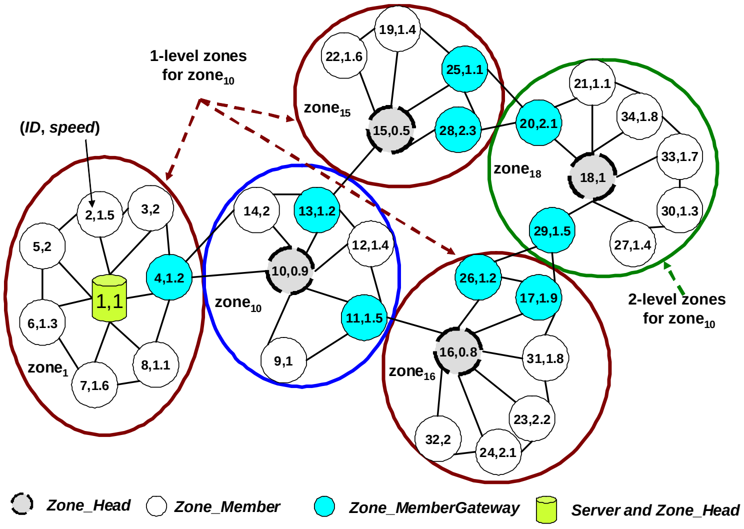


Fig. 5: r-level neighboring zones

zone $zone_{zh}$ which is denoted by $DoI(s, zone_{zh}, t)$. Based on $DoI(s, zone_{zh}, t)$ and the tradeoff between communication and replication costs, server s selects new replica zones. In the next subsections, first, we will explain the coverage confirmation mechanism and how HDAR determines for each zone the levels of other zones. Secondly, we will show how HDAR determines the value of $DoI(s, zone_{zh}, t)$. Finally, we will show how HDAR estimates the communication and replication costs for each active zone.

2.5.2.1 Zone Level and Coverage Confirmation Mechanism In zone-based architecture, every zone head zh has a set of nodes with $Zone\_MemberGeteway$ state. From hello messages received from this set of gateway nodes, each zh can know all neighboring zones. In HDAR, this set of neighboring zones is called *1-level zones* of zh. For each 1-level zone of zh, the set of its neighboring zones, which are not neighbors of zh, is called *2-level zones* of zh. For each $h \geq 3$, *h-level zones* of zh are defined similarly. For example, as shown in Fig. 5, the 1-level zones of $zone_{10}$ are $zone_1$, $zone_{15}$, and $zone_{16}$, while its 2-level zone is $zone_{18}$.

In HDAR, each server s sends a confirmation message to all neighboring servers of at most r-level as follows: (a) the server s sends the message to its zone head zh, (b) zh uses its set of gateway nodes to send this message to 1-level neighboring zones,

(c) each zone head zh' in 1-level zones received the message, will decrease r by 1 and sends the message to its 1-level zones, and (d) steps (b) and (c) will be repeated until r is equal to 0. Note that, any zone head discards the same message if it is received again. The confirmation message contains: its id, its zone id, its received service demand, the number of its clients, and the number of its active zones. By using the received confirmation messages, server s constructs a *Confirmation Table (CFT)* which stores the following information of each neighboring server: its id, its received service demand, the number of its clients, the number of its active zones, and its server level as shown in Fig. 6.

2.5.2.2 Service Demand Level of Active Zone First, server s determines the average number of nodes for all active zones at time t which is denoted by $an_s(t)$ and computed by:

$$an_s(t) = \frac{\sum\limits_{zone_{zh} \in ART} n_{zone_{zh}}(t) + \sum\limits_{sid \in CFT} tClients_{sid}(t)}{nz_s(t) + nf_s(t)} \tag{9}$$

where $nz_s(t)$ is the total number of active zones (the number of records in ART) that access s and $nf_s(t)$ is the total number of active zones which access the neighboring servers of at most r-level.

Secondly, we compute the service demand level, $sdl(s, zone_{zh}, t)$, for each active $zone_{zh}$ which accesses s as follows:

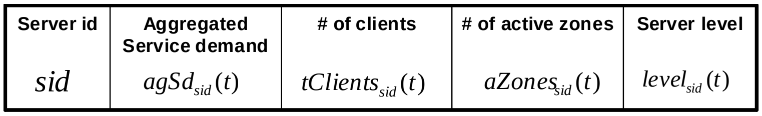
$$sdl(s, zone_{zh}, t) = \frac{sq(s, zone_{zh}, t)}{an_s(t)} \tag{10}$$

Finally, $DoI(s, zone_{zh}, t)$ is defined as:

$$DoI(s, zone_{zh}, t) = \begin{cases} \text{high} & \text{if } sdl(s, zone_{zh}, t) \geq SDL_{th}, \\ \text{low} & \text{if } sdl(s, zone_{zh}, t) < SDL_{th} \end{cases} \tag{11}$$

where, SDL_{th} is a predefined a service demand level threshold.

2.5.2.3 Communication and Replication Costs Estimation In HDAR, each server s uses its ART contents to estimate the communication and replication costs for each active zone zh. The server s estimates the communication cost of the active zone $zone_{zh}$ by the following equation:



Server id	Aggregated Service demand	# of clients	# of active zones	Server level
sid	$agSd_{sid}(t)$	$tClients_{sid}(t)$	$aZones_{sid}(t)$	$level_{sid}(t)$

Fig. 6: Confirmation Table (CFT) fields

$$com(s, zone_{zh}, t) = (E_{tr} + E_{rx}) \times sq(s, zone_{zh}, t) \times [d(zone_{zh}, s, t) + d(s, zone_{zh}, t)] \quad (12)$$

and estimates the replication cost of the active zone zh by the following equation:

$$rep(s, zone_{zh}, t) = k \times (E_{tr} + E_{rx}) \times d(s, zone_{zh}, t) \quad (13)$$

where k is the number of service packets (program and data).

2.5.2.4 HDAR Replication Rule Based on $DoI(s, zone_{zh}, t)$ and the estimation of communication and replication costs, we propose HDAR replication mechanism. In this mechanism, every server s selects multiple active zones without replicas to host new replicas once the replication process is triggered. These active zones with replicas are determined based on the following rule:

"*Replicate a service hosted at a server s to an active zone* $zone_{zh}$ *if* $DoI(s, zone_{zh}, t)$ *is high or if* $rep(s, zone_{zh}, t)$ *is less than* $com(s, zone_{zh}, t) \times RI$*.*"

In this rule, the new replica zones are selected if: (i) their degree of interest is high which means that the service is highly demanded in these zones or (ii) the estimation of their replication costs are lower than their communication costs, which means that the new replicas will be resulted in lower cost compared to the communication cost.

Algorithm 3 shows the behavior of servers to realize the proposed replication mechanism. Every active server s executes this algorithm every replication interval RI based on its ART and CFT contents or the tradeoff between the communication and the replication costs to select new replica zones. In lines 1 to 4, the algorithm initializes the following variables: the number of active zones in ART, $an_s(t)$, the number of active zones in CFT, $nf_s(t)$, the total nodes in ART, $artNodes$, and the total nodes in CFT, $cftNodes$. In lines 5 to 8, the algorithm calculates the number of active zones

Table 4: Calculation Parameters Table.

s	$zone_{zh}$	$sdl(s, zone_{zh}, t)$	$com(s, zone_{zh}, t)$	$rep(s, zone_{zh}, t)$
1	$zone_{10}$	$\frac{6}{7} < 1$	28.8	19.2
1	$zone_{16}$	$\frac{3}{7} < 1$	28.8	38.4
1	$zone_{18}$	$\frac{10}{7} > 1$	168	67.2

$nz_s(t)$ and the total number of nodes in all active zones which exist in ART. In lines 9 to 12, the algorithm calculates the number of active zones $nf_s(t)$ and the total number of nodes in all active zones which exist in CFT. In line 13, the algorithm calculates the average number of nodes in each zone, $an_s(t)$. In lines 14 to 25, the algorithm determines active zones which will host new service replicas as follows: (i) it calculates the average number of received service requests, $sdl(s, zone_{zh}, t)$, by using the number of received service requests sent from each zone, $sq(s, zone_{zh}, t)$, for each active zone $zone_{zh}$ and (ii) based on $sdl(s, zone_{zh}, t)$ values, the algorithm determines which active zones have a high degree of interest as shown in if statement in line 16. As a result, if $sdl(s, zone_{zh}, t)$ is larger than or equal to SDL_{th}, then $DoI(s, zone_{zh}, t)$ of this active zone $zone_{zh}$ is high and the service is replicated to this active zone. If $DoI(s, zone_{zh}, t)$ of this active zone is low, the algorithm replicates the service to this active zone if its $com(s, zone_{zh}, t)$ is larger than its $rep(s, zone_{zh}, t)$ as shown in lines 21 to 23. Otherwise, the service is not replicated.

As an example to show the replication mechanism in HDAR, assume that there is a service at node 1 (in 1's zone as shown in Fig. 4) and its ART as shown in Table 3 at time t. We suppose that the number of service packets (program and data) k is 4 and that E_{tr} and E_{rx} are 1.5 mW and 0.9 mW, respectively. Assume that SDL_{th} is 1. As shown in Fig. 4, $zone_{10}$, $zone_{16}$ and $zone_{18}$ are three active zones for the server 1 (i.e. $nz_1 = 3$) and $zone_{15}$ is a passive zone.

So, we have,

$$\begin{aligned} an_1(t) =& \frac{n_{zone_{10}}(t) + n_{zone_{16}}(t) + n_{zone_{18}}(t)}{nz_1(t) + nf_1(t)} \\ =& \frac{6+7+8}{3+0} = 7 \end{aligned} \tag{14}$$

Algorithm 3 $HDAR$

```
/*Replication mechanism at server $s$*/
$nz_s(t) = 0$
$nf_s(t) = 0$
$artNodes = 0$
$cftNodes = 0$
for each $zone_{zh} \in ART$ do
    $artNodes = artNodes + n_{zone_{zh}}$
    $nz_s(t) = nz_s(t) + 1$
end for
for each $sid \in CFT$ do
    $cftNodes = cftNodes + tClients_{sid}(t)$
    $nf_s(t) = nf_s(t) + aZones_{sid}(t)$
end for
$an_s(t) = (artNodes + cftNodes)/(nz_s(t) + nf_s(t))$
for each $zone_{zh} \in ART$ do
    $sdl(s, zone_{zh}, t) = sq(s, zone_{zh}, t)/an_s(t)$
    if $sdl(s, zone_{zh}, t) \geq SDL_{th}$ then
        $DoI(s, zone_{zh}, t) = high$
    else
        $DoI(s, zone_{zh}, t) = low$
    end if
    if $DoI(s, zone_{zh}, t) = high$ or $rep(s, zone_{zh}, t) < com(s, zone_{zh}, t) \times RI$ then
        $replicateService(s, zone_{zh})$
    end if
end for
```

where the algorithm calculates the average density, an_1, of all active zone for the server 1 (here nf_1 is equal to 0, because there is no any neighboring servers to server 1). After that, by using equation (14), the algorithm calculates the average number of received service requests and estimates the communication and replication costs for each active zone by using equations (10), (12), and (13) as shown in Table 4.

By using the calculated values in Table 4, the algorithm determines the degree of interest for each active zone as follows:

$$DoI(1, zone_{10}, t) = low \tag{15}$$

$$DoI(1, zone_{16}, t) = low \tag{16}$$

$$DoI(1, zone_{18}, t) = high \tag{17}$$

Finally, from equations (15), (16), and (17), HDAR selects $zone_{10}$ and $zone_{18}$ to host new replicas because $DoI(1, zone_{18}, t)$ is high and $DoI(1, zone_{10}, t)$ is low but its value of $com(1, zone_{10}, t) \times RI$ is larger than its $rep(1, zone_{10}, t)$ as shown in Table 4. As a result, the network will contain three servers (1, 10 and 18) as shown in Fig. 7. In the next step, each server will execute the replication algorithm based on its ART and CFT contents to select new replica zones. As shown in Fig. 7, if we assume that the active zone of server 10 is $zone_{16}$ and the active zone of server 18 is $zone_{15}$. This means that ART of server 1 is empty, ART of server 10 contains $zone_{16}$, and ART of server 18 contains $zone_{15}$. If we assume that each server sends a confirmation message up to 1-level neighboring servers (i.e. r=1). This means that CFT of server 1 contains information for server 10, CFT of server 10 contains information for server 1, and CFT of server 18 is empty. In this case, servers 1, 10, and 18 execute HDAR mechanism by using contents of their ARTs and CFTs. As a result, server 1 does not replicate the service because there is no active zone with high degree of interest, server 10 may make a decision of replicating a new service into $zone_{16}$, and server 18 may make a decision of replicating a new service into $zone_{15}$, in the future.

2.6 Performance Evaluation

In order to evaluate the effectiveness of DAR and HDAR, we compared their performance with SDP [25] for the following metrics:

Energy consumption: energy consumption is the summation of (a) Service cost: energy amounts consumed for communication and replication defined by objective function (4) and (b) Confirmation cost: energy amounts consumed for confirmation messages and are defined as follows:

$$TotalFCost(S(t), t) = \sum_{s \in S(t)} FrmCost(s, NigS_r(s, t), t) \tag{18}$$

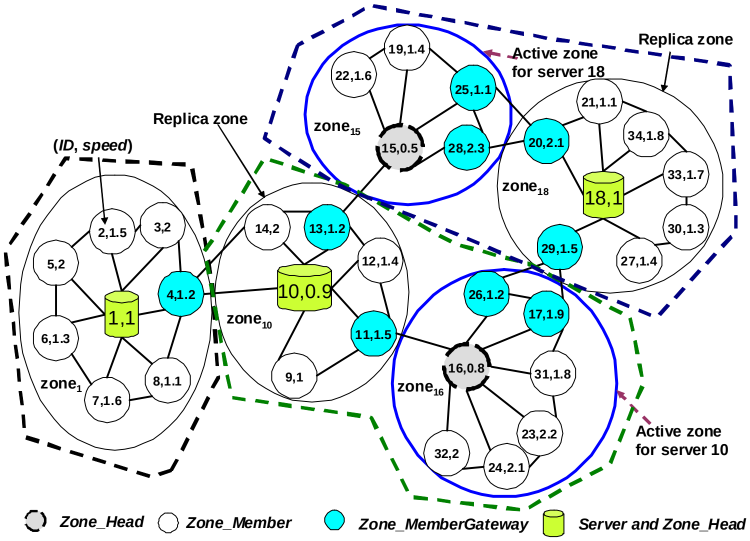


Fig. 7: HDAR Example with created replicas.

where

$$FrmCost(s, NigS_r(s,t), t) \quad = \quad (E_{tr} \quad + \quad E_{rx}) \quad \times \quad \sum_{s' \in NigS_r(s,t)} d(s, s', t) \quad (19)$$

and $NigS_r(s,t)$ is the set of neighboring servers of s at level less than or equal to r. Note that, there is no confirmation cost in case of DAR and SDP.

Replica ratio: ratio of the number of service providers at each point of time during the network lifetime to the total number of nodes in the network. In order to avoid wasting the network resources, a low replica ratio together with high service availability and low energy consumption are desired.

Service availability: we use the same definition of service availability as in [22, 25], that is, the ratio of the number of service replies received to the number of service requests sent during the network lifetime.

2.6.1 Simulation Environment

The QUALNET [33] simulator was used with input parameters as listed in Table 7, such as network size, speed, transmission range, simulation time, etc. In addition, the node mobility was based on two mobility models: (i) Random way point (RWP) mobility model [34] where the minimum speed was 0 $meter/second$, the pause

Table 5: Configuration Parameters for MANETs.

Configuration parameter	**Value in simulation**
Number of nodes	25 to 200
Maximum node's speed	[1...10] meters/second
Field size	$500m \times 500m$
Transmission range	100 m
Bandwidth	2 $Mbps$ wireless channel
Routing Protocol	*Ad-hoc On-Demand Distance Vector Routing* (AODV) [32]
MAC Protocol	IEEE 802.11b without power control protocol
E_{tr}	1.5 mW
E_{rx}	0.9 mW
Simulation time	700 seconds
SDL_{th}	{0.8, 0.9, 1.0, 1.1, 1.2}
RI	{40, 50, 60, 70 ,80} seconds

time was 0 $second$ (continuous movement) and the maximum speed between [1...10] $meters/second$ and (ii) Realistic mobility model generated by MobiREAL [35] which is a simulator to model and simulate realistic mobility of nodes. To simulate client's requests, each client maintained a requesting rate according to Poisson distribution with average λ which equals to 6 requests in time interval of 60 seconds.

Initially, there was one service with $2Kbytes$ of object size (program and data) and the size of one packet is $512bytes$ (so, the number of service packets k was 4) which was put at a certain node. For HDAR, we experimented three cases by using several different values for r and are denoted by *HDAR-{1,2,5}*. For SDP, we experimented two cases by using several different replication and hibernation thresholds which are denoted by *SDP-RH 4/5-1*, we used both hibernation and replication mechanisms with 1 and 4 or 5 requests as hibernation and replication thresholds, respectively. We repeated every simulation 5 times then averaged the results.

2.6.2 Random Way Point Scenario

In order to show the performance of HDAR, we compared HDAR with SDP [25] and DAR in terms of replica ratio, service availability, and energy consumption against

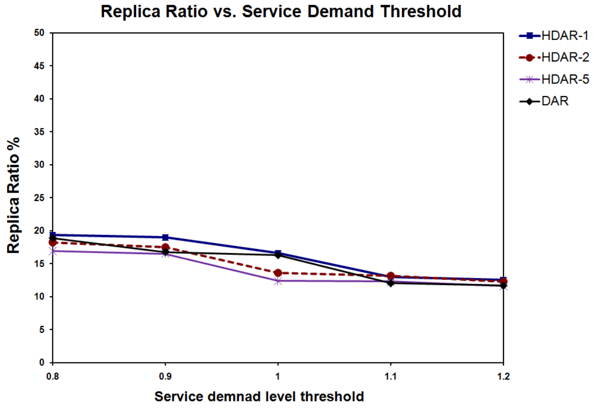


Fig. 8: Replica ratio vs. Service demand threshold

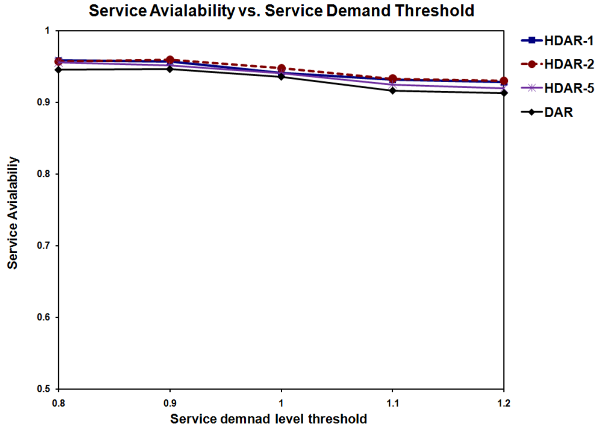


Fig. 9: Service availability vs. Service demand threshold

replication interval, maximum node's speed, and network size, as well as the average hop counts against network size. We show the results in Figs. 8 to 20.

2.6.2.1 Service Demand Threshold Effects Figs. 8, 9, and 10 show the replica ratio, the service availability, and the energy consumption, respectively, against the service demand threshold when maximum node's speed was 1 $meter/second$, and network size was 100.

1) Replica ratio and service availability: As shown in Figs. 8 and 9, the replica ratio and the service availability of all algorithms decreased as the service demand threshold

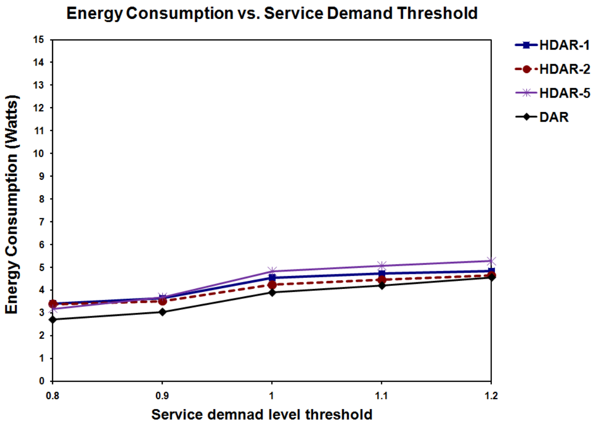


Fig. 10: Energy consumption vs. Service demand threshold

increased as expected.

2) Energy consumption: As shown in Fig. 10, the energy consumption increases as expected as the service demand threshold increases. This is because, the number of replicas in the network decreases.

As a result, $SDL_{th} = 1$ is a good trade-off between service availability and energy consumption. Therefore, we used the $SDL_{th} = 1$ for our experiments.

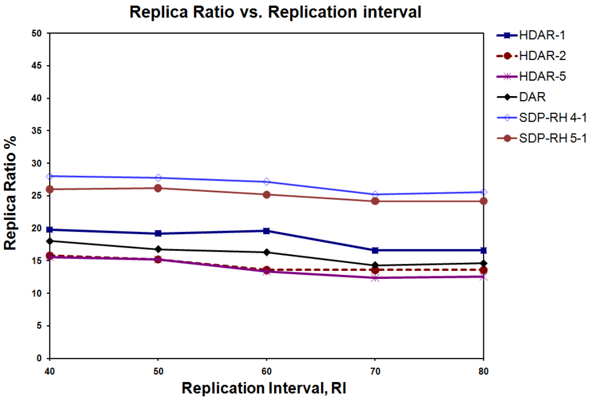


Fig. 11: Replica ratio vs. Replication interval

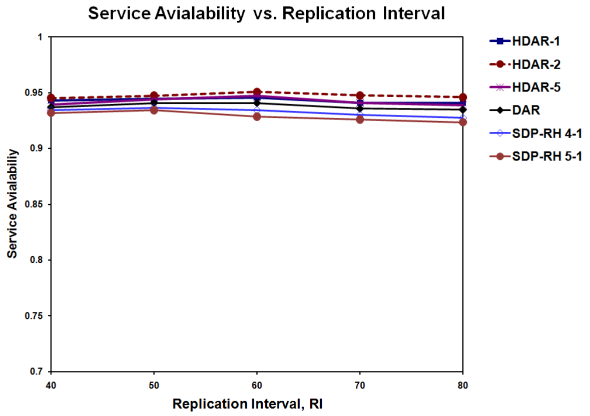


Fig. 12: Service availability vs. Replication interval

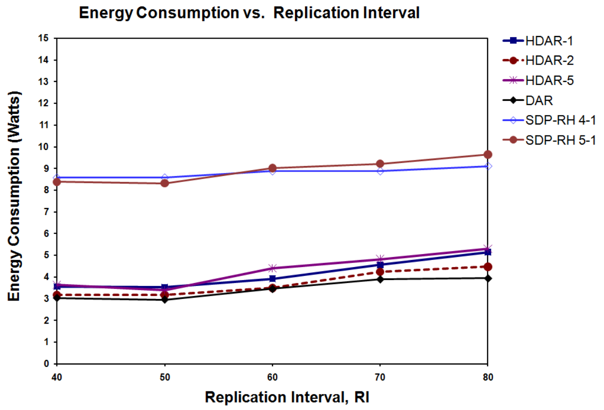


Fig. 13: Energy consumption vs. Replication interval

2.6.2.2 Replication Interval Effects Figs. 11, 12, and 13 show the replica ratio, the service availability, and the energy consumption, respectively, against the replication interval when maximum node's speed was 1 $meter/second$, and network size was 100.

1) Replica ratio and service availability: As shown in Fig. 11, the replica ratio of all algorithms decreases as the replication interval increases. This is because, the number of executed replication processes to create new replicas decreases. The replica ratio for SDP was higher than DAR and HDAR. In addition, the replica ratio for SDP was affected by values of replication and hibernation thresholds. That is, the number of replicas decreased as the replication threshold increased. In case of HDAR, the replica ratio is less affected by the replication interval when r was 2. As shown in Fig. 12, the service availability of all algorithms were less affected when the replication interval increased. The service availability of HDAR was higher than DAR and SDP and was between 0.94 and 0.95. In case of DAR, the service availability was between 0.93 and 0.94, while the service availability of SDP was between 0.92 and 0.93 when replication threshold was 4 requests. Also, the service availability of SDP was affected by replication and hibernation thresholds and decreased as the replication threshold increased.

2) Energy consumption: As shown in Fig. 13, the energy consumption increased as expected as the replication interval increased. This is because, when the replication interval increases, the communication cost increases. The energy consumption for DAR was much lower than SDP. In addition, the energy consumption for SDP was affected by values of replication and hibernation thresholds. In case of HDAR, the energy consumption was much lower than SDP but higher than DAR. This is because, HDAR consumes additional energy for confirmation messages between servers for different values of r. When r was 2, the energy consumption was less than other values of r.

As a result, when r was 2, HDAR achieved higher service availability than DAR and SDP with reasonable energy consumption and less affected by the replication interval. In addition, the replica ratio was less affected than other values of r.

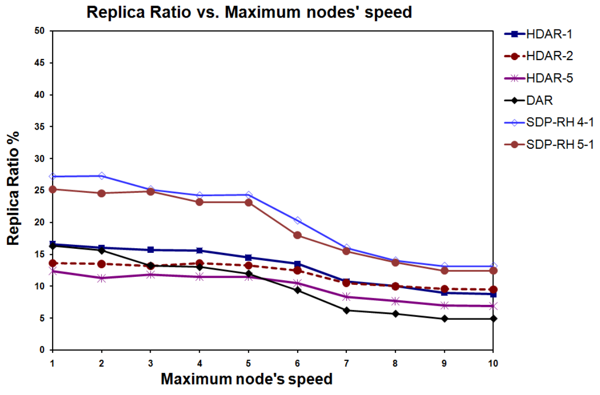


Fig. 14: Replica ratio vs. Maximum node's speed

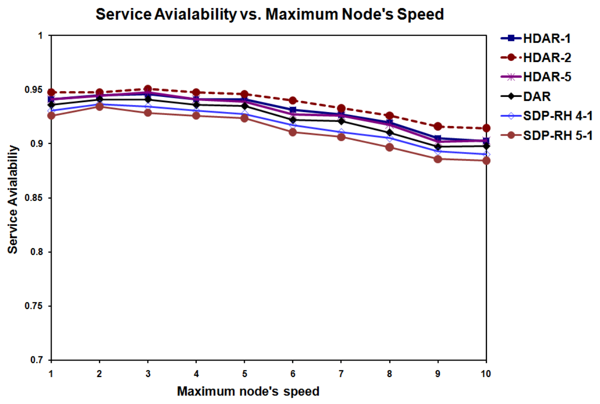


Fig. 15: Service availability vs. Maximum node's speed

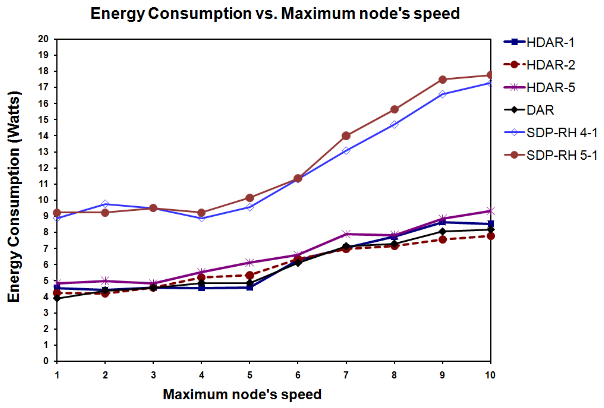


Fig. 16: Energy consumption vs. Maximum node's speed

2.6.2.3 Maximum Node's Speed Effects Figs. 14, 15, and 16 show the replica ratio, the service availability, and the energy consumption, respectively, against the maximum node's speed when the network size was 100 and the replication interval was 70 seconds.

1) **Replica ratio and service availability:** As shown in Fig. 14, the replica ratio of all algorithms decreased as the maximum node's speed increased. For low speed (from 1 to 5 *meters/second*), the replica ratio decreased slowly. For high speed (from 6 to 10 *meters/second*), the replica ratio decreased quickly. As shown in Fig. 15, the service availability of all algorithms decreased as the maximum node's speed increased. For low speed (from 1 to 5 *meters/second*), the service availability decreased slowly. For

high speed (from 6 to 10 *meters/second*), the service availability decreased quickly. However, HDAR achieved higher service availability than DAR and SDP. This is because, HDAR considers the service demand at neighboring servers and the tradeoff between the communication and replication costs, while DAR and SDP do not. In addition, when r was 2, HDAR achieved higher service availability than other values of r.

2) Energy consumption: As shown in Fig. 16, the energy consumption increased as the maximum node's speed increased. This is because, when the maximum node's speed increases, the locations of replicas changes rapidly and the number of replicas decreases. So, the communication cost increases. The energy consumption for DAR and HDAR were much lower than SDP. In addition, the energy consumption for SDP was affected by values of replication and hibernation thresholds. In case of HDAR, the energy consumption was much lower than SDP but higher than DAR. This is because, HDAR consumes additional energy for confirmation messages between servers for different values of r.

As a result, when r was 2, HDAR achieved higher service availability than DAR and SDP with reasonable energy consumption.

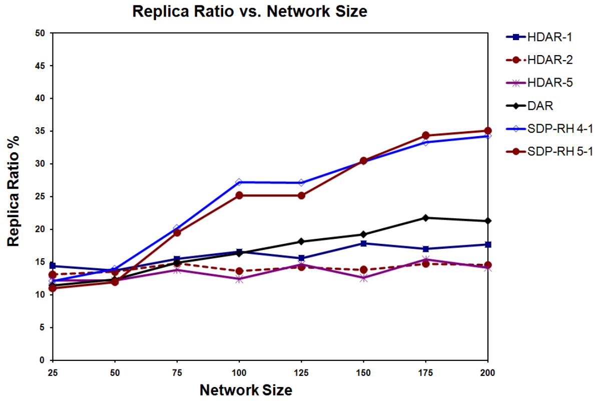


Fig. 17: Replica ratio vs. Network size

2.6.2.4 Network Size Effects Figs. 17, 18, 19, and 20 show the replica ratio, the service availability, the energy consumption, and the average hop counts, respectively,

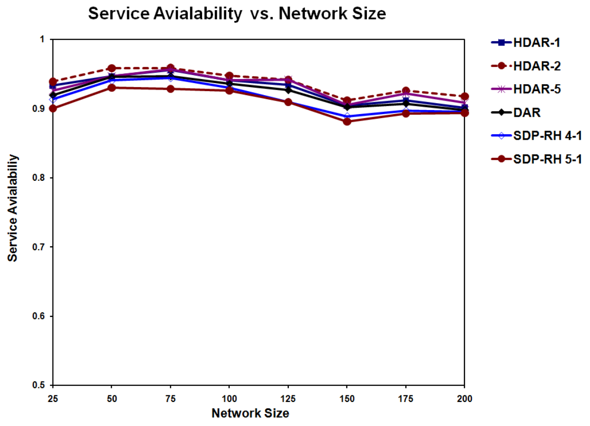


Fig. 18: Service availability vs. Network size

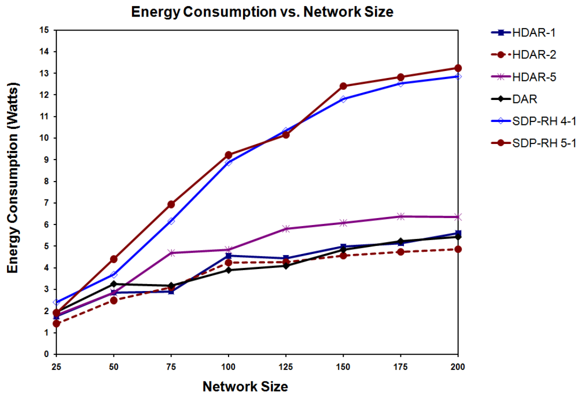


Fig. 19: Energy consumption vs. Network size

against the network size when the maximum nod's speed was 1 $meter/second$ and the replication interval was 70 seconds.

1) Replica ratio and service availability: As shown in Fig. 17, the replica ratio of DAR and SDP increased as the network size increased. This is because, when the number of nodes increased, the number of requests increased and new replicas were created. However, DAR showed better scalability than SDP. In addition, the replica ratio for SDP was affected by values of replication and hibernation thresholds. That is, the number of replicas decreased as the replication threshold increased. On the other hand, the replica ratio of HDAR was fluctuated but almost independent of the

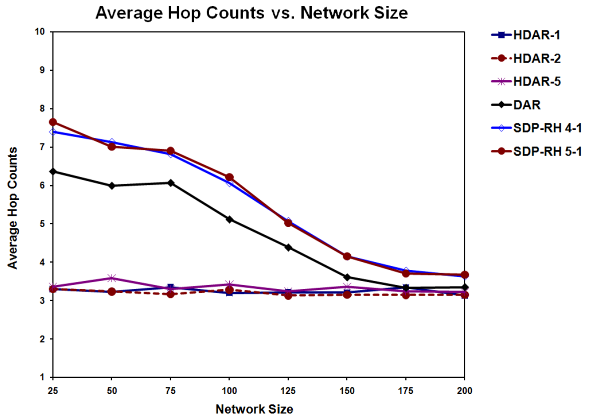


Fig. 20: Average hop counts vs. Network size

network size. This is because, HDAR uses the confirmation messages between servers and considers the tradeoff between the communication and the replication costs for each zone. Also, when r was 2 the replica ratio is less fluctuated than other values of r. As shown in Fig. 18, the service availability of DAR was higher than SDP and the service availability of SDP was affected by replication and hibernation thresholds and decreased as the replication threshold increased. On the other hand, the service availability of HDAR was higher than DAR and SDP, because HDAR considers the service demand at the neighboring servers and the tradeoff between the communication and replication costs, while DAR and SDP do not.

2) Energy consumption: As shown in Fig. 19, the energy consumption increased as expected as the network size increased. This is because when the number of nodes increased, the number of requests increased and new replicas were created with additional cost. The energy consumption for DAR was scalable and kept the energy consumption much lower than SDP. In addition, the energy consumption for SDP was affected by values of replication and hibernation thresholds. In case of HDAR, the energy consumption was much lower than SDP but higher than DAR. This is because, HDAR consumes additional energy for confirmation messages between servers for different values of r. When r was 2, the energy consumption was almost less than other values of r.

As a result, when r was 2, HDAR achieved higher service availability than DAR and SDP with reasonable energy consumption. In addition, the replica ratio was less fluctuated than other values of r.

3) Average hop counts: As shown in Fig. 20, the average hop counts of DAR and SDP decreased as the network size increased. This is because, when the number of nodes increased, the number of replicas increased and the average hop counts decreases. Also, the average hop counts of DAR was lower than SDP. On the other hand, the average hop counts for HDAR was lower than DAR and SDP and was independent of the network size. This is because, HDAR considers the service demand at the neighboring servers and the tradeoff between the communication and replication costs, while DAR and SDP do not.

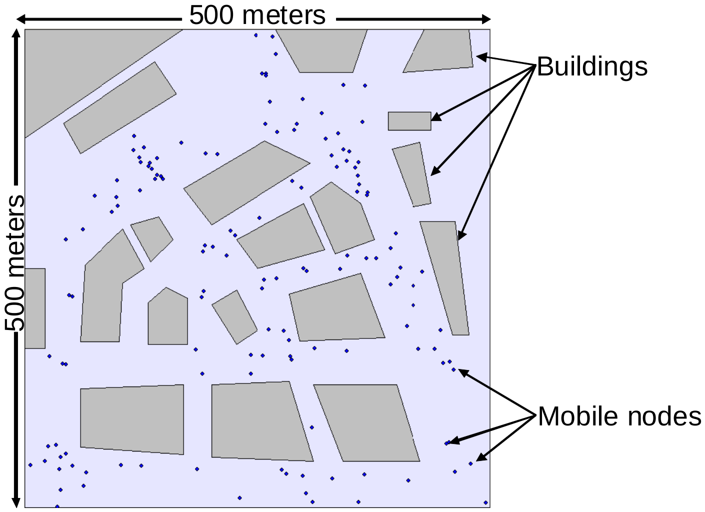


Fig. 21: Urban map for MobiReal simulator

2.6.3 Realistic Mobility Scenario

In this scenario, we used MobiREAL simulator [35] to generate a realistic mobility model where 100 nodes were distributed in the simulation field as shown in Fig. 21. MobiREAL is a simulator to reproduce the user traffic based on the observation of the actual traffic density at each street. This simulator allows to describe how mobile nodes change their destinations, routes and speeds/directions based on their positions,

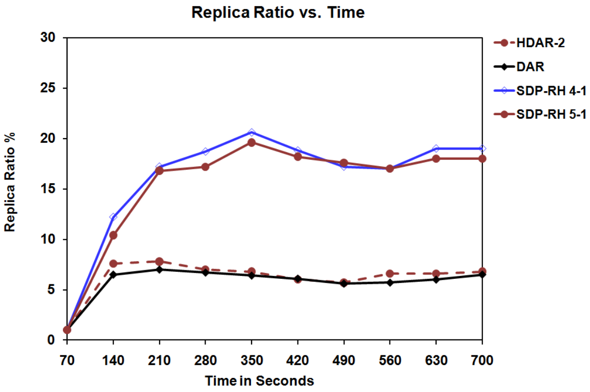


Fig. 22: Replica Ratio vs. Time

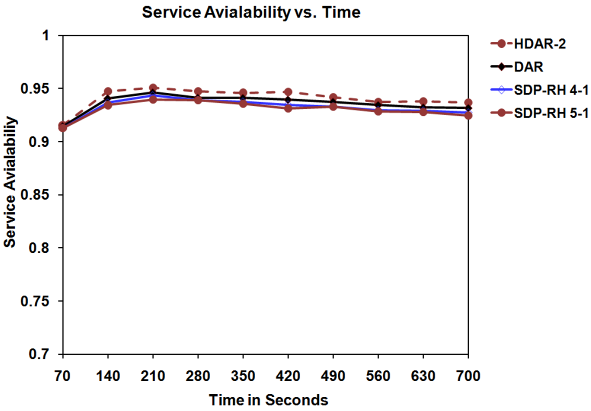


Fig. 23: Service availability vs. Time

surroundings such as obstacles (e.g. buildings) and neighboring nodes. In our simulations, realistic mobility means that the movement of mobile nodes in realistic environment through the incorporation of obstacles and the construction of realistic movement paths.

In order to show the effect of the realistic mobility[1], we compared the proposed protocol with DAR and SDP when network size was 100 nodes, r was 2 (the best value for r in case of random way point mobility model), the replication interval was 70 seconds, the initial speed was 1 *meter/second*, and the other parameters were the same as in random way point model. For SDP, we experimented for two cases, SDP-

[1] We did not consider the effect of obstacles such as buildings on the radio strength between nodes.

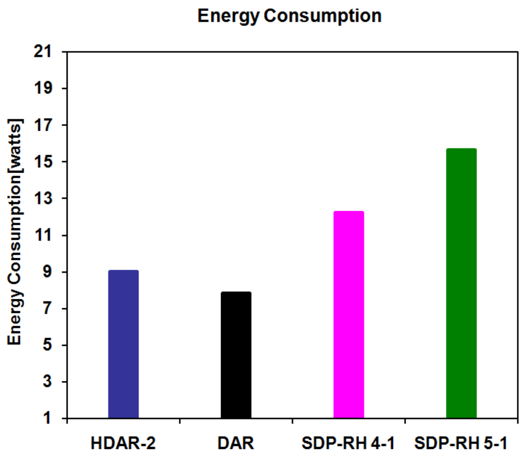


Fig. 24: Energy consumption through network lifetime

RH 4/5-1, that were defined in the previous section. We show the results in Fig. 22, Fig. 23, and Fig. 24.

1) Replica ratio and service availability: As shown in Fig. 22, the replica ratio of SDP fluctuated overtime. While, for HDAR and DAR the replica ratio was more stable than SDP. In addition, the replica ratio for SDP was affected by values of replication and hibernation thresholds. As shown in Fig. 23, the service availability of HDAR was higher than DAR and SDP and was between 0.91 and 0.95.

2) Energy consumption: As shown in Fig. 24, the energy consumption for DAR was much lower than SDP. In addition, the energy consumption for SDP was affected by values of replication and hibernation thresholds. In case of HDAR, the energy consumption was much lower than SDP but higher than DAR. This is because, HDAR consumes additional energy for confirmation messages between servers where DAR did not use such messages.

As a result, HDAR achieved higher service availability than DAR and SDP with reasonable energy consumption.

2.7 Conclusion for Replication Methods

In this chapter, two distributed adaptive service replication methods for MANETs were presented. Our protocols first divide the whole network into disjoint zones with diameters of at most 2 hops, select a node with minimum moving speed in each zone as a zone head, and construct a virtual backbone network connecting all zone heads. By using this zone structure, our protocols select a new replica node according to the topology, number of requests from clients, and the tradeoff between communication an replication costs for each zone. Our protocols are scalable and it can control the locations and the number of service replicas, keeping the network-wide energy consumption as low as possible and improving the service availability. Simulations demonstrated that our methods improves the performance of service provision in terms of the energy consumption and service availability compared with existing methods. In addition, the path length between a client and a server is minimized independently of network size.

3. Probabilistic Coverage Methods in People-Centric Sensing

3.1 Introduction

PCS realizes that people with mobile devices can act as mobile sensors to sense and gather information from the environment to serve sensing applications and their users. In PCS, the coverage depends on the uncontrollable mobility of people, therefore it is difficult to achieve full coverage of the target AoI. Consequently, it is preferable to measure the expected coverage degree as a ratio.

Here, we describe our motivation with the following application scenario. The realtime urban sensing scenarios drive an interesting motivating application. In a city sensing application, for instance, users want to know the information in a specific AoI such as interesting spots, crowded places, events on specific locations, and so on. A local torrential rain prediction is another application scenario as shown in Fig. 25, where a weather news company uses PCS to predict 1 hour before occurrence. In such applications, a user/comapny sends a query about a geographic area as the AoI, a required coverage ratio α (i.e., the required percent of coverage of the AoI), the required information (e.g., noise level), and a query interval (maximum allowable response time) T. Then, the query responding process will be carried out by some people with mobile devices in the AoI, which satisfy the query requirements. Here, to minimize cost, it is desirable to select a minimal number of people with mobile devices that can provide the desired information. We refer to this problem as the *(α, T)-coverage problem.*

Here, we formally describe the (α, T)-coverage problem as follows. Given a target field that is composed of a set of points, an AoI as a subset of it, a set of mobile nodes, and a query with a required coverage ratio α and a specified time interval T, we define the (α, T)-coverage problem to be the problem of finding a minimal set of mobile nodes such that each point in the AoI is visited and sensed by at least one node within T with a probability of at least α. To solve this problem, we need to predict the future locations visited by each mobile node depending on its current location when a query is initiated and its mobility. Thus, we model the mobility of the mobile nodes with a discrete Markov chain. The solution for this problem depends critically on the number and the initial locations of mobile nodes inside and near the target AoI when a query is

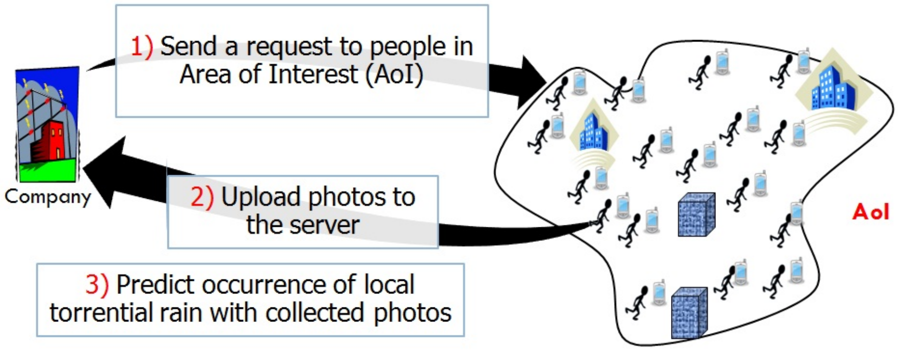

1) Send a request to people in Area of Interest (AoI)
Company
2) Upload photos to the server
3) Predict occurrence of local torrential rain with collected photos
AoI

Fig. 25: Application scenario for sensing coverage in PCS

initiated. One possible solution for this problem is the random selection of nodes. The main drawback of random selection is inefficiency by selecting a set of nodes that are likely to visit the same locations in AoI in the future and this set may not be minimal to achieve the coverage. To avoid this drawback, we should carefully select a minimal set of nodes that are not likely to visit the same locations.

3.2 Related work

Many studies have proposed data gathering protocols to realize efficient communication between sensor nodes in wireless sensor networks (WSNs) [37–40]. Some studies also have proposed the use of mobile sensor nodes in WSNs to improve coverage, lifetime, and/or fault-tolerance [41, 42].

Recently, information collection by pedestrians in PCS has received increasing attentions. PCS is different from existing WSNs because we cannot control the mobility of mobile nodes. In addition, the two important criteria in PCS are coverage of the AoI and time. There are several studies and research projects based on PCS [43–50].

Cartel [43] is a mobile communications infrastructure based on car-mounted communication platforms exploiting open WiFi access points in a city, and provides urban sensing information such as traffic conditions. CitySense [44] provides a static sensor mesh offering similar types of urban sensing data feeds. SensorPlanet [45] is a platform that enables the collection of sensor data on a large and heterogeneous scale, and establishes a central repository for sharing the collected sensor data. Bubble-sensing [46] is a sensor network that allows mobile phone users to create a connection between tasks and places of interest in the physical world. Mobile users are able to affix task bubbles at places of interest and then receive sensed data as it becomes available in a delay-tolerant fashion. PriSense [47] relies on data slicing and mixing and binary search to enable privacy-preserving queries, where each node slices its data into $(n + 1)$ data slices, randomly chooses n other nodes, and sends a unique data slice to each of them. Finally, each node sends the sum of its own slice and the slices received from others to the aggregation server. Anonysense [48] is a privacy-aware architecture for realizing pervasive applications based on collaborative, opportunistic sensing by personal mobile devices. AnonySense allows applications to query and receive context through an expressive task language and by leveraging a broad range of sensor types on mobile devices, and at the same time respects the privacy of users. GreenGPS [50] is a navigation service that uses participatory sensing data to map fuel consumption on city streets and find the most fuel-efficient route for vehicles between arbitrary endpoints.

Most of these approaches focus on information collection, but do not consider the probabilistic coverage in PCS when the information collection period is restricted to a short time duration such as an on-demand query. They consider neither the difficulties of achieving sensing coverage of a relatively wide area nor the time requirements of on-demand sensing by mobile users. However, these two criteria are very important in PCS. To meet these criteria, it is also very important to estimate the area covered by each mobile node in a specified time interval. However, existing studies do not consider such a spatiotemporal coverage by mobile nodes.

In this chapter, we propose two probabilistic algorithms: *inter-location* and *inter-meeting-time* algorithms, to meet a coverage ratio α in time period T. The inter-location algorithm called ILB estimates the probability of locations in the AoI being visited by each mobile sensor node in T, and selects a minimal number of mobile nodes inside the AoI considering the distance between the nodes. The inter-meeting-time algorithm called IMTB selects a minimal number of nodes regarding the expected time until any two of the nodes will meet at a location. Sometimes, the required coverage may not be achieved due to an insufficient number of nodes existing inside the AoI. To meet the required coverage in this case, we also propose an extended algorithm which takes into account not only nodes existing inside the AoI, but also nodes outside the AoI.

The future estimated location of each node could be inaccurate when T is large, resulting in inaccurate coverage. For more accurate coverage, we propose an updating mechanism for the inter-location and the inter-meeting-time algorithms which aims to remove useless nodes and add some extra nodes that contribute more to AoI coverage. This updating mechanism is periodically executed every specified time interval during T.

In the rest of this chapter, we first describe assumptions, models, and definitions in Section 3.3, then we introduce our proposed algorithms in Section 3.4.2. In Section 3.5, we describe simulation experiments to evaluate the performance of the proposed algorithms for various parameter settings.

3.3 Assumptions, Models, and Definitions on Target PCS

3.3.1 System Model

We assume an application such that when requested, some of the mobile users take part in a task to obtain the latest environmental information such as noise level, sunshine intensity, temperature, exhaust gas concentration, and so on, over a specified geographical area of the urban district in a PCS fashion. We assume that those participating users are willing to serve as mobile sensors based on some incentive such as electronic currency or coupons given by a service provider.

We denote the whole service area by A. A road (street) network on which mobile users can move spans the area A. A service user wants to know the approximate condition of a specific area called the *Area of Interest (AoI)* produced by obtaining the environmental information about some locations in the AoI. Thus, we assume that there are multiple *sensing locations* with a uniform spacing Δ[2] (e.g., $\Delta = 50m$) on each road and that sensing coverage is achieved by obtaining the environmental information about all of the sensing locations in the specified AoI. We show an example road network with sensing locations in a service area in Fig. 26.

We represent the road network with sensing locations by a connected graph $G = (V, E)$, where V is the set of vertices corresponding to sensing locations (some of them are intersections) and E is the set of edges corresponding to segments between neighboring sensing locations on roads.

[2] We assume that each road can be divided into an integer number of segments with length Δ.

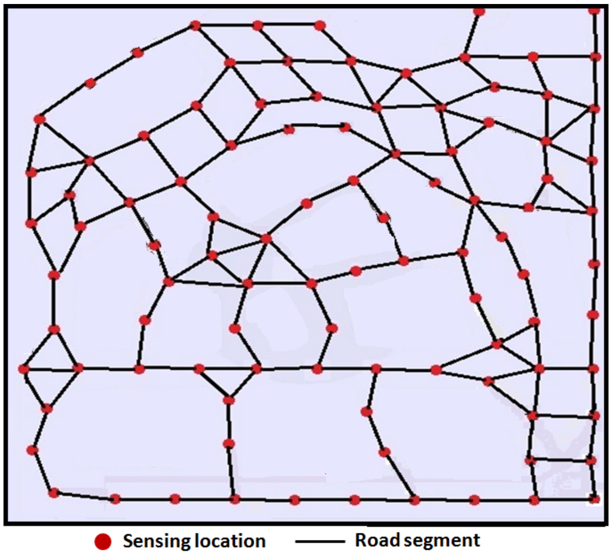


Fig. 26: Example of road network

Multiple service users of this application exist on service area A and are moving on graph G. Each mobile user is equipped with a portable computing device such as smartphones capable of accessing the Internet via a cellular network (CDMA, GSM) from any place in A, measuring the current location, and sensing the nearby environmental information with its built-in sensors (camera, microphone, light-intensity sensor, etc). Hereafter, we refer to a service user with a mobile device simply as a *node*.

We assume that time progresses discretely $(0, 1, 2, ...)$. Let U denote a set of nodes on G at time 0 (i.e., the time when a query is initiated). Each node moves from one vertex to one of its neighboring vertices on G in a unit of time. The mobility of nodes is based on a probabilistic model. Let $v_0^u \in V$ denote the initial (at time 0) location of node u. Let $Prob(u, t, v_0^u, v_t)$ denote the probability that each node u with its location v_0^u at time 0 visits a vertex $v_t \in V$ at time t.

3.3.2 Service Model

We assume that our target application provides users with an on-demand query service for sensing a specific AoI and we treat a single query at a time. We assume that there

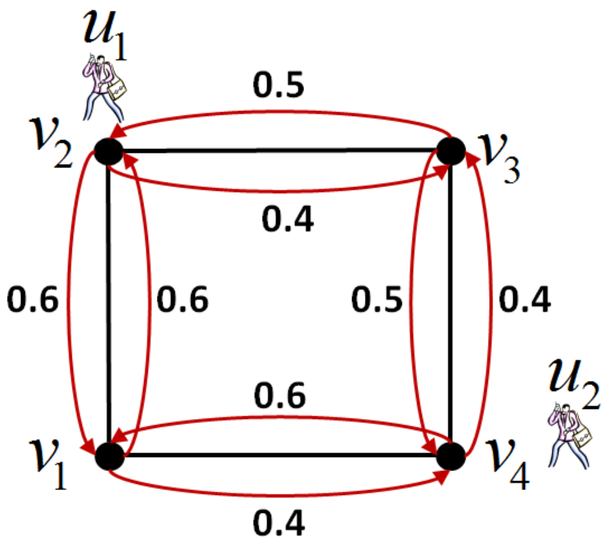


Fig. 27: Moving probability example for four sensing locations $\{v_1, v_2, v_3, v_4\}$ and initial locations of two nodes $\{u_1, u_2\}$.

is a fixed server s in the Internet that can communicate with nodes of U and executes required tasks.

We say that the AoI is α-*covered* if every sensing location in the AoI is visited (and thus the environmental information is sensed) by at least one node with a probability of at least α. Here, we call α the *coverage ratio*. In our application, a node sends s a query which asks for sensing a specified AoI with a specified coverage ratio α in a specified time interval T. We denote each query q by a quadruple $\langle AoI, S_{type}, \alpha, T\rangle$. Here, AoI is the area of interest in the service area specified by a set of sensing locations of V, and S_{type} specifies the type of environmental information to be sensed, such as temperature.

3.3.3 Definition of (α, T)-Coverage

We call the probability of a set of nodes $U'(\subseteq U)$ visiting a sensing location v $(\in V)$ in a time interval T, *the set coverage probability* denoted by $SetProb(v, U', T)$ and define it by the following equation.

Sensing location	**node u_1** $Prob(u_1, t, v_2, v)$			**node u_2** $Prob(u_2, t, v_4, v)$			$SetProb(v, U', 2)$
v	$t = 0$	$t = 1$	$t = 2$	$t = 0$	$t = 1$	$t = 2$	$U' = \{u_1, u_2\}$
v_1	0	0.6	0	0	0.6	0	$1 - (1 - 0.6)(1 - 0.6) = 0.84$
v_2	1	0	0.56	0	0	0.56	$1 - (1 - 1)(1 - 0.56)(1 - 0.56) = 1$
v_3	0	0.4	0	0	0.4	0	$1 - (1 - 0.4)(1 - 0.4) = 0.64$
v_4	0	0	0.44	1	0	0.44	$1 - (1 - 0.44)(1 - 1)(1 - 0.44) = 1$

Table 6: Visiting time and set coverage probabilities for the example in Fig. 27 with T = 2.

$$SetProb(v, U', T) = 1 - \prod_{u \in U'} \prod_{t=0}^{T} (1 - Prob(u, t, v_0^u, v)) \tag{20}$$

Fig. 27 shows an example for four sensing locations v_1, v_2, v_3, and v_4 and the moving probabilities between them. As shown in Fig. 27, there are initially two nodes u_1 and u_2 at sensing locations v_2 and v_4, respectively. Table 6 shows the set coverage probabilities of v_1, v_2, v_3, and v_4 by $U' = \{u_1, u_2\}$ when T = 2.

Definition 1. ***(α, T)*-coverage***: Given a graph $G = (V, E)$, an area specified by a set of sensing locations $AoI \subseteq V$, a set of nodes $U' \subseteq U$, a required coverage ratio α, and a time interval T, the area AoI is called **(α, T)-covered** if the following condition holds.*

$$\forall v \in AoI, SetProb(v, U', T) \geq \alpha \tag{21}$$

We formally define the (α, T)-coverage problem as follows:

Definition 2. *Given a service area as a connected graph $G = (V, E)$, a set of nodes U on G at time 0, and a query $q = \langle AoI, S_{type}, \alpha, T\rangle$, the (α, T)**-coverage problem** is the problem of selecting a minimal set of nodes $U' \subseteq U$ which achieves (α, T)-coverage of AoI.*

We define the objective function of this problem by the following equation.

$$\textbf{minimize } |U'| \tag{22}$$

$$\textbf{subject to } AoI \text{ is } (\alpha,\, T)\text{-}\textit{covered} \tag{23}$$

This problem is NP-hard since it implies, as a special case, the *Minimum Set Covering Problem* (MSCP) which is known to be NP-hard [51,52].

3.4 Proposed Methods

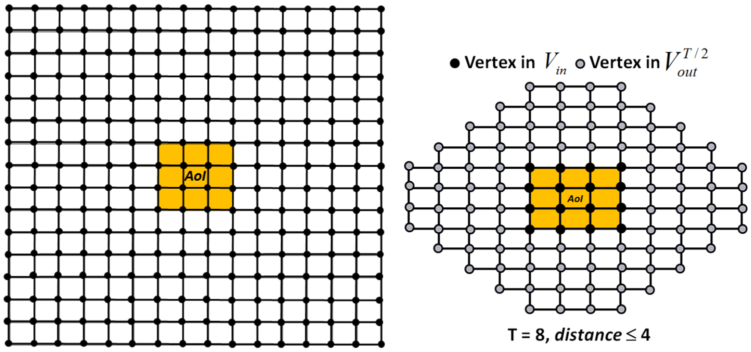
In this section, we propose three heuristic algorithms for to solve this problem, named *Inter-Location Based (ILB)*, *Inter-Meeting Time Based (IMTB)*, and *Extended Algorithm without Thresholds (EWOT)* based on the initial locations of nodes existing inside and near AoI when a query is initiated. For more accurate coverage, we propose an update mechanism for the ILB and IMTB algorithms that aims to adapt the number of selected nodes based on the latest location of nodes. This update mechanism is executed every specified time interval during the time period T. We assume that all algorithms are executed by the server s in a centralized fashion.

3.4.1 Preliminaries

Our basic idea is to select nodes that have higher probabilities of visiting distinct sensing locations in the specified AoI within a time interval T, prior to selecting other nodes.

The proposed algorithms depend on the probability $Prob(u, t, v_0^u, v_t)$ of each node u with initial location v_0^u visiting a location v_t at time t ($0 \leq t \leq T$). To simplify our explanation, we represent the graph $G = (V, E)$ for the service area by a grid of sensing locations (vertices) with a uniform spacing Δ between neighboring vertices and only vertical and horizontal edges (here, each edge is bi-directional), as shown in Fig. 28 (a). Let N denote the number of vertices (i.e., $|V|$) and x_i denote the i-th vertex of $V(1 \leq i \leq N)$. We model the node movement on the grid as a discrete Markov chain. For each node u, we define a vector with N states where the i-th state represents the probability that u is in vertex x_i.

Assuming that there are a sufficient number of nodes in target area A, we select nodes only within the specified AoI. Here, at time 0, we are given a query and the



(a) Original service area graph (b) Reduced service area graph

Fig. 28: An example of a service area graph with AoI and its reduction for T= 8

current distribution of nodes. Let $U_0(\subseteq U)$ denote the set of nodes which are located in the target AoI at time 0.

3.4.1.1 Computation of coverage probability of a vertex Let $\boldsymbol{P}$ denote the probability matrix with size $N \times N$, where its i-th row and j-th column element represents the probability of a node at vertex x_i to move to vertex x_j by a unit of time. We define an initial state vector $\boldsymbol{v_0^u}$ representing that a node u is initially located at $x_i \in V$ by the following equation.

$$\boldsymbol{v_0^u} = (p_1, p_2, ..., p_N) \tag{24}$$

where

$$p_j = \begin{cases} 0 \;\; (j \neq i) \\ 1 \;\; (j = i) \end{cases} \tag{25}$$

Then, we can calculate the coverage probability of vertex $x_k \in V$ by node u at time t by the following equation.

$$Prob(u, t, v_0^u, x_k) = [\boldsymbol{v_0^u} \times \boldsymbol{P}^t]_k \tag{26}$$

Here, $[\;]_k$ denotes the k-th element in the resulted vector.

3.4.1.2 Reduction of probability matrix size If the target service area contains many sensing locations, the probability matrix $\boldsymbol{P}$ will be large, resulting in a serious

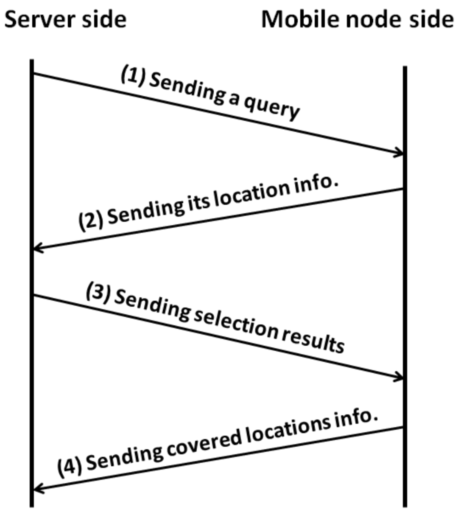

Server side
Mobile node side
(1) Sending a query
(2) Sending its location info.
(3) Sending selection results
(4) Sending covered locations info.

Fig. 29: Query execution sequence between the server and mobile nodes

computational overhead in the server s. However, we only select nodes in the specified AoI and thus we do not need to consider the nodes which move more than $T/2$ away from the border of the AoI since such nodes never come inside the AoI again. This fact allows us to reduce the size of the probability matrix from $N \times N$ to $(M+L) \times (M+L)$, where M is the number of sensing locations included in the AoI and L is the number of sensing locations outside the AoI such that their shortest distance to the AoI border is at most $T/2$. Here, note that $N \gg M + L$ holds for typical scenarios where AoI and T are reasonably small and N is large.

Let $V_{in}(\subseteq V)$ denote a set of vertices included in the AoI. Let $V_{out}(= V - V_{in})$ denote the set of vertices outside the AoI, but in the service area. Let $distance(x, y)$ denote the shortest distance from vertex x to vertex y on G. Let $V_{out}^{T/2}$ denote the set of vertices in V_{out} such that the shortest distance to at least one vertex of V_{in} is at most $T/2$. $V_{out}^{T/2}$ is defined by the following equation.

$$V_{out}^{T/2} = \{x \mid x \in V_{out} \wedge \exists y, distance(x, y) \leq T/2 \wedge y \in V_{in}\} \tag{27}$$

The vertices that belong to $V_{out}^{T/2}$ are illustrated in Fig. 28 (b).

Algorithm 4 *The Inter-location based algorithm (ILB)*

```
Input: $U$, $AoI$, $\alpha$, $T$, $G = (V, E)$
Output: $U'$
$U' \leftarrow \emptyset$
Compose $V_{in}$, $V_{out}^{T/2}$, $U_0$ from $AoI$ and $U$
$\boldsymbol{P} \leftarrow$ ComputeProbMatrix($AoI$,$V_{in} \cup V_{out}^{T/2}$)
for $\forall u \in U_0$ do
    Compose $u$'s initial state vector $\boldsymbol{v_0^u}$
end for
$d_{max} \leftarrow \max\limits_{u,u' \in U_0} \{d_{u,u'}\}$
$d_{th} \leftarrow \min(\frac{T}{\alpha \cdot d_{max}}, d_{max})$
while $SetProb(v, U', T) < \alpha, \forall v \in V_{in}$ do
    if $U_0 = \emptyset$ then
        return $\emptyset$
    end if
    Select $u \in U_0$ at random
    if $U' = \emptyset$ or $\min\limits_{u' \in U'} \{d_{u,u'}\} \geq d_{th}$ then
        $U' \leftarrow U' \cup \{u\}, U_0 \leftarrow U_0 - \{u\}$
    end if
end while
return $U'$
```

We can calculate the coverage probability of all vertices in V_{in} taking into account only the node moving probability at each vertex of $V_{in} \cup V_{out}^{T/2}$. Consequently, we define the new probability matrix $\boldsymbol{P'}$ for vertices of $V_{in} \cup V_{out}^{T/2}$.

We define the i-th row and j-th column element $p'_{i,j}$ of $\boldsymbol{P'}$ by the following equation.

$$p'_{i,j} = \begin{cases} p_{i,j} & (x_i, x_j \in V_{in} \cup V_{out}^{T/2} \wedge i \neq j) \\ \\ \sum\limits_{x_k \in Ngh(i)} p_{i,k} & (x_i \in V_{out}^{T/2} - V_{out}^{T/2-1} \wedge i = j) \end{cases} \tag{28}$$

Here, $Ngh(i)$ is the set of neighboring vertices outside $V_{out}^{T/2}$ and $p_{i,j}$ is the probability of the corresponding edge in the original matrix $\boldsymbol{P}$. Equation (28) represents that the moving probability from x_i to x_j is the same as the original matrix $\boldsymbol{P}$ if i is not equal to j. In addition, knowing that nodes once going outside $V_{out}^{T/2}$ cannot go inside the AoI in T, we for convenience set the probability of a node staying at the same location x_i at border of $V_{out}^{T/2}$ to the sum of probabilities of outgoing edges to outside $V_{out}^{T/2}$.

3.4.2 Algorithms

3.4.2.1 The Inter-Location Based Algorithm (ILB) The ILB uses the distance between nodes as a metric to select a set of mobile nodes. Intuitively, the more distant these nodes are, the more likely it is for these nodes to visit distinct sensing locations of AoI. We denote the distance between the initial locations of nodes u and u' in U_0 by $d_{u,u'}$ which is determined as the length of the shortest path between v_0^u and $v_0^{u'}$ on G. The ILB selects a minimal set of mobile nodes $U'(\subseteq U_0)$ such that the distance between any pair of nodes u and u' in U' is equal to or larger than a threshold d_{th}, and the specified AoI is (α, T)-covered. The above statement is defined as follows.

$$\textbf{minimize } |U'| \textbf{ subject to } (30) - (31) \tag{29}$$

$$d_{u,u'} \geq d_{th}, \forall u, u' \in U' \tag{30}$$

$$\text{AoI is } (\alpha,\, T)\text{-}\textit{covered} \tag{31}$$

The value of d_{th} should be dependent on three parameters: the total number of time steps T, the required coverage ratio α, and the maximum distance d_{max} that is the largest distance between the initial locations of two nodes in U_0. Intuitively, as T increases and/or α decreases, the number of selected nodes should decrease. On the contrary, as T decreases and/or α increases, the number of selected nodes must be increased to meet the (α, T)-coverage constraint. To reflect the above relationship among parameters, we define the distance threshold d_{th} by the following equation.

$$d_{th} = \min(\frac{T}{\alpha \cdot d_{max}}, d_{max}) \tag{32}$$

Algorithm 4 shows the node selection process of ILB. The input parameters are the set of mobile nodes U, the area of interest AoI, the required coverage ratio α, the query interval time T, and the service area graph $G = (V, E)$. In line 1, the algorithm initializes U' to be empty. In line 2, it composes the sets of vertices V_{in} and $V_{out}^{T/2}$, and the set of nodes in the AoI, U_0. In line 3, it composes the probability matrix $\boldsymbol{P}$. In lines 4 to 6, it composes the initial state vector for each node $u \in U_0$. In lines 7 and 8, the algorithm determines the maximum distance d_{max} between nodes in U_0 and the distance threshold d_{th}, as defined in equation (32). In lines 9 to 18, the algorithm selects a set of nodes U' as follows: (i) while the AoI is not (α, T)-covered, the algorithm checks the state of U_0 and if U_0 is empty, the algorithm returns $\emptyset$ (i.e., the current U_0 is not sufficient to satisfy the required coverage α), as shown in lines 9 to 12, (ii) the algorithm selects a node $u \in U_0$ at random, as shown in line 13; and (iii) it adds the node u to the selected set of nodes U' if U' is empty or the distance between u and each node $u' \in U'$ is no less than the threshold d_{th}, as shown in lines 14 to 16. Finally, in line 18, the algorithm returns the selected set of nodes U'.

3.4.2.2 The Inter-Meeting Time Based Algorithm (IMTB) The ILB algorithm is based on the distance between nodes. Hence, the selection process is location-dependent and does not take the query interval time T into consideration. To make the node selection more efficiently taking into account the value of T, we propose an inter-meeting time based (IMTB) algorithm which uses the expected first meeting time between nodes as a metric. This meeting time metric reflects the probability of nodes visiting distinct sensing locations of the AoI and describes the expected first meeting time of any pair of nodes $u, u' \in U_0$. Intuitively, as the meeting time between nodes increases, the probability of visiting distinct sensing locations also increases[3] because those nodes explore different locations until they meet for the first time. We denote the expected first meeting time between nodes u and u' in U_0 by $mt_{u,u'}$. The IMTB algorithm selects a minimal set of nodes $U'(\subseteq U_0)$ such that the meeting time $mt_{u,u'}$ between any pair of nodes u and u' in U' is no less than a meeting time threshold mt_{th}, and the specified AoI is (α, T)-covered. The above statement is defined as follows.

[3] This is not the case if the probability of a node staying at the same location is high, but we suppose the environment where most of the nodes near the AoI are likely to move directly to their destinations.

$$\textbf{minimize } |U'| \textbf{ subject to } (34) - (35) \tag{33}$$

$$mt_{u,u'} \geq mt_{th}, \forall u, u' \in U' \tag{34}$$

$$\text{AoI is } (\alpha, T)\text{-}\textit{covered} \tag{35}$$

The values of $mt_{u,u'}$ and mt_{th} are calculated as follows.

The expected first meeting time $mt_{u,u'}$ represents the earliest time when two nodes u and u' in U_0 may meet at some location $v_t \in V_{in}$ and is defined by the following equation.

$$mt_{u,u'} = \begin{cases} \min\limits_{t \in MT_{u,u'}} \{t\} & (MT_{u,u'} \neq \emptyset) \\ T & (MT_{u,u'} = \emptyset) \end{cases} \tag{36}$$

where $MT_{u,u'}$ is a set of possible meeting time between u and u' during the time period T and is defined by the following equation.

$$MT_{u,u'} = \{t \mid Prob(u, t, v_0^u, v_t) > 0 \wedge \ Prob(u', t, v_0^{u'}, v_t) > 0, \ 0 \leq t \leq T, \\ \exists \ v_t \in AoI\} \tag{37}$$

The meeting time threshold mt_{th} should be dependent on three parameters: the total number of time steps T, the required coverage ratio α, and the maximum expected first meeting time mt_{max} between pairs of nodes in U_0. Intuitively, as T increases and/or α decreases, the number of selected nodes will decrease. To reflect the above relationship among parameters, we define the meeting time threshold mt_{th} as follows.

$$mt_{th} = \min(\frac{T}{\alpha \cdot mt_{max}}, mt_{max}) \tag{38}$$

Algorithm 5 shows the node selection process of IMTB. The input parameters are the same as in Algorithm 4. In lines 1 to 6, the algorithm does the same steps as lines 1 to 6 in Algorithm 4. In lines 7 and 8, the algorithm determines the maximum expected first meeting time mt_{max} between nodes in U_0 and the threshold mt_{th}, as defined in equation (38). In lines 9 to 18, the algorithm selects a set of nodes U' as in Algorithm 4, except in line 14, where it adds the node u to the selected set of nodes U' if U' is empty or the expected first meeting time between u and each node $u' \in U'$ is no less than the threshold mt_{th}.

Algorithm 5 *The Iner-meeting time based algorithm (IMTB)*

Input: U, AoI, α, T, $G = (V, E)$

Output: U'

```
 1: $U' \leftarrow \emptyset$
 2: Compose $V_{in}$, $V_{out}^{T/2}$, $U_0$ from $AoI$ and $U$
 3: $\boldsymbol{P} \leftarrow$ ComputeProbMatrix($AoI$,$V_{in} \cup V_{out}^{T/2}$)
 4: for $\forall u \in U_0$ do
 5:     compose $u$'s initial state vector $\boldsymbol{v_0^u}$
 6: end for
 7: $mt_{max} \leftarrow \max_{u,u' \in U_0}\{mt_{u,u'} : mt_{u,u'} \neq \infty\}$
 8: $mt_{th} \leftarrow \min(\frac{T}{\alpha \cdot mt_{max}}, mt_{max})$
 9: while $SetProb(v, U', T) < \alpha, \forall v \in V_{in}$ do
10:     if $U_0 = \emptyset$ then
11:         return $\emptyset$
12:     end if
13:     Select $u \in U_0$ at random
14:     if $U' = \emptyset$ or $\min\limits_{u' \in U'}\{mt_{u,u'}\} \geq mt_{th}$ then
15:         $U' \leftarrow U' \cup \{u\}, U_0 \leftarrow U_0 - \{u\}$
16:     end if
17: end while
18: return $U'$
```

3.4.2.3 The Extended Algorithm without Thresholds (EWOT) As we described in the previous two subsections, the ILB and IMTB algorithms apply the selection process only on a set of nodes located inside AoI at time 0, U_0, and do not consider the nodes outside the AoI. The number of nodes inside the AoI at time 0 may not be sufficient to guarantee the α-coverage of the AoI in time period T, if it is too small.

Algorithm 6 *The Extended Algorithm without Thresholds (EWOT)*

Input: U, AoI, α, T, $G = (V, E)$

Output: U'

```
$U' \leftarrow \emptyset$
Compose $V_{in}$, $V_{out}^{T/2}$, $U_0$, $U_0^{T/2}$ from $AoI$ and $U$
$\boldsymbol{P} \leftarrow$ ComputeProbMatrix($AoI$,$V_{in} \cup V_{out}^{T/2}$)
for $\forall u \in U_0 \cup U_0^{T/2}$ do
    compose $u$'s initial state vector $\boldsymbol{v_0^u}$
end for
while $SetProb(v, U', T) < \alpha, \forall v \in V_{in}$ do
    if $U_0 \neq \emptyset$ then
        Select $u$ with the highest coverage contribution of $U_0$
        $U' \leftarrow U' \cup \{u\}, U_0 \leftarrow U_0 - \{u\}$
    else if $U_0 = \emptyset$ then
        if $U_0^{T/2} = \emptyset$ then
            return $\emptyset$
        end if
        Select $u$ with the highest coverage contribution of $U_0^{T/2}$
        $U' \leftarrow U' \cup \{u\}, U_0^{T/2} \leftarrow U_0^{T/2} - \{u\}$
    end if
end while
return $U'$
```

To cope with this situation, we extend the algorithms to add more nodes located outside the AoI in the selection process based on their contributions to the coverage of the AoI. Here, the contribution of a node means the expected number of locations in the AoI visited by the node during the time period T. The contribution of a node located outside the AoI should be dependent on its initial location and the time period T. In other words, it should be dependent on the shortest distance from the added node to the AoI. Intuitively, if this distance of a new added node is more than T, then the node will not visit any locations in the AoI within the time period T. So, the distance must be less than or equal to T. In order to avoid a very large number of added nodes, we only add nodes if the shortest distance to the AoI is less than or equal to $\lfloor \frac{T}{2} \rfloor$. We denote the extended algorithm without thresholds by *EWOT*.

Algorithm 6 shows the node selection process of EWOT. The input parameters are the same as in Algorithms 4 and 5. In line 1, the algorithm initializes U' to empty. In line 2, it composes the sets of vertices V_{in} and $V_{out}^{T/2}$, and the sets of nodes U_0 and $U_0^{T/2}$ (this contains all nodes that initially exist in $V_{out}^{T/2}$). In line 3, it composes the probability matrix $\boldsymbol{P}$. In lines 4 to 6, it composes the initial state vector for each node $u \in U_0 \cup U_0^{T/2}$. In lines 7 to 19, the algorithm selects a set of nodes U' as follows: (i) while the AoI is not (α, T)-covered, if U_0 is not empty, the algorithm selects a node u with the highest coverage contribution of U_0 and adds it to the selected set of nodes U', as shown in lines 7 to 10. (ii) if U_0 is empty (i.e., the current U_0 is not sufficient to satisfy the required coverage α), the algorithm checks the state of $U_0^{T/2}$, and if it is empty, the algorithm returns $\emptyset$, as shown in lines 11 to 14, (iii) the algorithm selects a node u with the highest coverage contribution of $U_0^{T/2}$, as shown in line 15; (iv) it adds the node u to the selected set of nodes U', as shown in line 16. Finally, in line 19, the algorithm returns the selected set of nodes U'.

Fig. 29 shows the execution sequence process of a query for the proposed algorithms between the server and the mobile nodes. The steps of this execution sequence process as follows: (1) the server sends the query to all mobile nodes that exist in the service area, (2) each mobile node sends its location information to the server, (3) the server performs the selection algorithm steps (ILB, IMTB, or EWOT) and notifies the selected nodes to start their sensing job, and (4) finally, at the end of the query time interval, all selected mobile nodes send the sensed data of their covered locations to the server.

3.4.2.4 The ILB and IMTB with Updating Mechanism As we described in the previous two subsections, the ILB and IMTB algorithms are based only on the initial locations of nodes inside the AoI and do not consider the latest locations of nodes during the query period T. There can be a scenario that some nodes initially exist in the AoI and may go out of the AoI after some time during the query period T. Also, some nodes may initially exist outside of the AoI and may go into the AoI after some time during the query period T. If we track the location of nodes in and near the AoI during period T, we can achieve more accurate coverage with lower cost by removing useless nodes and adding some extra nodes that contribute more coverage. For more accurate coverage, we propose an updating mechanism for the ILB and IMTB algorithms that aims to adapt the number of selected nodes based on the latest location of nodes. This updating mechanism is executed every specified time interval during the time period T.

Let $t_{current}$ denote the current time step. The updating mechanism consists of the following steps

1. Calculate the remaining required coverage ratio, β[4] ($\beta = \alpha - \gamma$, where γ is the already achieved coverage ratio).

2. Estimate the coverage probability for all uncovered locations in the AoI by the nodes in U' if their current locations exist in the AoI by using the ILB or IMTB algorithms.

3. If the estimated coverage probability is less than β, then one-by-one add a new node to the selected set while all locations in the AoI are (β, $T - t_{current}$)-covered.

4. If the estimated coverage probability is larger than β, then one-by-one remove a node from U' as long as all uncovered locations in AoI are (β, $T - t_{current}$)-covered.

By using this updating mechanism, the ILB and IMTB algorithms can adapt the number of selected nodes by adding or removing some nodes to improve the accuracy of the coverage probability of the AoI as much as possible. This updating mechanism is executed periodically every specified time interval which is called the *updating interval* UI.

It is preferable to determine the value of UI internally. In other words, it should be dependent on T and α. So, we use the distance threshold d_{th} and the meeting time threshold mt_{th} of ILB, and IMTB, respectively to determine the value of UI as follows.

$$UI = \begin{cases} d_{th} \quad for \quad ILB \\ mt_{th} \quad for \quad IMTB \end{cases} \tag{39}$$

We refer to the ILB and IMTB with the updating mechanism by *ILB-up* and *IMTB-up*, respectively.

[4] Here, to minimize the total overhead, β represents the maximum deficit coverage ratio among all locations in AoI.

3.4.2.5 Complexity Here, we evaluate the computing time of the proposed algorithms according to the size of matrix $\boldsymbol{P}$, $(M + L)^2$, the number of nodes in U_0 (i.e., inside AoI), n, and the total number of steps, T. According to the coverage probability of a vertex which is defined in equation (26), the computing time of ILB and IMTB is $\boldsymbol{O}\ (n \cdot (M+L)^2 \cdot T)$, while the computing time for EWOT is $\boldsymbol{O}((n+n') \cdot (M+L)^2 \cdot T)$, where n' is the number of nodes in $U_0^{T/2}$ (i.e., outside AoI).

3.5 Performance Evaluation

In this section, we show the results of simulation experiments that examine the coverage performance of the proposed algorithms in terms of the number of nodes selected and the accuracy of the achieved coverage ratio. We compared the proposed algorithms with the random selection method which repeats selecting a node randomly among all nodes inside the AoI until satisfying (α, T)-coverage and does not use any distance or time thresholds.

3.5.1 Simulation Environment

The QualNet [33] simulator was used with the input parameters listed in Table 7, such as service area size, number of nodes, node speed, etc. In addition, the node mobility was based on a discrete Markov model as described in Section 3.4.2. The service area was represented as a grid of sensing locations arranged with uniform spacing, 50 *meters*. We selected the AoI as a rectangular region where its position was selected at random within the service area. The ratio of its size to the service area size, called *AoI-Size*, was selected from {0.01, 0.25, 0.45, 0.5, 0.65, 0.85} and the corresponding number of sensing locations in each AoI was {4, 36, 56, 66, 77, 99}. The initial node location was selected at random among all sensing locations in the service area. We repeated every simulation experiment 5 times with different initial node distributions, then averaged the results.

We measured the performance of the proposed algorithms in terms of the number of selected nodes and the achieved coverage ratio by changing the number of nodes, the AoI-Size, the total number of time steps (query interval time), and the required coverage ratio. Here, we define the achieved coverage ratio as the ratio of the number of sensing locations visited in the AoI by at least one node to the total number of sensing locations in the AoI. We say that the algorithms satisfy the required coverage ratio if the average achieved coverage ratio of several simulation runs is no less than the required ratio.

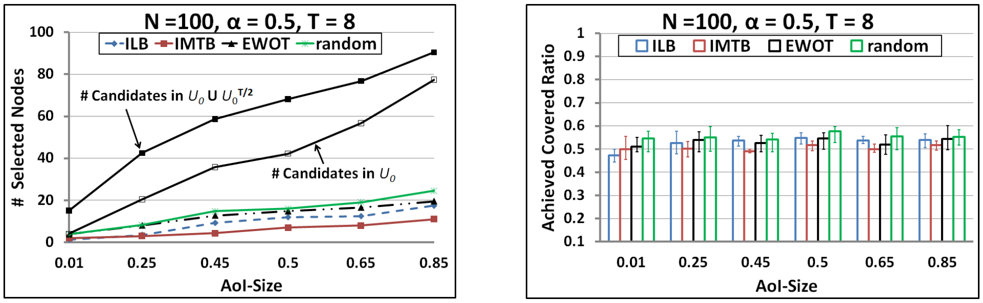
Table 7: Configuration Parameters for PCS.

Configuration parameter	**Value in simulation**
# nodes	25 to 200
Node speed	1 m/s
Field size	$500m \times 500m$
Required coverage, α	0.2, 0.4, 0.5, 0.6, 0.8, 0.9
Total # sensing locations in A	121
AoI-Size (# sensing locations)	0.01 (4), 0.25 (36), 0.45 (56), 0.5 (66), 0.65 (77), 0.85 (99)
Δ	50 m
Total # steps (time period), T	2, 4, 6,..., 20

3.5.2 Simulation Results without Updating Mechanism

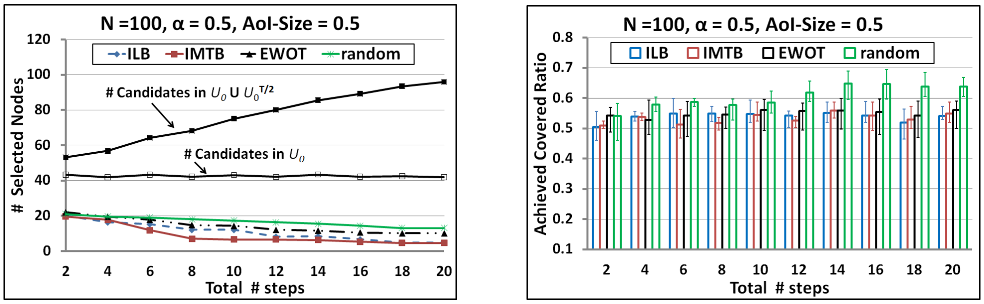
In this section we show the simulation results for the proposed algorithms without the updating mechanism in two cases. In the first case, the moving probabilities of a node at a location to its neighboring locations were equal probabilities (i.e., uniform and equal to 0.25). To show the performance of the proposed algorithms under non-uniform moving probabilities, in the second case, the moving probabilities of a node at a location to its neighboring locations were unequal probabilities. We show simulation results in Figs. 30-36 (The black lines with empty and solid rectangles in Figs. 30-36(a) represent the number of candidate nodes in U_0 and $U_0 \cup U_0^{T/2}$, respectively).

3.5.2.1 Equal Moving Probabilities Fig. 30 shows the performance for different AoI-Size in the case of medium required coverage ratio and medium number of time steps. The number of nodes was 100, the required coverage α was 0.5, and the total number of steps was 8. In Fig. 30 (a), the number of selected nodes increased as the AoI-Size increased. This is because, when the AoI-Size increased, we needed more nodes to satisfy the required coverage ratio. As shown in Fig. 30 (a), the number of



(a) Number of selected nodes (b) Achieved coverage ratio

Fig. 30: Performance for different AoI sizes

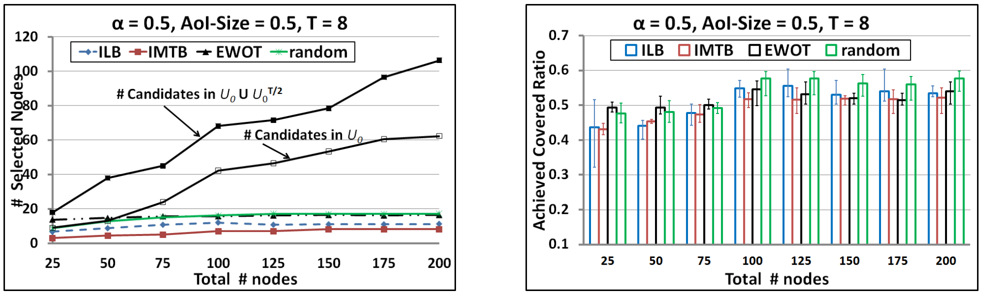


(a) Number of selected nodes (b) Achieved coverage ratio

Fig. 31: Performance for different time steps

selected nodes for the proposed algorithms was much smaller than the number of candidates nodes in the AoI and it was reduced by 75.46%, 79.77%, and 57.06% for ILB, IMTB, and EWOT, respectively, while for the random algorithm, its reduction was 53.62%. In Fig. 30 (b), the required coverage was almost satisfied by all algorithms. When the AoI-Size was 0.01, the number of selected nodes and the variance of ILB was smaller than other algorithms. For a larger AoI-Size, the number of selected nodes and the variance of IMTB were smaller than other algorithms. As a result, for a smaller values of the AoI-Size, the ILB is the best, while the IMTB is the best in the case of a larger AoI-Size.

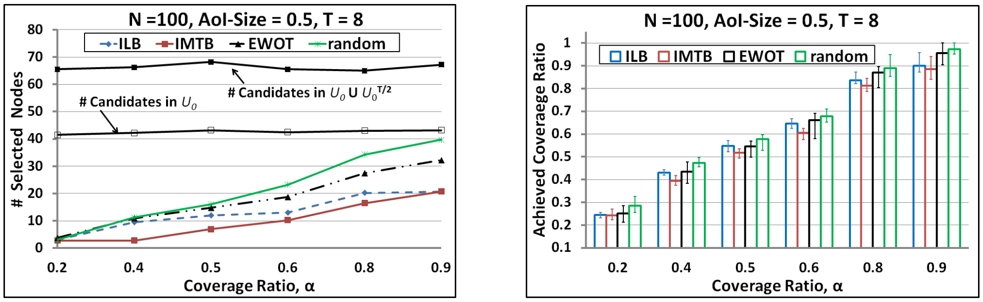
Fig. 31 shows the performance for different numbers of time steps with a medium size AoI and a medium required coverage ratio. The AoI-Size was 0.5. In Fig. 31 (a), the number of selected nodes decreased as the total number of steps increased. This is because the distance and meeting time threshold increases in proportion to the total number of steps. The number of selected nodes for IMTB was lower than other algorithms since the required coverage is medium and the meeting time increased when



(a) Number of selected nodes

(b) Achieved coverage ratio

Fig. 32: Performance for different number of nodes



(a) Number of selected nodes

(b) Achieved coverage ratio

Fig. 33: Performance for different required coverage ratios

the AoI-Size is medium. The number of selected nodes was reduced by 74.55%, 79%, and 66.6% of the number of candidates nodes in the AoI for ILB, IMTB, and EWOT, respectively, while for random algorithm, its reduction was 60.83%. In Fig. 31 (b), all algorithms satisfied the required coverage and the variance of IMTB was smaller than other algorithms.

Fig. 32 shows the performance for different numbers of nodes with a medium size AoI, a medium required coverage ratio, and a medium number of time steps. The AoI-Size was 0.5. In Fig. 32 (a), when the number of nodes was 25 to 125, the number of selected nodes increased as the number of nodes increased. This is because, when the number of nodes increases, the algorithms add more nodes to satisfy the required coverage. However, when the number of nodes was larger than 125, the number of selected nodes was fixed since the number of selected nodes is bound by the number of nodes needed to satisfy the required coverage. Also, when the number of nodes was 25 to 75, the number of selected nodes was reduced by 49.93% of the number of candidate nodes inside and outside the AoI for EWOT. For a larger number of nodes, it was reduced by 78.35%, 85.23%, 68.74%, and 67.38% of the number of candidate nodes in the AoI for ILB, IMTB, EWOT, and random algorithms, respectively. In Fig. 32 (b), the required coverage was not satisfied by ILB, IMTB, and random algorithms when the total number of nodes was 25 to 75, while the EWOT algorithm satisfied the required coverage with small variance. This is because the EWOT algorithm takes into account nodes that also exist outside the AoI and it can add more nodes to meet the required coverage. For a larger number of nodes, all algorithms satisfied the required coverage and the variance of IMTB was smaller than other algorithms. As a result, for a small number of nodes inside AoI, the EWOT is the best, while the IMTB is the best in the case of a larger number of nodes.

Fig. 33 shows the performance for different required coverage ratio with a medium size AoI and a medium number of time steps. The AoI-Size was 0.5. In Fig. 33 (a), the number of selected nodes increased as the required coverage ratio increased. This is because, as the required coverage ratio increases, we need more nodes to satisfy it. The number of selected nodes for IMTB was lower than other algorithms and it was reduced by 76.68% of the number candidate nodes in the AoI. On the other hand, it was reduced by 69.46%, 58%, and 50.45% for ILB, EWOT, and random algorithms, respectively. In Fig. 33 (b), the required coverage was satisfied by all algorithms and the variance of IMTB is smaller than other algorithms.

Here, we summarize the simulation results as follows.

- ILB, IMTB, and EWOT algorithms reduce the number of selected nodes to a great extent for (α, T)-coverage compared to the number of candidate nodes in the AoI.
- For a small AoI, ILB can select a smaller number of nodes to meet the required coverage with a smaller variance than IMTB, EWOT, and random algorithms.
- For medium and large AoI, IMTB can select a smaller number of nodes to meet the required coverage with a smaller variance than ILB, EWOT, and random algorithms.
- When only a small number of nodes are initially located in the AoI, only the EWOT algorithm can meet the required coverage.

3.5.2.2 Unequal Moving Probabilities In the real environment, the moving probabilities of a node at any location to its neighboring one are almost unequal. In order to investigate to what extent the unequalness of the moving probability affects the performance of the proposed methods, we conducted simulations according to the following two scenarios.

- **a) random moving probabilities:** in this scenario, the moving probability of a node at a location i to one of its neighboring locations is determined randomly between 0.01 and 0.09 such that the sum of all moving probabilities to its neighboring locations is equal to 1.
- **b) biased moving probability p:** in this scenario, we constructed a model by defining a moving probability parameter p as shown in Fig.34. In the simulations, the value of p was selected from {0.05, 0.1, 0.15, 0.2, 0.25}. Based on this model, when p is small, most of the nodes are likely to move towards a specific direction with higher probabilities (e.g., towards bottom right corner). Here, $p = 0.25$ corresponds to the case of equal moving probability.

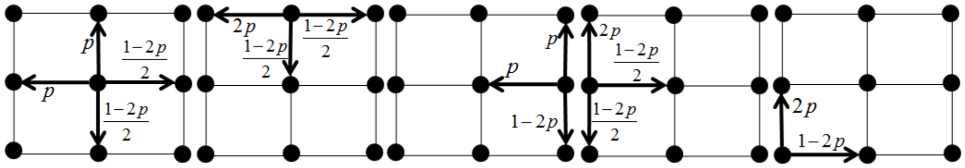
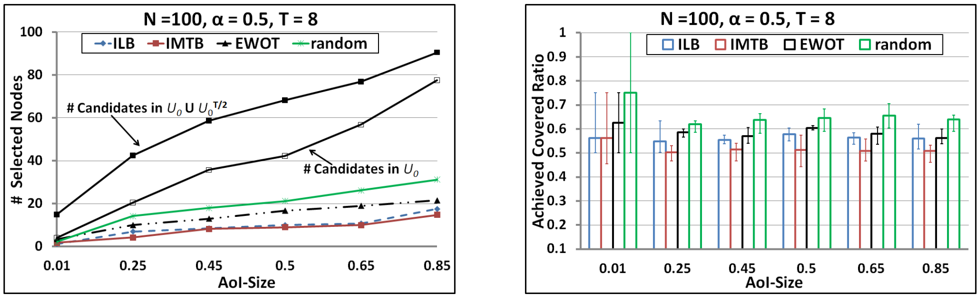


Fig. 34: Moving probabilities cases for probability p

Fig. 35 shows the performance for a different AoI-Size by using random moving probabilities. The number of nodes was 100, the required coverage α was 0.5, and the total number of steps was 8. In Fig. 35 (a), the trend on the number of selected nodes was almost similar to the case of Fig. 30 (a), but more nodes were selected. This is because, in this scenario, the probability matrix is not uniform and there are a smaller number of nodes that visit some sensing locations in AoI. In Fig. 35 (b), the required coverage is almost satisfied by all algorithms.

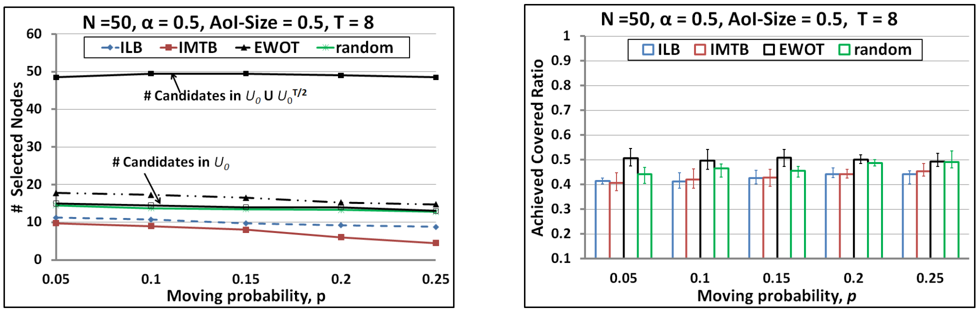
Fig. 36 shows the performance for different values of p. Where a sufficient number of nodes exists in the AoI, a clear impact of p value may not occur on the AoI coverage. So, it is preferable to evaluate the performance of the proposed algorithms when there



(a) Number of selected nodes

(b) Achieved coverage ratio

Fig. 35: Performance for different AoI sizes with random moving probability



(a) Number of selected nodes

(b) Achieved coverage ratio

Fig. 36: Performance for different moving probabilities, p

is an insufficient number of nodes inside the AoI. Therefore, in this simulation, the number of nodes was 50. In Fig. 36 (a), the number of selected nodes decreased as p increased. This is because, when p increases, the expected number of different visited locations for each node increases. In Fig. 36 (b), the required coverage was satisfied only by EWOT. This is because, there is an insufficient number of nodes inside the AoI. Also, EWOT reduced the number of selected nodes by 66.74% of the number of candidate nodes inside and outside the AoI. The variance of ILB, IMTB, and random algorithms decreased as p increased. This is because, when p increases the nodes tend to move in different directions and the expected number of different covered locations increases. As a result, if there is an insufficient number of nodes inside the AoI and most of the nodes tend to move towards a specific direction, the EWOT algorithm is the best among all algorithms.

3.5.2.3 Traffic Overhead and Resource Consumption Ratio In this section, we show the traffic overhead and the efficiency in mobile nodes' resources consumption of the proposed algorithms.

The main objectives of the proposed node selection algorithms are (i) achieving AoI coverage ratio α in time period T and (ii) minimizing the overall cost consisting of (1) the incentive fees (determined depending on resource consumption of mobile nodes for sensing and uploading) paid to the nodes that perform sensing and uploading and (2) the total traffic amount.

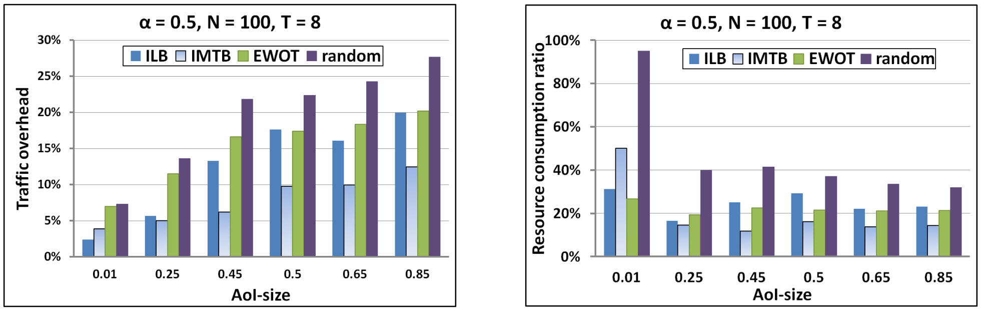
Here, we defined the method in which the node selection is not performed and all candidate nodes are performing the sensing and uploading operations as the standard method. According to the query execution sequence in Fig. 29, to execute the standard method, only two processes (1) and (4) are needed. On the other hand, to execute the proposed algorithms, the four processes (1), (2), (3), and (4) are needed. Here, the traffic overhead of the proposed algorithms depends on the number of mobile nodes in each process of the query execution sequence. Now, we will explain how to measure the traffic overhead as follows.

Let N denote the total number of mobile nodes in service area A, let B denote the number of all candidates in the AoI, and let C denote the number of selected candidates (for the proposed algorithms). In the case of the standard method, the number of mobile nodes in (1) and (4) are N and B, respectively. So, the total traffic (in terms of the number of transmitted messages) for the standard method is equal to $N + B$. In the case of the proposed algorithms, the number of mobile nodes in (1), (2), (3), and (4) are N, B, C, and C, respectively. So, the total traffic (in terms of the number of transmitted messages) for the proposed algorithms is equal to $N + B + 2C$. We will define the traffic overhead for the proposed algorithms as follows.

$$traffic\,overhead = \frac{2C}{N+B} \tag{40}$$

Also, to show the efficiency in mobile nodes' resource consumption of the proposed algorithms compared to the standard method, we defined the resource consumption ratio of the number of selected candidates to the total number of candidates existing in AoI as follows.

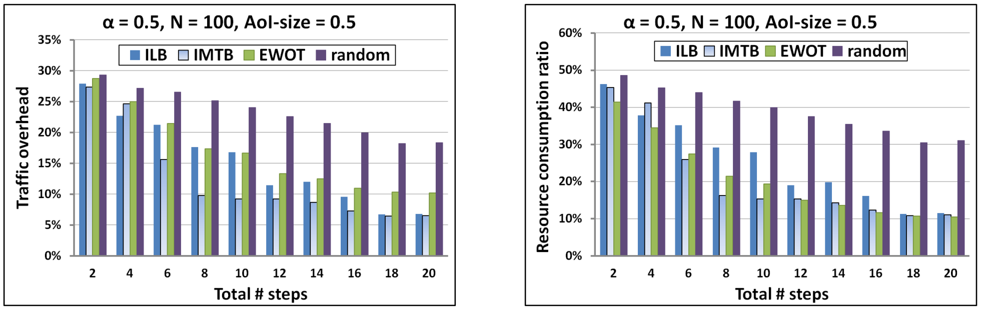
$$Resource\ consumption\ ratio = \frac{C}{B} \tag{41}$$



(a) Traffic overhead

(b) Resource consumption ratio

Fig. 37: Traffic overhead and resource consumption ratio for different AoI sizes



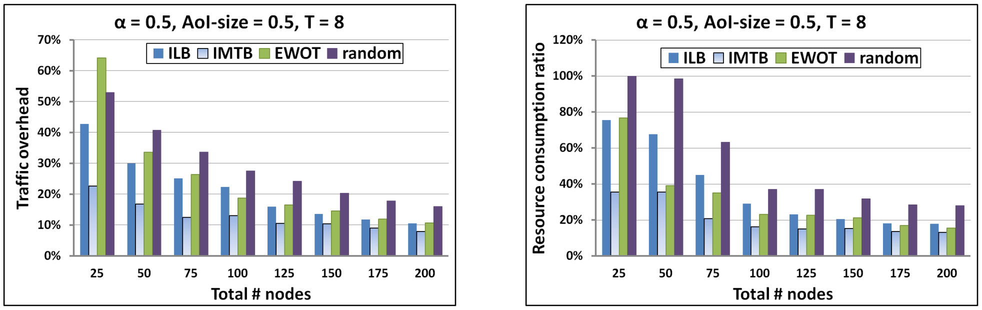
(a) Traffic overhead

(b) Resource consumption ratio

Fig. 38: Traffic overhead and resource consumption ratio for different values of steps

According to equation (40) and equation (41), we measured the traffic overhead and the resource consumption ratio for ILB, IMTB, EWOT, and random algorithms compared to the standard method for different values of AoI size, number of steps, number of nodes, and required coverage in the case of the equal moving probabilities scenario.

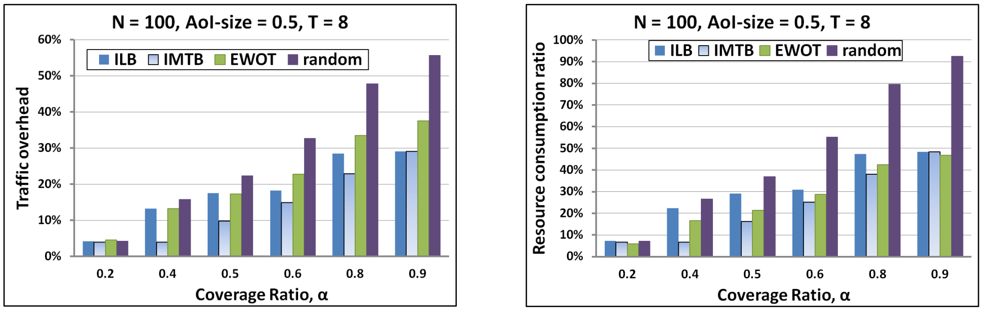
Fig. 37, shows the traffic overhead and the resource consumption ratio for a different AoI-Size. The number of nodes was 100, the required coverage α was 0.5, and the total number of steps was 8. In Fig. 37 (a), the traffic overhead increased as the AoI-Size increased. When the AoI-Size was 0.01, the traffic overhead of ILB was lower than other algorithms and it was 2%. For a larger AoI-Size, the traffic overhead of IMTB was lower than other algorithms and it increased from 5% to 12.4% as the AoI size increases. In Fig. 37 (b), the resource consumption ratio of the proposed algorithms was much lower than the standard method. When the AoI-Size was 0.01, the resource consumption ratio of ILB and EWOT was lower than other algorithms and it



(a) Traffic overhead

(b) Resource consumption ratio

Fig. 39: Traffic overhead and resource consumption ratio for different # of nodes



(a) Traffic overhead

(b) Resource consumption ratio

Fig. 40: Traffic overhead and resource consumption ratio for different values of required coverage, α

was 31% and 27%, respectively. For a larger AoI-Size, the resource consumption ratio of IMTB was lower than other algorithms and it was between 12% to 16%.

Fig. 38 shows the traffic overhead and the resource consumption ratio for a different numbers of time steps. The AoI-Size was 0.5. In Fig. 38(a), the traffic overhead decreased as the number of steps increased. The traffic overhead of IMTB was lower than other algorithms and it decreased from 27% to 6% as the number of steps increases. In Fig. 38 (b), the resource consumption ratio of the proposed algorithms was much lower than the resource consumption ratio in the case of the standard method. The resource consumption ratio of IMTB was lower than other algorithms in most values of time steps and it was between 11% and 45%.

Fig. 39 shows the traffic overhead and the resource consumption ratio for a different numbers of nodes. In Fig. 39 (a), the traffic overhead decreased as the number of nodes increased. The traffic overhead of IMTB was lower than other algorithms and it decreased from 23% to 8% as the number of nodes increases. In Fig. 39 (b), the resource consumption ratio of the proposed algorithms was lower than the standard method. The resource consumption ratio of IMTB was lower than other algorithms and it was between 13% and 36%.

Fig. 40 shows the the traffic overhead and the resource consumption ratio for a different required coverage ratio. In Fig. 40 (a), the traffic overhead increased as the required coverage ratio increased. The traffic overhead of IMTB was lower than other algorithms and it increased from 4% to 29% as the required coverage ratio increases. In Fig. 40 (b), the resource consumption ratio of the proposed algorithms was lower than the standard method. The resource consumption ratio of IMTB was almost lower than other algorithms and it was between 7% and 48%.

3.5.2.4 Sensitivity of P Matrix In this section, we study the sensitivity of matrix P on the performance of the proposed algorithms by adding a noise parameter called σ to the moving probability matrix P as follows.

$$P = (p_{i,j} \pm \sigma_k), \;\; k = 0, 1, 2, ..., T-1, s.t. \sum_{j=1}^{M+L} (p_{i,j} \pm \sigma_k) = 1, 1 \leq i \leq M+L \quad (42)$$

According to equation (42), we conducted simulation experiments to show the impact of σ on the performance of the proposed algorithms against the AoI-size. This is because the size of matrix P depends on the AoI-size and the impact of σ will be more visible. The value of σ was randomly selected between 0 and 1.

As shown in Fig.41, the performance of all algorithms was affected by the value of σ where the variance of all algorithms was bigger than the variance of all algorithms in the case of Fig. 30 (b). In addition, the IMTB still achieved the lowest variance compared to other algorithms.

3.5.3 Simulation Results with Updating Mechanism

In this section, we show the simulation results which we conducted for ILB-up and IMTB-up. We measured the performance of ILB-up and IMTB-up in terms of the number of selected nodes, the achieved coverage ratio, the total number of sensing times, and the communication overhead. In the simulations, the required coverage α was 0.5 and the AoI-Size was 0.5. In order to evaluate the overhead of the updating

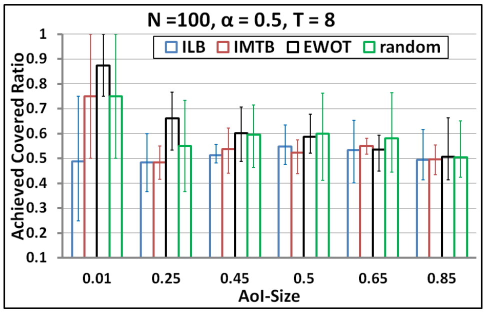


Fig. 41: Effect of σ on the achieved coverage ratio for different AoI size

mechanism, we define the total number of sensing times as the total number of times at which the selected nodes perform a sensing action. It is defined as follows.

$$totalSensingTimes = \sum_{u \in C} nst_u, \quad C = \bigcup_{t \in UT} C_t \tag{43}$$

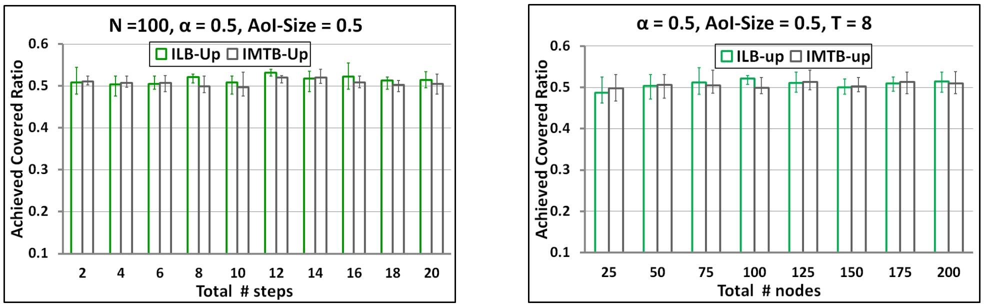
where, nst_u is the number of sensing times of a node u, C is the set of all selected nodes during the time period T, C_t is the set of selected nodes at updating time t (C_0 represents the initial selected set), and UT is the set of updating times.

Also, we define the communication overhead as the total number of candidate nodes for all updating times during the time period. It is defined as follows.

$$ComOverhead = \sum_{t \in UT} candidates(t) \tag{44}$$

where, $candidates(t)$ is the number of candidate nodes at time t. Here, the candidate nodes are the nodes inside the AoI or within distance $(T - t)$ from the AoI border. We show the simulation results in Fig.42 and Fig.44.

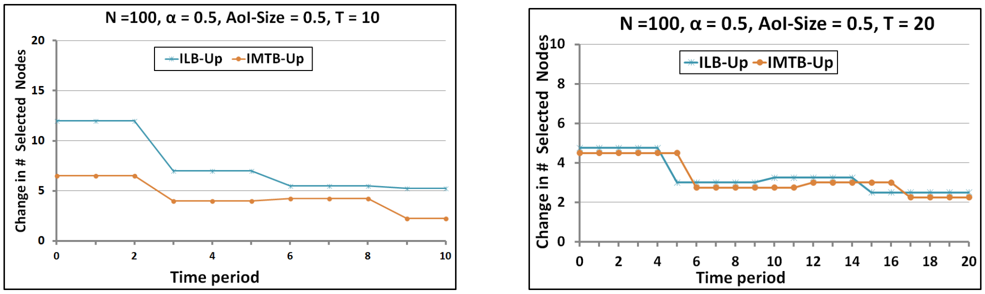
Fig. 42 (a) shows the performance for a different numbers of time steps with a medium size AoI and a medium required coverage ratio. The number of nodes was 100. As shown in Fig. 42 (a), the required coverage was satisfied by ILB-up and IMTB-up. The accuracy of ILB-up and IMTB-up was better than ILB and IMTB in Fig. 31 (b) and their variances were lower than ILB and IMTB. This is because, ILB-up and IMTB-up adapt the number of selected nodes during the time period and ILB and IMTB do not. Fig. 42 (b) shows the performance for a different numbers of nodes with a medium size AoI and a medium required coverage ratio. The number of time steps was 8. While ILB and IMTB did not satisfy the required coverage ratio when the total



(a) Achieved coverage vs. Total # of steps

(b) Achieved coverage vs. Total # of nodes

Fig. 42: Coverage performance for update algorithms



(a) T = 10

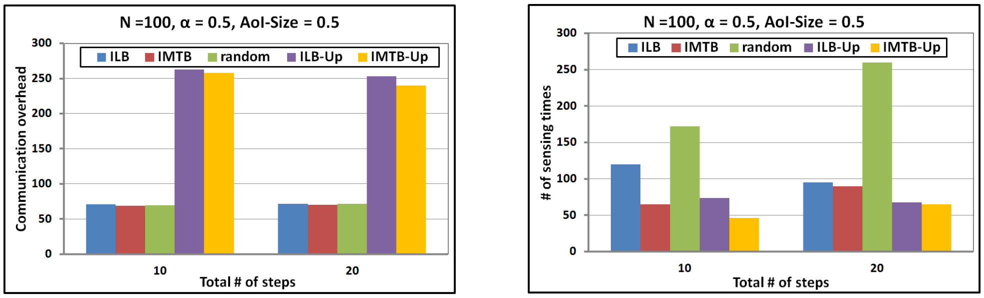
(b) T = 20

Fig. 43: Change in number of selected nodes during time period, T = 10, 20

number of nodes was 25 to 75 (Fig. 32 (b)), ILB-up and IMTB-up satisfied the ratio thanks to the update mechanism. For a larger number of nodes, all algorithms satisfied the required coverage.

Fig. 43 shows the change in the number of selected nodes during a time period when T was 10 and 20, respectively. As shown in Figs. 43 (a) and 43 (b), ILB-up and IMTB-up adapt the number of selected nodes by adding or removing nodes to improve the accuracy as much as possible during the time period.

Figs. 44 (a) and 44 (b) show the communication overhead and the total number of sensing times when the time period was 10 and 20 steps. In Fig. 44 (a), the communication overhead for ILB-up and IMTB-up was larger than ILB, IMTB, and random algorithms since ILB-up and IMTB-up requires all nodes to communicate in and near the AoI at each update time. In Fig. 44 (b), the total number of sensing times for ILB-up and IMTB-up was smaller than ILB, IMTB, and random algorithms since the



(a) Communication overhead when T is 10 and 20

(b) Number of sensing times when T is 10 and 20

Fig. 44: Communication overhead and # of sensing times for update algorithms

update mechanism selects only necessary nodes taking into account the already covered sensing locations at each update time.

In conclusion, the update mechanism can be used for applications that require a high accuracy of AoI coverage and are not concerned with the communication overhead. On the other hand, if low communication overhead is required, it is better to use ILB and IMTB without the update mechanism.

3.5.4 Realistic Scenario Evaluation

To show the performance of the proposed algorithms on a realistic scenario, we conducted a simulation on a specific city map near Osaka station in Japan. Fig. 45 shows the city map with its road network and its sensing locations with a uniform spacing which was 50 meters. Here, the number of nodes was 100, the required coverage α was 0.5, and the AoI-Ratio was 0.5. The rectangle in Fig. 45 represents the selected AoI area. We used MobiREAL simulator [35] to generate a realistic mobility trace for mobile nodes from the actually observed number of nodes on each street. Based on the city map and the generated trace, the moving probability of mobile nodes was determined. We show the simulation results in Fig. 46 (The black line with solid diamonds in Fig. 46 (a) represents the number of candidate nodes in the AoI).

In Fig. 46 (a), the trend on the number of selected nodes is almost similar to the case of Fig. 31, but more nodes were selected when the number of steps was bigger. This is because, the probability matrix in this experiment is not uniform and there is a smaller number of nodes moving towards some sensing locations in AoI. In Fig. 46 (b),

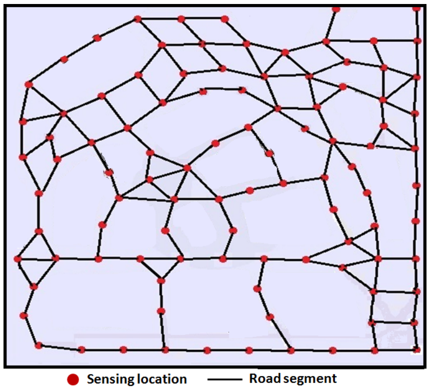
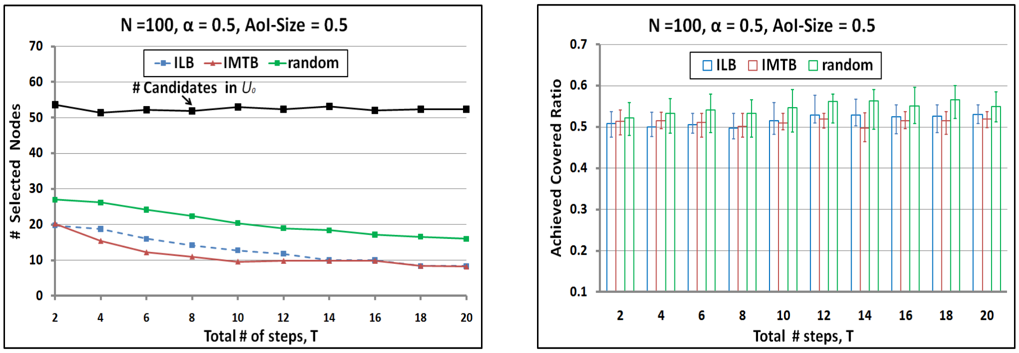


Fig. 45: A city map representing a service area with roads and sensing locations



(a) Number of selected nodes vs. Total # of steps

(b) Achieved coverage Ratio vs. Total # of steps

Fig. 46: Performance in realistic scenario

ILB and IMTB satisfied the required coverage and the variance of IMTB was in most cases smaller than ILB.

3.6 Conclusion for Probabilistic Coverage Methods

In this chapter, we tackled the $(\alpha,\, T)$-coverage problem in people centric sensing with a motivating application scenario. We formulated this problem as an optimization problem with the objective of minimizing the number of selected nodes to meet the demanded coverage ratio α within a query interval time T. To resolve this problem, we proposed heuristic algorithms. Our simulation results showed that the proposed algorithms achieved $(\alpha,\, T)$-coverage with good accuracy for a variety of values of α,

T, AoI size, and moving probability, and that the inter-meeting time based algorithm selects a smaller number of nodes without deteriorating coverage accuracy. Also, the proposed algorithms reduce the cost (number of sensing times) to a great extent compared to the case of selecting all nodes in the AoI. In addition, our updating mechanism adapts the number of selected nodes by removing useless nodes and adding some extra nodes that contribute more to AoI coverage.

4. Conclusion and Future work

In this thesis, focusing in mobile networks, we studied the improvement of information sharing service availability in MANETs and the information gathering in people-centric sensing.

In Chapter 2, two new distributed adaptive service replication methods for MANETs were presented. Our protocols first divide the whole network into disjoint zones with diameters of at most 2 hops, select a node with minimum moving speed in each zone as a zone head, and construct a virtual backbone network connecting all zone heads. By using this zone structure, our protocols select a new replica node according to the topology, number of requests from clients, and the tradeoff between communication an replication costs for each zone. Our protocols are scalable and they can control the locations and the number of service replicas, keeping the network-wide energy consumption as low as possible and improving the service availability. Simulations demonstrated that our method improves the performance of service provision in terms of the energy consumption and service availability compared with existing methods. In addition, the path length between a client and a server is minimized independent of network size.

In Chapter 3, we tackled the (α, T)-coverage problem in people centric sensing with a motivating application scenario. We formulated this problem as an optimization problem with the objective of minimizing the number of selected nodes to meet the demanded coverage ratio α within a query interval time T. To resolve this problem, we proposed heuristic algorithms. Our simulation results showed that the proposed algorithms achieved (α, T)-coverage with good accuracy for a variety of values of α, T, AoI size, and moving probability, and that the inter-meeting time based algorithm selects a smaller number of nodes without deteriorating coverage accuracy. Also, the proposed algorithms reduce the cost (number of sensing times) to a great extent compared to the case of selecting all nodes in the AoI. In addition, our updating mechanism adapts the number of selected nodes by removing useless nodes and adding some extra nodes that contribute more to AoI coverage.

In this thesis, the proposed service replication methods, in Chapter 2, did not consider synchronization and consistency of service replicas, the presence of selfish nodes and its effect on the network performance, and the difference of resources among mobile nodes. In the future work, we can improve our proposed distributed service replication algorithms to solve the synchronization and consistency of service replicas by implementing a zone-tree-based scheme which constructs a virtual updating tree between all head zones of the network and all service data units can be transferred along the updated tree. Also, to solve the presence of selfish nodes, we can use a detection scheme to detect the presence of these selfish or malicious nodes. Also, we will consider the difference of resources among mobile nodes in the replication process by dividing mobile nodes into groups based on their capabilities of resources as storage, battery, and processing power. For example, by dividing mobile nodes into high and low resources groups. In addition, we will implement the proposed method and experiment in the real world to know its performance under realistic conditions. Also, in this thesis, the proposed probabilistic coverage methods, in Chapter 3, consider only the case where a single query is issued at a time. In the future work, we will try to make the proposed algorithms adaptive in case of multiple simultaneous queries to minimize the overhead.

Acknowledgements

First of all, I would like to express my deep gratitude to Professor Minoru Ito for his support and constant encouragement. This work could not be achieved without his support.

I am deeply grateful to Professor Hiroyuki Seki for his useful comments and valuable discussions.

I deeply thank Professor Keiichi Yasumoto for his kind support and suggestions throughout my whole graduate study. I learned a lot from him about academic thinking, technical writing, presentation skills and how to do research. His advices will be very helpful in my future academic activity.

I would also like to thank Associate Professor Naoki Shibata of Shiga University for his support and valuable technical discussions. I learned a lot from him about technical things such as algorithms and programming techniques.

I would also like to thank Assistant Professors Tomoya Kitani of Shizuoka University and Yukiko Yamauchi of Kyushu University for thier support and valuable technical discussions. I learned a lot from them about technical things such as academic thinking and how to do research.

I would also like to thank Assistant Professor Weihua Sun for helping me many times and giving me many advices.

I would also like to thank all current and previous members in Foundation of Software laboratory.

I would also like to thank all members in Mathematic Department in Faculty of Science, Al-Azhar University for their support to finish my study.

I would also like to thank my Egyptian Government for giving me this opportunity to complete my study and get PhD degree in Japan. I deeply thank my government for its financial support.

I would like to thank my parents and my family for providing a lot of advices and instructions that helped me to bear the difficulties and gave me the enthusiasm to complete my studies successfully. I present them my success and I promise them to continue the way of success. I offer my sincere appreciation and respect to them. Thanks to my father, mother, and my wife.

Finally, I would like to thank all people from whom I have received a kindness.

References

[1] L. Kleinrock: "Nomadic Computing: An Opportunity," ACM SIGCOMM Computer Communications Review, Vol. 25, No. 1, pp. 36-40, Jan. 1995.

[2] U. Varshney and R. Vetter: "Emerging Mobile and Wireless Networks," Communications of the ACM, Vol. 43, No. 6, pp. 73-81, Jun. 2000.

[3] R.O. LaMaire, A. Krishna and P. Bhagwat: "Wireless LAN and Mobile Networking: Standards and Future Directions," IEEE Communications Magazine, Vol. 34, No. 8, pp. 86-94, Aug. 1996.

[4] B. Miller: "Satellites Free The Mobile Phone," IEEE Spectrum, Vol. 35, No. 3, pp. 26-35, Mar. 1998.

[5] A.R. Noerpel and Y.B. Lin: "Wireless Local Loop: Architecture, Technologies, and Services," IEEE Personal Communications Magazine, Vol. 5, No. 3, pp. 74-80, Jun. 1998.

[6] C.E. Perkins: "Mobile IP," IEEE Communications Magazine, Vol. 35, No. 5, pp. 84-99, May 1997.

[7] D. Raychaudhuri and N. D. Wilson: "ATM-based Transport Architecture for Multiservices Wireless Personal Communication Networks," IEEE Journal on Selected Areas in Communications, Vol. 12, No. 8, pp. 1401-1414, Oct. 1994.

[8] U. Varshney: "Supporting mobile Computing Using Wireless ATM," IEEE Computer, Vol. 30, No. 1, pp. 131-133, Jan. 1997.

[9] D. P. Agrawal: "Future Directions in Mobile Computing and Networking Systems," Proc. of Workshop sponsored by the NSF, University of Cincinnati, Jun. 1999.

[10] R. Malladi and D. P. Agrawal: "Current and Future Applications of Mobile and Wireless Networks," Communication of ACM, Vol. 45, No. 10, pp. 144-164, Oct. 2002.

[11] L. Levine: "Campus-Wide Mobile Wireless: Mobility and Convergence," Syllabus Oct. 2002.

[12] R. McGhee and R. Kozma: "New Teacher and Student Roles in the Technology-Supported Classroom," the American Educational Research Association Annual Meeting, Oct. 2001.

[13] J. McKenzie: "The Unwired Classroom: Wireless Computers Come of Age," The Educational Technology Journal, Vol. 10, No. 4, Jan. 2001.

[14] S.H. Kim,C. Mims and K.P. Holmes: "An Introduction to Current Crends and Benefits of Mobile Wireless Technology Use in Higher Education," AACE Journal, Vol. 14, No. 1, pp. 77-100, 2006.

[15] F. Maginnis, R. White and C. Mckenna: "Customers on the Move: m-Commerce Demands A Business Object Broker Approach to EAI," eAI Journal, pp. 58-62, Nov. 2000.

[16] "Web Services and Service-Oriented Architectures," *http://www.service-architecture.com/web-services/articles/service-oriented_architecture _soa_definition.html*

[17] S. Dustdar and L. Juszczyk: "Dynamic Replication and Synchronization of Web Services for High Availability in Mobile Ad-hoc Networks," *Service Oriented Computing and Applications*, Vol. 1, No. 1, pp. 19-33, Apr. 2007.

[18] L. Juszczyk: "Replication and Synchronization of Web Services in Ad-hoc Networks," *Masters thesis, Technischen University at Wien*, 2005.

[19] M. Hauspie, D. Simplot, and J. Carle: "Partition Detection in Mobile Ad hoc Networks Using Multiple Disjoint Paths Set," *Proc. of the 2nd Mediterranean Workshop on Ad-Hoc Networks*, Jun. 2003.

[20] K. Wang and B. Li: "Efficient and Guaranteed Service Coverage in Partitionable Mobile Ad-hoc Networks," *Proc. of IEEE INFOCOM'02*, Vol. 1, pp. 1089-1098, Jun. 2002.

[21] B. Li and K. H. Wang: "Nonstop: Continuous Multimedia Streaming in Wireless Ad hoc Networks with Node Mobility," *IEEE Journal on Selected Areas in Communications*, Vol. 21, No. 10, pp. 1627-1641, Dec. 2003.

[22] A. Derhab, N. Badache and A. Bouabdallah: “A Partition Prediction Algorithm for Service Replication in Mobile Ad hoc Networks,” *Proc. of the 2nd Conf. on Wireless On-demand Network Systems and Services*, pp. 236-245, Jan. 2005.

[23] A. Derhab and N. Badache: “A Pull-based Service Replication Protocol in Mobile Ad hoc Networks,” *European Trans. on Telecommunications*, Vol. 18, No. 1, pp. 1-11, Oct. 2005.

[24] C. A. Bellavista and P. E. Magistretti: “REDMAN: An Optimistic Replication Middleware for Read-only Resources in Dense MANETs,” *Elsevier Journal of Pervasive and Mobile Computing*, Vol. 1, No. 3, pp. 279-310, Aug. 2005.

[25] M. Hamdy and B. Konig-Ries: “A Service Distribution Protocol for Mobile Ad hoc Networks,” *Proc. of the 5th Int'l. Conf. on Pervasive Services*, pp. 141-146, Jul. 2008.

[26] V. D. Park and M. S. Corson: “A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks,” *Proc. of IEEE INFOCOM'97*, pp. 1405-1413, Apr. 1997.

[27] S. E. Athanaileas, C. N. Ververidis and G. C. Polyzos: “Optimized Service Selection for MANETs Using an AODV-based Service Discovery Protocol,” *Proc. of the 6th Mediterranean Ad Hoc Networking Workshop*, pp. 9-16, Jun. 2007.

[28] G. Cornuejols, G. Nemhauser, and L. Wolsey: “The Uncapacitated Facility Location Problem,” *Discrete Location Theory, P. Mirchandani and R. Francis, Eds. John Wiley & Sons*, pp. 119-171, 1990.

[29] R. Ramanathan and M. Steenstrup: “Hierarchically-orgainsed Multihop Mobile Networks for Qulaity-of-service Support,” *Mobile Networks and Applications*, Vol. 3, No. 2, pp. 101-119, Jun. 1998.

[30] A. Ephremides, J. E. Wieselthier and D. J. Baker: “A Design Concept for Reliable Mobile Radio Networks with Frequency Hopping Signaling,” *Proc. of the IEEE*, Vol. 75, No. 1, pp. 56-73, Jan. 1987.

[31] M. Gerla and J. T. Tsai: "Multicluster, Mobile, Multimedia Radio Network," *ACM Journal of Wireless Networks*, Vol. 1, No. 3, pp. 255-265, Oct. 1995.

[32] C. E. Perkins, E. M. Royer and S. R. Das: "Ad hoc On-demand Distance Vector (AODV) Routing," *IETF INTERNET DRAFT, MANET working group*, draft-ietf-manet-aodv-10.txt, Jan. 2002.

[33] Qualnet 4.0 network simulator, *Scalable Network Technologies Inc.*, 2007.

[34] D. Johnson and D. Maltz: "Dynamic Source Routing in Ad hoc Wireless Networks," *Mobile Computing*, pp. 153-181, 1996.

[35] K. Maeda, A. Uchiyama, T. Umedu, H. Yamaguchi, K. Yasumoto, and T. Higashino: "Urban Pedestrian Mobility for Mobile Wireless Network Simulation," *Int'l. Elsevier Journal on Ad hoc Networks*, Vol. 7, No. 1, pp. 153-170, Jan. 2009.

[36] A. Campbell, S. Eisenman, N. Lane, E. Miluzzo, R. Peterson, H. Lu, X. Zheng, M. Musolesi, K. Fodor, and G. Ahn: "The Rise of People-Centric Sensing," *IEEE Internet Computing Special Issue on Mesh Networks*, Vol. 12, No. 4, pp. 12-21, Jul./Aug. 2008.

[37] J. N. Al-Karaki and A. E. Kamal: "Routing Techniques in Wireless Sensor Networks: A survey," *IEEE Wireless Communications*, Vol. 11, No. 6, pp. 6-28, Dec. 2004.

[38] K. Akkaya, and M. Younis: "A survey on Routing Protocols for Wireless Sensor Networks," *IEEE Wireless Communications*, Vol. 3, No. 3, pp. 325-349, May 2005.

[39] L. Lima, and J. Barros: "Random Walks on Sensor Networks," *Proc. of the 5th Intl. Syposium on Modeling and Optimization in Mobile, Ad hoc, and Wireless Networks*, pp. 1-5, Apr. 2007.

[40] R. Shah, S. Roy, S. Jain, and W. Brunette: "Data Mules: Modeling a Three-tier Architecture for Sparse Sensor Networks," *Proc. of IEEE Workshop on Sensor Network Protocols and Applications*, pp. 30-41, May 2003.

[41] R. Katsuma, Y. Murata, N. Shibata, K. Yasumoto, and M. Ito: "Extending k-Coverage Lifetime of Wireless Sensor Networks Using Mobile Sensor Nodes," *Proc. of the 5th IEEE Intl. Conf. on Wireless and Mobile Computing, Networking and Communications (WiMob'09)*, pp. 48-54, Oct. 2009.

[42] W. Wang, V. Sirinivasan,and K. Chua: "Trade-offs Between Mobility and Density for Coverage in Wireless Sensor Networks," *Proc. of the 5th IEEE Intl. Conf. on Mobile Computing and Networking (MobiCom'07)*, pp. 39-50, Sep. 2007.

[43] B. Hull, V. Bychkovsky, Y. Zhang, K. Chen, M. Goraczko, A. Miu, E. Shih, H. Balakrishnan, and S. Madden: "CarTel: A Distributed Mobile Sensor Computing System," *Proc. of the 4th Intl. Conf. on Embedded Networked Sensor Systems (SenSys'06)*, pp. 125-138, Nov. 2006.

[44] R. Murty, A. Gosain, M. Tierney, A. Brody, A. Fahad, J. Bers, and M. Welsh: "CitySense: A Vision for an Urban-Scale Wireless Networking Testbed," *Proc. of Intl. Conf. on Technologies for Homeland Security*, pp. 583-588, May 2008.

[45] V. Tuulos, J. Scheible, and H. Nyholm: "Combining Web, Mobile Phones and Public Displays in Large-Scale: Manhattan Story Mashup," *Proc. of the 5th Intl. Conf. on Pervasive Computing*, pp. 37-54, May 2007.

[46] H. Lu, N. Lane, S. Eisenmanb, and A. Campbella: "Bubble-sensing: Binding Sensing Tasks to The Physical World," *Pervasive and Mobile Computing*, Vol. 6, No. 1, pp. 58-71, Feb. 2010.

[47] J. Shi, R. Zhang, Y. Liu, and Y. Zhang: "PriSense: Privacy-preserving Data Aggregation in People-centric Urban Sensing Systems," *Proc. of INFOCOM'10*, pp. 758-766, Mar. 2010.

[48] C. Cornelius, A. Kapadia, D. Kotz, D. Peebles, M. Shin, and N. Triandopoulos: "Anonysense: Privacy-aware People-centric Sensing," *Proc. of MobiSys08*, pp. 211-224, Jun. 2008.

[49] T. Abdelzaher and others: "Mobiscopes for Human Spaces," *IEEE Pervasive Computing*, Vol. 6, No. 2, pp. 20-29, Apr. 2007.

[50] R. K. Ganti, N. Pham, H. Ahmadi, S. Nangia, and T. F. Abdelzaher: "GreenGPS: a Participatory Sensing Fuel-efficient Maps Application," *Proc. of MobiSys10*, pp. 151-164, Jun. 2010.

[51] C. Vercellis: "A Probabilistic Analysis of The Set Covering Problem," *Annals of Operations Research*, Vol. 1, No. 3, pp. 255-271, Oct. 1984.

[52] R.M. Karp: "Reducibility Among Combinatorial Problems," *Proc. of a symposium on Complexity of Computer Computations*, pp. 85-103, Agu. 1972.

List of Major Publications

Journal Papers

1. Ahmed, A., Yasumoto, K., Ito M., Shibata, N., and Kitani, T.: "HDAR: Highly Distributed Adaptive Service Replication for MANETs," *IEICE Trans. on Information and Systems*, Vol. E94-D, No. 1, pp. 91-103 (Jan. 2011). (corresponding to Chap. 2)

2. Ahmed, A., Yasumoto, K., Yamauchi, Y., and Ito M.,: "Probabilistic Coverage Methods in People-Centric Sensing," *Journal of Information Processing (IPSJ)*, Vol.52, No.10, pp. 2902-2919, (Oct. 2011). (corresponding to Chap. 3)

International Conference

1. Ahmed, A., Yasumoto, K., Shibata, N., Kitani, T., and Ito M.,: "DAR: distributed adaptive service replication for MANETs," *Proc. of the 5th IEEE Int'l. Conf. on Wireless Mobile Computing, Networking, and communications (WiMob2009)*, pp. 91-97, (Oct. 2009). (corresponding to Chap. 2.4)

2. Ahmed, A., Yasumoto, K., Yamauchi, Y., and Ito M.,: "Distance and Time Based Node Selection for Probabilistic Coverage in People-Centric Sensing," *Proc. of the 8th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON'11)*, pp. 134-142, (Jun. 2011). (corresponding to Chap. 3.4.2.1 and 3.4.2.2)

3. Ahmed, A., Yasumoto, K., Yamauchi, Y., and Ito M.,: "Probabilistic Coverage in People-Centric Sensing," *Proc. of the 18th IEEE Int'l. Conf. on Network Protocols (ICNP 2010) Poster Session*, (Oct. 2010). (corresponding to Chap. 3.4.2.1 and 3.4.2.2)

Other Publications

Domestic Conference

1. Ahmed, A., Yasumoto, K., Shibata, N., Kitani, T., and Ito M.,: “Two-Layer Distributed Service Placement Method on Mobile Ad-hoc Networks,” *Technical Report of IPSJ*, 2008-MBL-46, pp. 41–48 (Sep. 2008).

2. Ahmed, A., Yasumoto, K., Shibata, N., Kitani, T., and Ito M.,: “DASR: Distributed Adaptive Service Replication for MANETs,” *Proc. of the 2009 Multimedia, Distributed, Cooperative and Mobile Symposium (DICOMO2009)*, pp. 823–835 (Jul. 2008).

3. Ahmed, A., Yasumoto, K., Yamauchi, Y., and Ito M.,: “Probabilistic Methods for Spatio-Temporal Coverage in People-Centric Sensing,” *Technical Report of IPSJ*, 2010-MBL-55, pp. 1–8 (Oct. 2010).

4. Ahmed, A., Yasumoto, K., Yamauchi, Y., and Ito M.,: “Node Selection Methods for Probabilistic Coverage in People Centric Sensing,” *Technical Report of IPSJ*, 2011-MBL-59, pp. 1–8 (Sep. 2011).