MAN, YANMAO, M.S. Information Distribution and Recharging Dispatch Strategy
in Large Wireless Networks. (2017)
Directed by Dr. Jing Deng. 53 pp.

Large wireless networks are envisioned to play increasingly important roles as more and more mobile wireless devices and Internet of Things (IoT) devices are put in use. In these networks, it is often the case that some critical information needs to be readily accessible, requiring a careful design of the information distribution technique. In this work, we at first propose PeB, Periodic Broadcast, that takes advantage of periodic broadcast from the information server(s) to leave traces for nodes requesting for the information while maintaining a low overhead. Similar to swarm intelligence, PeB requires each node to keep track of traces, or past records of information flow, through itself toward information servers. We present our extensive investigation of the PeB scheme on cost and network dynamics as compared to other state-of-the-art techniques. When the devices run out of battery, they become static and need to be recharged by the wireless charging vehicles (WCVs). Often times, WCV receives a number of charging requests and form a Hamiltonian cycle and visit these nodes one-by-one. We also propose a heuristic algorithm, termed *Quad*, that generates a Hamiltonian cycle in a square plane. We then focus on the theoretical study of the length of the Hamiltonian cycles in such networks.

INFORMATION DISTRIBUTION AND RECHARGING DISPATCH STRATEGY IN LARGE WIRELESS NETWORKS

by

Yanmao Man

A Thesis Submitted to
the Faculty of The Graduate School at
The University of North Carolina at Greensboro
in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Greensboro
2017

Approved by

Committee Chair

To my parents.

APPROVAL PAGE

This thesis written by Yanmao Man has been approved by the following committee of the Faculty of The Graduate School at The University of North Carolina at Greensboro.

Committee Chair ______________________________
Jing Deng

Committee Members ______________________________
Shan Suthaharan

____
Lixin Fu

____
Date of Acceptance by Committee

____
Date of Final Oral Examination

ACKNOWLEDGMENTS

Firstly, I would like to express my sincere gratitude to my advisor Prof. Jing Deng for the continuous support of my master study and related research on Wireless Adhoc Networks, for his patience, motivation, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my master study.

Besides my advisor, I would like to thank the rest of my thesis committee: Prof. Shan Suthaharan and Prof. Lixin Fu for their insightful comments and encouragement, but also for the hard questions which incented me to widen my research from various perspectives.

I would like to thank Xiami Music, Sennheiser IE80 and iPhone 6s for keeping my ears busy. Thank Baozoudashijian and TV sitcom Friends for their accompaniment during my dining time.

Last but not the least, I would like to thank my family for supporting me spiritually throughout writing this thesis and my my life in general.

TABLE OF CONTENTS

Page

CHAPTER

CHAPTER I

INTRODUCTION

1.1 Large Wireless Networks

Large wireless networks are expected to play increasingly important roles in the global expansion of information availability in the sense of virtual smart communities. Examples include Internet of Things (IoT), mobile ad hoc networks (MANETs), wireless sensor networks (WSNs), and vehicular ad hoc networks (VANETs). For example, different vehicles can move inside a large region with different patterns while they request for various information from some service nodes.

In these large wireless networks, there will usually be information that should be readily accessible by regular nodes throughout the entire network. Examples of such information are security certificates, public key copies, and service directories. The main goal of the information distribution is to store such information on one or several servers (or caches) so that, regular nodes can request for them when necessary.

Many techniques allow nodes to search for information in large wireless networks. For example, a Time-To-Live (TTL) based flooding can be used, but this has been found to cause the broadcast storm problem [NTS99]. In [JB04], a swarm-intelligence based scheme termed Ant-Based Evidence Distribution (ABED) was proposed, where reply packets with certificate copy leave traces for future querying packets. The Dynamic Unicast scheme [AWB16] shares similar strategy that records traces in Forwarding Information Base (FIB). Such approaches do not work well in mobile networks because traces can become outdated quickly. Kim and Ko [KK15] used content servers to broadcast Eligible Forwarder Selection (EFS) messages, which traverse the network graph in order to form a spanning tree. Such an approach depends heavily on the formation of the tree and its frequent broadcast behavior results in much overhead. The Anycast routing protocol was proposed in [HWH06, HHW10] for ad hoc pervasive networks. Anycast exploits ant colony behaviors to find a shorter path to a neighboring server efficiently and quickly.

Network dynamics in large wireless networks can easily render the above approaches inefficient or even unusable. In order to solve the information distribution problem in such large and dynamic networks, a delicate balance needs to be achieved between freshness of indication toward information servers and overhead to provide such indications. In this work, we propose a periodic broadcast scheme, termed PeB, that is based on traces and infrequent updates to allow query messages to route toward their closest information server with low overhead.

1.2 Wireless Rechargeable Networks

Wireless devices are usually run on energy stored on batteries. Due to physical constraints, these batteries will run out of energy sooner or later. Wireless power transfer technology is expected to pave the way for the so-called wireless rechargeable networks, in which wireless devices will have their batteries replenished from time to time, virtually extending their lifetimes forever [XSHS12a, RLX14].

Most often, there is one, or multiple wireless charging vehicles (WCVs) in such networks. WCV receives charging requests from wireless devices and chooses a list of them to visit/charge one after another. In this work, we focus on the distribution of the potentially travel distances of such charging trips. We are interested in the following question: assuming that there M devices that will be charged on a charging trip, what is the length distribution, such as expected distance, of the charging trip?

Our results obviously are applicable to not only wireless charging, but also data collection with mobile base stations. Note that Krivelevich et al. investigated Hamiltonian cycles in random graphs, but did not study the issue of its length distribution [KLS16].

Our contributions are summarized as follows.

(1) We propose *PeB*, Periodic Broadcast, that takes advantage of infrequent broadcasts from the (mobile) information server(s) to leave traces for nodes requesting for the information.

(2) We further propose a novel localcast strategy, *Localcast Reply*, that allows reply messages to find returning paths even when there exist broken links.

(3) We develop an analysis model to estimate the best announcement frequency for different network scenarios and confirm it with simulations.

(4) We perform extensive simulations to evaluate PeB with other related schemes in important performance metrics including delivery ratio, overhead, and overall efficiency. Our evaluation confirms the superiority of our PeB scheme over several state-of-the-art schemes.

(5) We propose *Quad*, Quadtree-based heuristic algorithm, that generates a Hamiltonian cycle given a network within a square plane. Our features include light time and space complexities, and only 5% more in length than near-optimal TSP algorithm.

1.3 Document Organization

The work is organized as follows: Chapter II discusses recent related works. In Chapter III, our schemes PeB and Quad, is explained in detail. Chapter IV presents our analysis on PeB's announcement frequency and distribution of Hamiltonian cycle length. Simulation-based performance evaluation is presented in Chapter V. In Chapter VI, we summarize our work and discuss future works.

CHAPTER II

RELATED WORK

2.1 Information Distribution

Information dissemination is a challenging task in large wireless networks due to the topology uncertainties. Flooding is a simple and straight-forward method to send out queries toward the certificate nodes. There have been many works trying to improve flooding efficiency. Cheng and Heinzelman [CH05] designed an optimal flooding strategy minimizing cost and latency for target discovery, although a large volume of querying traffic and potential collisions would be generated, especially with multiple queries. Similar approaches were taken in peer-to-peer systems. For example, Passive Distributed Indexing (PDI) [LW02] addresses the file sharing problem in mobile scenarios by eliminating the need of flooding to the entire network.

Swarm intelligence [Ken06] is the property of a system in which the collective behaviors of group species interact with the environment that they reside. The environment causes them to act as cohesively and highly self-organized. Research on ant-colony based swarm intelligence has been used to address dynamic optimization problems, such as traveling salesman problem and routing in communication networks. The power of such an approach can be explained below: consider how a certain ant specie find the shortest path to food source merely by laying and following pheromone on trails. In a simple case, ants starting from the nest to search for food would leave pheromone on trails. While some trails go nowhere, others lead them to food sources especially when some ants come back with food. Other ants follow (or are attracted by) these pheromones on the trails. Based on this observation, Jiang and Baras [JB04] considered query messages as forward ants that may be broadcasted or unicasted. The return information is considered as backward ants that leave traces for future forward ants to follow. However, communication cost and latency are high in their approach because many forward ants are likely to be broadcasted (flooded) in the neighborhood. Furthermore, complicated reinforcement rule can lead to trail loops in mobile networks, diminishing the benefits of such traces.

There are other techniques using swarm intelligence. A core-based multicast routing algorithm based on swarm intelligence was introduced by Shen and Jaikaeo [SJ05]. A similar announcement strategy was used in their proposed algorithm, although for multicast routing. An anycast routing protocol using swarm intelligence for service server was proposed and studied by Hoh et al. [HWH06, HHW10]. Compared to ABED, the proposed scheme in [HWH06, HHW10] has a unique feature of the EXPLORE packet sent by every node and forwarded with a small probability. When an EXPLORE message is received, the server responds the same way as it receives a regular QUERY message. In [LT10], a Stigmergic Landmark Routing was proposed for MANETs. The main feature of the proposed protocol is the use of landmarks, instead of all nodes, to record directions toward successful routing paths. A similar approach was suggested by Garcia and Pedraza [GP08], although load balancing is obtained through the exploitation of available information on forwarding choices by intermediate nodes. An energy saving and load balancing routing technique was proposed in [DRT09], using a novel pheromone updating policy based on multiple performance metrics. In [MDO$^+$10], an energy-aware routing was designed based on swarm intelligence and multipath transmissions.

Other techniques did not use swarm intelligence [DGV09] [ZZCD10] [JW05]. Power consumption is yet another important issue. Many strategies for cache management have been proposed and studied with the objective of minimizing energy consumption and balancing network load. Barbara and Imielinski [BI95] proposed three strategies that use invalidation reports and a stateless server. Broadcasting Timestamps (BT) technique allows clients to report timestamp of each cached item to server which, in turn, updates timestamp or purges cached items according to these reports. In Amnesic Terminals (AT), the server actively informs nodes the identifiers of the items that changed since the last invalidation report. SIG is the third technique that focuses on a new protocol between server and clients by signing a set of cached items instead of just one item. Other caching approaches have also been investigated [DGV09, ZZCD10, DKTM11]. Du et al. utilized cooperation zones, historical profiles, and hop-by-hop resolution to achieve cooperative caching in MANETs with proper settings of the cooperation zone radius [DGV09]. Zhao et al. used a novel asymmetric cooperative cache approach to identify the best cache location in wireless peer-to-peer networks [ZZCD10]. Ye et al. proposed to a decentralized cache placement technique for mobile peer-to-peer networks [YLC11]. Jin and Wang [JW05] argued that demand-driven proportional strategy for replication is far from optimal in 2-D mesh. They proposed an optimal strategy that replicates an object such that the number of its replicas is proportional to $p^{0.667}$, where p is the access probability of the object in multi-hop wireless mesh networks.

In [NSC03], Nuggehalli et al. addressed the issue of energy-conscious cache placement in wireless ad hoc networks. While caching of information stored at the server on several distributed nodes can improve information access delay, the energy consumption can be much higher due to the delivery of such information to the caching locations. A polynomial time algorithm was designed to compute the sub-optimal caching locations [NSC03]. In this work, however, we are interested in the problem of lowering query overhead in mobile wireless networks.

Recently, there are a lot of interests in Content-Centric Networking (CCN) architecture, where data is forwarded based on content names rather than IP addresses. Several works are closely related to our work and will be compared in Chapter V. In Dynamic Unicast [AWB16], querying node floods INTEREST until reaching one of the content servers. Content server replies DATA along the reverse path. Every intermediate node that DATA passes through records the pre-hop into FIB for future relays. Each entry in FIB will expire and be deleted in a certain period of time (validity time) unless new DATA comes from the same node, resetting the validity time. Also, there is only one entry in FIB because Single Face Forwarding (SFF) works better than Parallel Face Forwarding (PFF) in terms of overhead and latency. In Kim&Ko [KK15], content server periodically announces its existence by broadcasting EFS messages, which is basically equivalent to transforming a graph to a spanning tree using breadth-first search (BFS). Once the tree is formed, a querying node can unicast its INTEREST packet to its parent node, or Eligible Forwarder (EF), toward the content server. Then the DATA packet is replied along the reverse path.

Many works tried to adopt CCN into MANETs. Kim et al. [KK15] proposed a three-tier strategy that retrieves content in multi-hop wireless CCN. Anastasiades et al. [AWB16] proposed a routing protocol named Dynamic Unicast, which depends on broadcast and dynamical unicast links to retrieve content in MANETs.

Information-Centric Network (ICN) is another popular approach on which many works has been focusing, e.g., [LLYG14] trying to adopt ICN into DTN and [VES+13] trying to adopt ICN into MANETs.

The problem we are focusing on shares many similarities with data dissemination in wireless networks [IGE+03, DQA04, ZCP07, LLPG10]. In fact, the certificate information we discuss in this work can be any other information that is needed throughout the network. In order to improve data availability, researchers proposed and investigated data caching [XHLL04, ZCP07, JH08, KDT10, YZS11]. However, we focus on a technique making information available for query instead of information delivery by simple broadcast.

2.2 Travel Salesman Problem and Wireless Rechargeable Networks

Martello presented an algorithm to identify Hamiltonian circuits in directed graphs in [Mar83]. The algorithm used branching decisions and backtracking to filter and record potential Hamiltonian circuits.

The well-known Traveling Salesman Problem (TSP) is certainly related to what we are trying to achieve. There are many works have been done on TSP [ABCC11]. Arora [Aro98] proposed an approximation algorithm for Euclidean TSP problem, where coordinates of nodes are at first perturbated, then a shifted Quadtree is constructed. Finally, dynamic programming is performed to find the optimal Hamiltonian cycle. However, we focus on the distance distribution of such routes in this work.

Wireless Sensor Recharging problem has been focused on in recent years. The problem is to schedule a route for one or more wireless charging vehicles (WCVs) so that they could recharge those devices in low battery. A recent survey is presented in [LWN+16]. The optimal traveling path for WCVs being the shortest Hamiltonian cycle was proved in [XSHS12b]. A hexagone-based scheme was proposed in [XSH$^+$15]. However, the main target of this work is not to give a recharging dispatch algorithm. Instead, we aim to give a probabilistic distribution of Hamiltonian cycle given the density and the size of square plane where devices locate in.

CHAPTER III

SCHEME DESIGN

3.1 PeB: Periodic Broadcast for Information Distribution

3.1.1 Primary

The operation of PeB is based on query messages and reply messages, which are standard in many information distribution systems or routing techniques. What makes PeB unique are the traces and the way how the traces are updated. Traces are information stored on each node and will lead query messages toward the information servers, each of which carries a copy of the same information.

See Figure 1 for an illustration of the PeB scheme, where Node D is one of the information servers. The querying node, node S, is searching for it. Broadcast Query (BQ) messages are sent from node S and re-broadcasted by intermediate nodes with no trace to the any information server. Node F has a trace toward the information server and sends out a Unicast Query (UQ) message instead. Unicast Reply (UR) messages are sent from node D back to node S (the red dotted line). The large dashed circle (in blue) shows the range of the periodic broadcast sent from node D. Each node stores a Trace Table (TT) and uses it to choose the best next-hop node to send UQ messages whenever possible.

Every node stores a data structure called Trace Table (TT), which serves as a guide for local decision making (see Figure 1). Each entry on TT records the trace for a known information source. In network routing, trace is basically a probability value of reaching the information server when the node is chosen to forward the

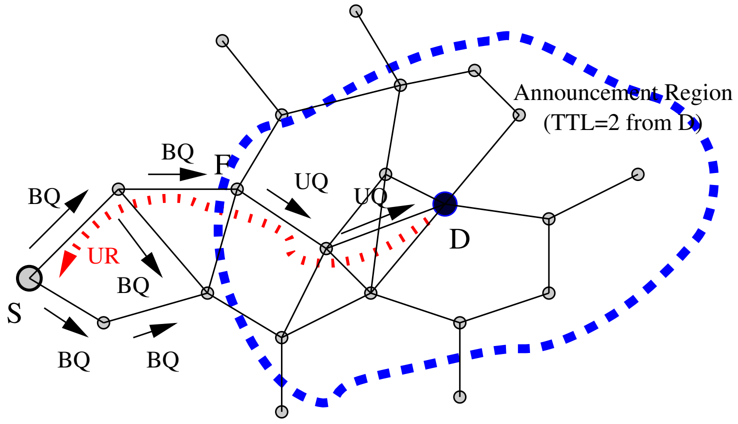


Figure 1. Illustration of the PeB scheme.

Table 1. Trace Table

Neighbor	N_1	N_2	.....	N_m
Trace	τ_{N_1}	τ_{N_2}	.....	τ_{N_m}

query packets. On Table 1, we show a Trace Table example. The node has m active neighbors, $N_1, N_2, \cdots N_m$. The trace of the information server through node N_1 is recorded as τ_{N_1}, N_2 with τ_{N_2}, etc. These values will be updated through two mechanisms: the Unicast Reply (UR) messages and Broadcast Announcement (BA) messages. The UR messages will travel through the reverse of the original path through which the query message has gone. And the traces will be updated (details to be provided in Chapter 3.1.2). The BA messages will be sent from the information servers periodically to announce the existence and locations of the information servers. The frequency and scope of the broadcast will affect the PeB scheme's performance and will be investigated in Section 4.1 and Chapter V.

The PeB scheme shares some similarities with other schemes with the swarm intelligence approach. In ant swarm intelligence, for instance, trail pheromone is a

Table 2. Different Messages in the PeB Scheme

Acronym	Details
BA	Broadcast Announcement message
BQ	Broadcast Query message
UQ	Unicast Query message
UR	Unicast Reply message
LR	Localcast Reply message

secreted chemical substance that laid on trails by ants when they bring food back to their nest. It attracts other ants and serves as a guide to find food. As more and more ants take the same route, they too lay pheromone, further amplifying concentration of pheromone and attractiveness of the trail. An evaporation process that slowly lowers the pheromone concentration is responsible to erase any old and outdated trails. There are several aspects that distinguish the PeB scheme from swarm intelligence:

- Periodic broadcast from information servers.
- PeB does not have an evaporation process. Such a process is unnecessary for PeB since periodic updates are sent from the information servers.

The acronyms of different messages are summarized in Table 2.

3.1.2 Scheme Details

Our scheme can be divided into three phases: Announcement phase, Query phase, and Response phase. While the Response phase usually follows the Query phase, the Announcement phase has its own schedule that is independent of the other two phases.

3.1.2.1 Announcement Phase

In the Announcement phase, the information servers make periodic announcements of their existence in their neighborhood. This is one of the main differences of the PeB scheme with other previously proposed schemes such as [JB04]. We argue that such periodic announcements update the trace table in the neighborhood of the certificate copies especially in mobile networks and allow the query messages to be routed toward the information server more accurately.

In contrast, other schemes, such as [AWB16], rely on the broadcast technique to look for the information servers and keep track of the traces left by the reply messages. While such a technique works well in static networks, mobile networks introduce a much more difficult problem: outdated traces. Trace can become invalid quickly in mobile networks and the implemented trace degradation technique cannot reflect the process accurately. This is also true for techniques that employ the evaporation process, which is in parallel to network topology changes.

The announcement will be made in the local neighborhood of each information server. In particular, the announcement messages (BA) will be flooded to the TTL-hop (BA-TTL) of the neighborhood. We argue that an appropriate BA-TTL value will allow a majority of the querying nodes to have access to updated traces. Obviously, a large BA-TTL will lead to higher overhead and should be avoided. In addition, a larger BA-TTL may not always be better even if we neglect the extra cost that it introduces. This is because each information server has a region of nodes that should contact itself for the information. Further increasing BA-TTL over this region size can only complicate the path selection process at the forwarding nodes. We investigate the effect of different BA-TTL values in Chapter V.

Another important system parameter in the Announcement phase is the *announcement frequency*, f. A large f will cause too much of message transmission overhead; on the other hand, a small f can easily leave the traces outdated especially in large

Table 3. Announcement Message Format

FIELDS	REMARKS
SEQ_NUM_	announcement sequence number
LEN_	packet length
SRC_ADDR_	address of announcement node
SERVER_ID_	carrying server identity
LAST_SENDER_	ID of the previous sender
HOP_COUNT_	hop count from the information server
TTL_	time-to-live

wireless networks with mobile nodes. In the rest of this work, we will also employ the term, *announcement interval*, which is equivalent to $1/f$. We present our investigation of f based on network dynamics in Chapter V. The information carried by the announcement messages includes server identity, announcement sequence number, TTL, etc. An illustrative packet format is provided in Table 3.

When a node overhears an announcement message, it updates its trace toward the last sender (LAST_SENDER_) of the message:

$$\tau_{n+1} = \gamma\tau_n + (1-\gamma)(H-h) \ , \tag{3.1}$$

where γ represents the weight toward the old estimate, h is the hop count from the announcement server, and H is the maximum hop count of the network. We use the trace as $H-h$ because the shorter the hop count is, the higher weight the node should have. Furthermore, $H-h$ ensures that the term is non-negative.

3.1.2.2 Query Phase

A node requesting for the information initiates the Query phase, in which a query message is sent and intermediate nodes help to route the query message toward an information server.

The difference of regular packet routing and the Query phase is the self-duplication of query message. In fact, the duplication rate of the query message in the Query phase is between unicast and broadcast. In unicast, a single copy of the message is forwarded throughout a network path. In broadcast, every node receives a copy of the message and forwards it at least once. In the Query phase of the PeB scheme, only those nodes without any trace toward the information server will (re)broadcast the query messages (BQ); all other nodes with some traces toward the information server will unicast the query messages (UQ) toward the best candidate, which will forward the message in a similar fashion.

The logic behind such a strategy is rather straightforward: when none of the neighbors know about any information server, the sending node has no other choice but to send it to every neighbor. When one or more neighbors have traces toward an information server, the neighbor with highest trace will be chosen to forward the query message.

The query message also carries the IDs of nodes through the path of which it has traversed. Such a path information will be needed in the Response phase, which is in charge of sending the requested information back to the querying node. The path information can also be used to eliminate routing loop.

Care must be taken to remove query messages running the network endlessly, either through a loop or some mis-routes. In order to ensure this behavior, a TTL value is inserted in each query message generated by the querying node. Every node processing a message will decrement the TTL value. When TTL reaches 0, the message will be purged from the network.

3.1.2.3 Response Phase

Once a query message reaches one of the information servers, the Response phase will be initiated by the information server. In particular, a UR message will be generated and sent back to the querying node.

There are two issues worth of discussion: the UR message will use the route information stored on the arriving query message; there might be more than one query message arriving at the information server. This is an obvious result of the BQ transmission. The information server can either ignore the subsequent messages or reply to some of them. In this work, we assume that the information server only replies to the first request and ignores the rest.

All nodes forwarding the UR message will update their trace table based on its hop count from the information server and the last sender of the UR message. The trace update formula is the same as the updating formula for nodes receiving BA messages. Even though a different weight can be used, we use the same weight in both UR and BA messages.

However, there might be broken paths due to mobility or other network dynamics when UR messages are on the way. In this case, the UR can no longer be forwarded further. In order to solve this problem, we propose an strategy named *Localcast Reply* that helps the UR messages to skip the broken link and then go on further. In particular, when facing a broken path, the UR message will turn to a Localcast Reply (LR) message which will be broadcasted with a small TTL (LR-TTL) until reaching one of those nodes left in the path that the UR message was supposed to go through. Then the LR message will switch back to UR message that continues to be unicasted. See Figure 2 for an example. r_5 is the querying node and i_0 is an information server.

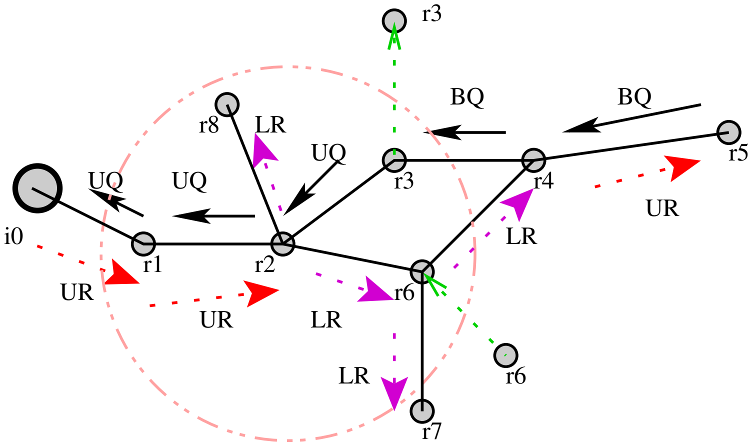


Figure 2. Illustration of Localcast Reply strategy.

Query message is forwarded along the path $[r_5, r_4, r_3, r_2, r_1, i_0]$ that the UR message is supposed to pass through in reverse direction to reach r_5. Unfortunately, r_3 moves out of the transmission range of r_2, forming a broken path since the link between r_2 and r_3 is gone. However, at the same time r_6 moves in and the new link between r_2 and r_6 is created. When the UR message arrives at r_2, who notices the broken link between itself and r_3, it broadcasts an LR message to its neighbors with TTL of 2 hops. r_6 receives this LR message and re-broadcasts it to r_4 and r_7. In this case, r_4 is on the original (reverse) path. Therefore, the UR message continues to be unicasted to r_5 by r_4. We will investigate TTL of LR message (LR-TTL) in Chapter V.

3.2 Quad: Quadtree-based Heuristic Algorithm for TSP

3.2.1 Problem Statement

We focus on a network with a size of $r \times r$ with N nodes and each of these N nodes needs to be charged from time to time. Nodes send their expected battery life to WCV from time to time and WCV picks $M < N$ of these requests to be charged.

We are interested in the following problem:

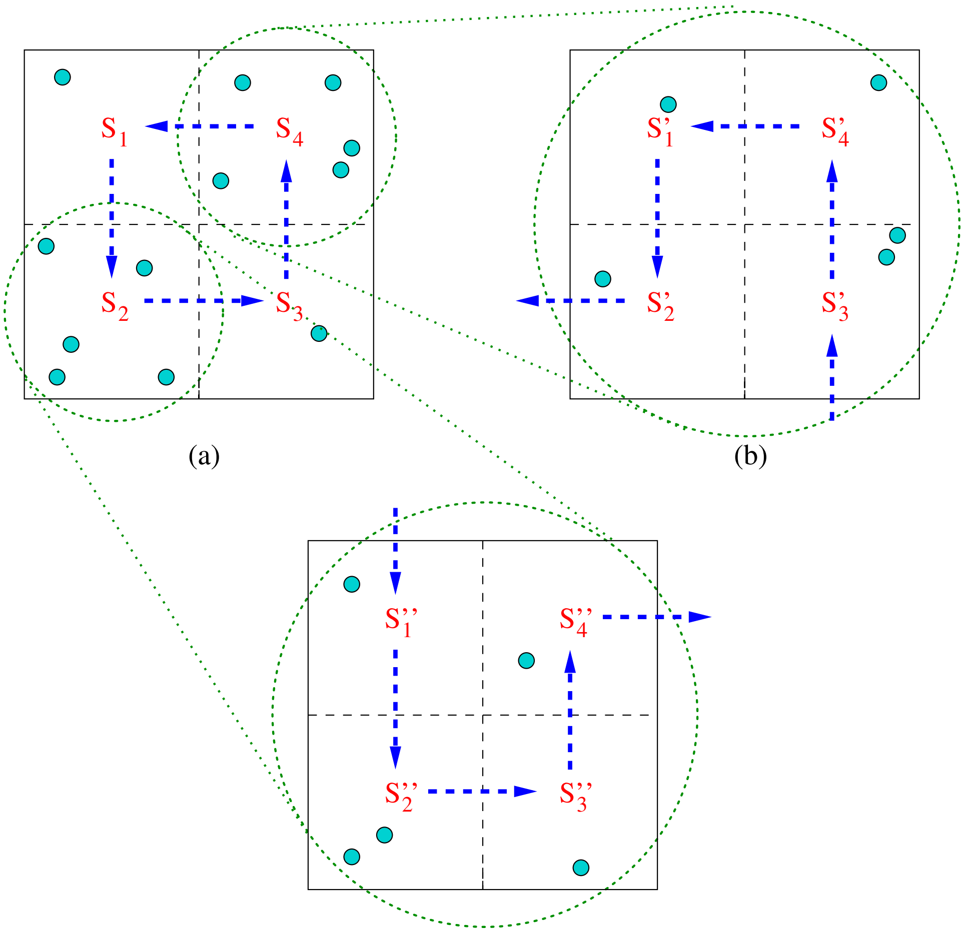
Hamiltonian Cycles (HAM): *What is the distribution of the Hamiltonian cycles in the above system if we ignore charging deadlines $\tau_i(t)$ for all $i = 1, 2, \cdots M$?*

Before we can derive such a distribution, we give a heuristic algorithm termed *Quad*.

3.2.2 Primary

Quad is based on Divide-and-Conquer strategy that takes the locations of M nodes in a square block as input and generates the Hamiltonian Cycle for WCV to follow. Note that the presented algorithm is partially similar to the idea given in [Aro98]. See Figure 3 for the principle of Quad algorithm.

In Figure 3a, a square block is divided into 4 equal sub-blocks labeled as S_i, $i = 1, 2, 3, 4$. WCV follows the blue dotted arrows to visit each sub-block before returning to the starting point in S_1. In Figure 3b, we zoom into sub-block S_4, which is further divided into 4 sub-blocks. WCV follows the routing sequence of (S'_3, S'_4, S'_1, S'_2). When finishing visiting S'_2, WCV goes back to the starting point located in S_1. In Figure 3c, we zoom into S_2. The routing sequence here is $(S''_1, S''_2, S''_3, S''_4)$. When done with S''_4, WCV moves to the next bigger block S_3 and will follow the routing sequence there. As shown in Figure 3a, M nodes are placed in a $r \times r$ square block S_0, which is divided into four $r/2 \times r/2$ sub-blocks that are marked as S_i, $i = 1, 2, 3, 4$. WCV visits each sub-block from one to another before returning to the starting point. Each sub-block is further divided quadruply into smaller ones, where WCV follows similar route to visit the nodes and then goes to the next bigger block as illustrated in Figure 3b and 3c when we zoom in. One example of full routing pattern is shown in Figure 4 when we assume that every block contains exactly one nodes, which is actually unnecessary.



(a) (b)

(c)

Figure 3. Principle of Quad Algorithm.

Normally, some blocks may be empty and some blocks do not need to be divided such deeply, because there is at most one node in each of them. More realistic example of division is shown in Figure 5, where we can observe that all blocks contain at most one node and WCV only visits the non-empty blocks rather than all blocks.

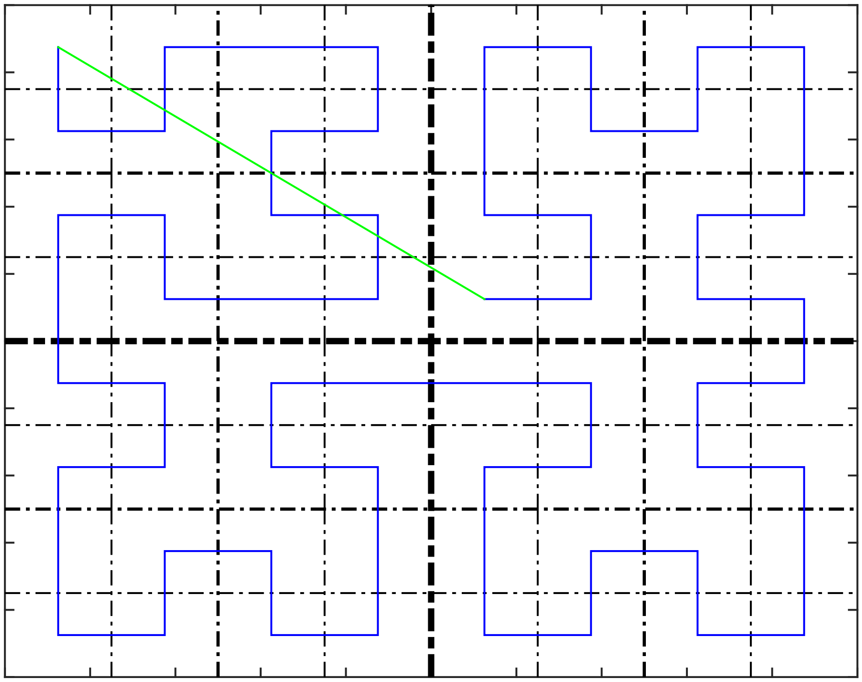


Figure 4. Full pattern to visit all blocks.

3.2.3 Algorithm Details

Because the blocks are divided into 4 equal sub-blocks when necessary, we use *Quadtree* [Sam84] as the data structure where every parent owns exactly 4 children. Each vertex in $\mathcal{T}$ represents a block and its children represent the sub-blocks. In this subsection, the terms "block" and "vertex" are exchangeable, so are the terms "sub-block" and "child".

The biggest block is represented by vertex *root*, which is the root of $\mathcal{T}$. For each vertex *parent*, it contains two elements, *seq* and *nodes*. *seq* decides the visiting sequence of its sub-blocks. For example, if $seq = (S_3, S_4, S_1, S_2)$, then the route is shown in Figure 3b. *nodes* is the set of nodes that locate within this block. Take S_4 in Figure 5 for an example, $nodes_4 = \{n_7, n_8, n_9\}$.

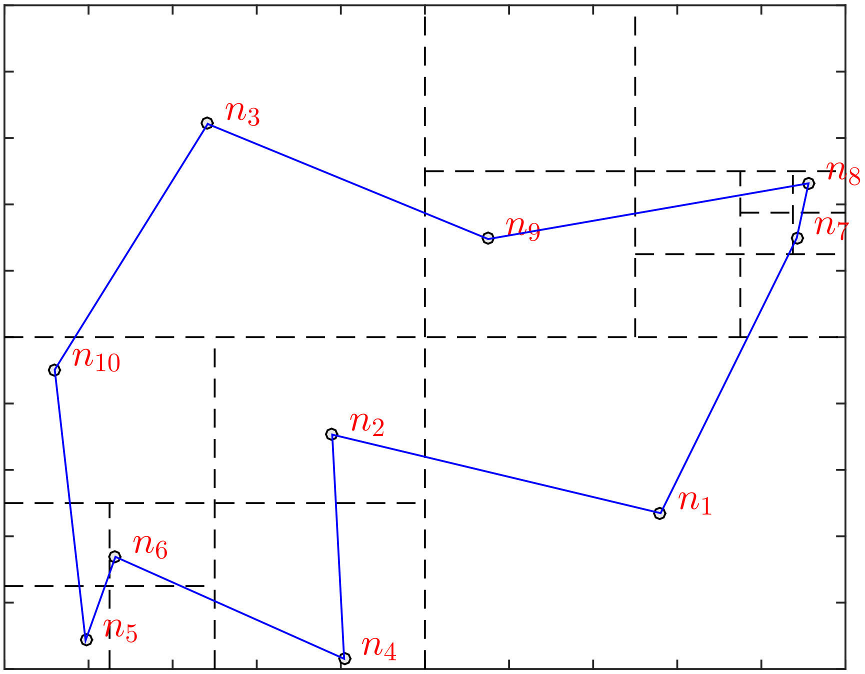


Figure 5. An example of route in recharging network.

The Quadtree $\mathcal{T}$ is built up in Algorithm 3.1 from Line 1 to Line 20 in the way of Breadth-first Search (BFS) with a queue Q. For every vertex *parent* in Q, we at first test if its *nodes* set contains at most one nodes. If so, then we continue to the next iteration because we do not have to further divide this block. If not, then we have to at first derive the sequences belonging its children seq_i, $i = 1, 2, 3, 4$.

In order to generate the pattern shown in Figure 4. The *seq* relationship between parents and their children is interesting. Suppose sequence of the *parent* is $seq = (S_a, S_b, S_c, S_d)$, the sequence of its first child S_a is $seq_a = (S_a, S_d, S_c, S_b)$. Note that the first child is unnecessarily S_1. For example, in Figure 3b, the first child S_a is S'_3. The second and third children share the same sequence with *seq*. The fourth one depends on whether the *parent* is the root of $\mathcal{T}$ or not. If so, then $seq_d = (S_c, S_d, S_a, S_b)$.

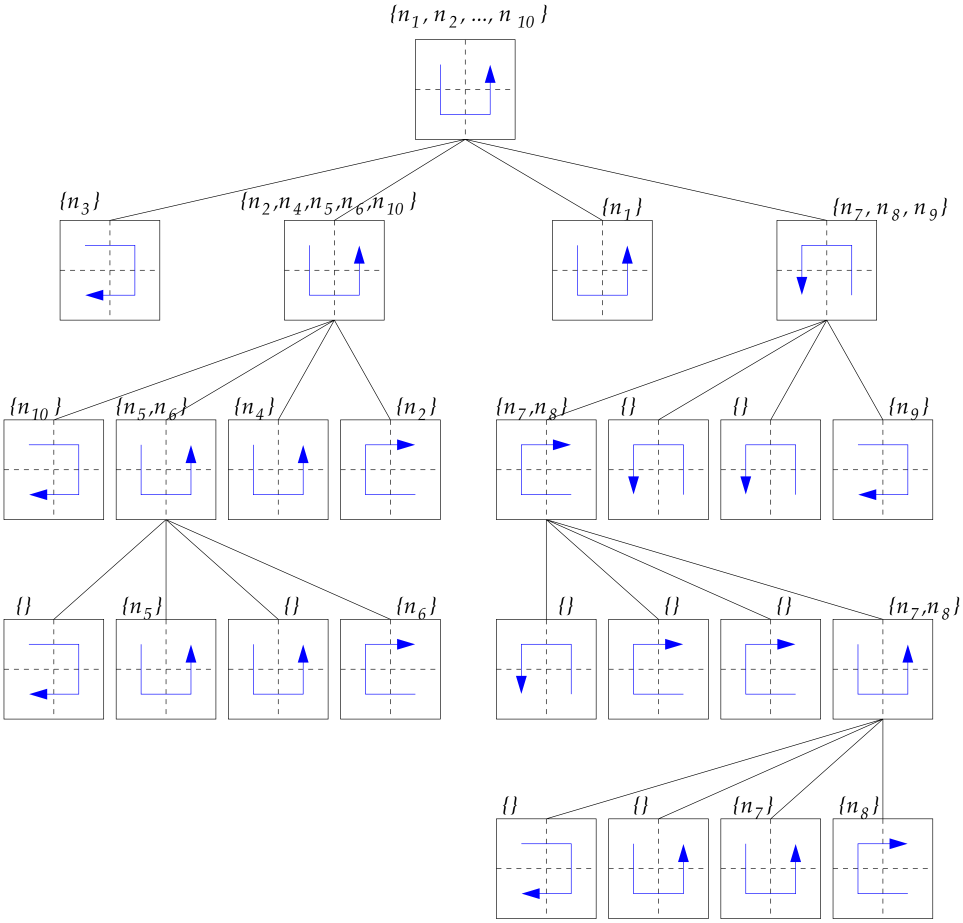


Figure 6. An example of Quadtree $\mathcal{T}$.

Else, then $seq_d = (S_c, S_b, S_a, S_d)$. The reason of such a difference is that if the *parent* is the root, which represents the biggest block, then when WCV finishes the fourth sub-block, it goes back to the starting point (see S'_2 in Figure 3b). If *parent* is not the root, the WCV goes to the next bigger block (see S''_4 in Figure3c).

Having the seq_i and $nodes_i$ of the 4 sub-blocks, we attach them to *parent* as its children in the order of how *seq* indicates and put them into Q. When Q becomes empty and the **while** loop breaks, we run DFS on $\mathcal{T}$ and output those *nodes* having exactly one node in them. Given the locations of $M = 10$ nodes in Figure 5, an example of Quadtree generated by Algorithm 3.1 is shown in Figure 6, where each vertex of $\mathcal{T}$ is represented by a solid square, in which the blue solid arrow indicates the routing sequence *seq*. The sets *nodes* are written on top of the squares. Solid black lines represents the parent-child relationship. We can see that there are exactly 4 children for each parent with more than one node in its *nodes* set. Those do not have child are with either one node or none in their *nodes* sets. When we run DFS on this $\mathcal{T}$, meanwhile output only the non-empty *nodes* of the leave, we could have the routing sequence $\{n_3, n_{10}, n_5, n_6, n_4, n_2, n_1, n_7, n_8, n_9\}$, which is the same to the result plotted in Figure 5.

Algorithm 3.1.

```
$seq \leftarrow (S_1, S_2, S_3, S_4)$
$root \leftarrow (seq, nodes)$
Add root into a queue $Q$
while $Q \neq \emptyset$ do
    $parent \leftarrow Q.poll()$
    $(seq, nodes) \leftarrow parent$
    if Size of $nodes_0$ is 1 then
        Continue
    end if

    $(S_a, S_b, S_c, S_d) \leftarrow seq$
    $seq_1 \leftarrow (S_a, S_d, S_c, S_b)$
    $seq_2 \leftarrow seq_3 \leftarrow seq$
    if parent is root then $seq_4 \leftarrow (S_c, S_d, S_a, S_b)$
    else $seq_4 \leftarrow (S_c, S_b, S_a, S_d)$
```

```
    end if

    for $i = 1$ to $4$ do
        $nodes_i \leftarrow$ nodes within the sub-block seq
        $child_i \leftarrow (seq_i, nodes_i)$
        Add $child_i$ into Q
        Attach $child_i$ as No.i child of parent
    end for
end while
Run Depth-First Search (DFS) from root, meanwhile output non-empty set nodes of every leaf.
```

CHAPTER IV

ANALYSIS

4.1 Announcement Frequency

As discussed in Chapter III, a high announcement frequency, f, incurs large announcement overhead with many broadcasts. On the other hand, a low f will run the risk of leaving outdated traces for intermediate nodes. The optimum f should allow the information server to send out announcements frequent enough that traces are not outdated due to mobility or other network dynamics, but it does not generate too much announcement overhead.

The accurate analysis of node movement, residual link lifetime [GdWFM02,SW04, HH09], and announcement overhead can be rather complicated. Therefore, we develop an approximate model to analyze link stability in mobile wireless networks and derive the best announcement frequency. In particular, we investigate the statistics of *link lifetime*, the duration between the time that a link is established because two mobile nodes have just moved close to each other and the time that it is broken because the distance between these two nodes has increased beyond the wireless communication range. We define variable $\lambda(A, B)$ to represent the lifetime of the link established between nodes A and B. We will then use the statistics distribution of λ to study the expected lifetime of a path toward information servers.

When two nodes, A and B, establish a new communication link, the distance between them is simply R, the wireless communication range (see Figure 7). Assume that the movements of these two nodes are statistically identical and also independent, with the absolute value of speed uniformly distributed between 0 and V_m, where V_m is the maximum speed of all nodes, and the direction uniformly distributed between 0 and 2π. Define the speed and direction of node A's movement as u, θ. Similarly define v and ϕ for node B. Denote $\boldsymbol{U}(0, V_m)$ a uniform distribution between 0 and V_m. We have

$$u, v \in \boldsymbol{U}(0, V_m) \quad \text{and} \quad \theta, \phi \in \boldsymbol{U}(0, 2\pi) \ .$$

Denote the absolute value of relative speed between nodes A and B as w and the movement angle as ψ. The CDF of relative speed between nodes A and B can be expressed as

$$\begin{aligned} & F_W(w) = Pr\{W < w\} = \oint_{u,v,\theta,\phi} \\ & Pr\{(u\cos\theta - v\cos\phi)^2 + (u\sin\theta - v\sin\phi)^2 < w^2 \\ & |U = u, V = v, \Theta = \theta, \Phi = \phi\} \\ & \cdot f_U(u) f_V(v) f_\Theta(\theta) f_\Phi(\phi) du dv d\theta d\phi \end{aligned}$$

and

$$f_W(w) = \frac{\partial F_W(w)}{\partial w} \ . \tag{4.1}$$

It can be proven, at least numerically, that the direction of the relative movement between nodes A and B, Ψ, follows $\boldsymbol{U}(0, 2\pi)$:

$$f_\Psi(\psi) = \frac{1}{2\pi} \qquad 0 \le \psi \le 2\pi \ . \tag{4.2}$$

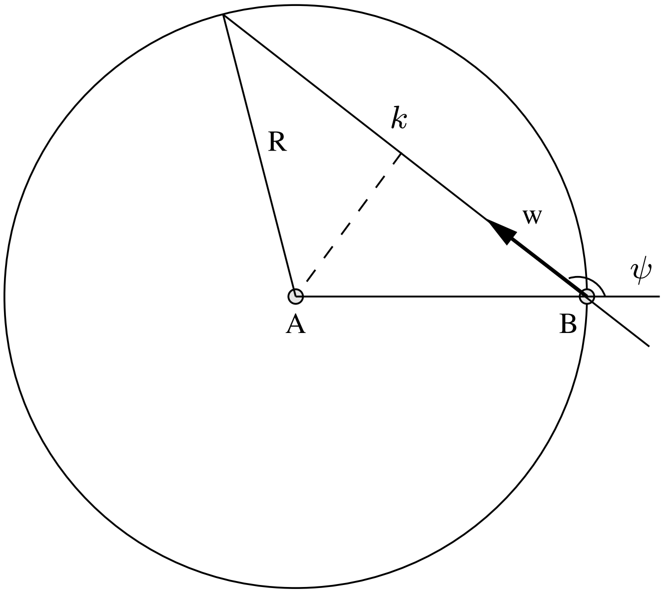


Figure 7. Nodes A and B with relative movement of w and angle of ψ. The actual distance of travel before nodes A and B become out of range is k.

With such a relative speed between nodes A and B, the lifetime of link A-B can be computed as (see Figure 7):

$$\lambda(A, B) = \frac{k}{w} \ , \tag{4.3}$$

where k is the line cut by the relative movement (of node B on node A) on the circle centered at node A and with radius of R:

$$k = 2R\cos(\pi - \psi) \ . \tag{4.4}$$

Note that k is maximum ($2R$) when $\psi = \pi$ and minimum (0) when $\psi = \pi/2$. Also, when $0 < \psi < \pi/2$, it means that nodes A and B are moving apart with link establishment occurred earlier on. Define random variable Λ as the lifetime of link

A-B. The CDF of Λ can be computed as

$$\begin{aligned} & F_\Lambda(\Lambda < \lambda) \\ = \ & \int_{w=0}^{2V_m} \int_{\psi=0}^{2\pi} U(\frac{k}{w} < \lambda) f_W(w) f_\Psi(\psi) dw d\psi \ , \end{aligned} \tag{4.5}$$

and

$$f_\Lambda(\lambda) = \frac{\partial F_\Lambda(\lambda)}{\partial \lambda} \ . \tag{4.6}$$

We can now compute the chance of a link broken due to node mobility. Assume that the used path spans from outside of L hops toward the information server. If any hop on the path is broken, the path is considered broken and unusable. Such an assumption is obviously an average and approximation of all paths that are used. In practice, we expect L to be closely related to the number of information servers, node density in the network, as well as scope of BA message, BA-TTL. A detailed calculation of L is considered out of the scope of this work, but it should be great than BA-TTL unless the number of information servers are large. There are increasing number of nodes with increasing hop counts toward the information server and many of them may reside outside of the BA-TTL-hop region.

We first compute the probability that all L hops will not break given that the path is used at time t, where $0 \le t \le 1/f$, where f is the announcement frequency. For each hop, the chance of not breaking is

$$P_1 = Pr\left[\Lambda > (\frac{1}{f} - t)\right] = 1 - \int_0^{\frac{1}{f}-t} f_\Lambda(\lambda) d\lambda \ . \tag{4.7}$$

The chance that none of the L hops will break, $\eta(f)$, is simply

$$\eta(f) = (P_1)^{\mathrm{L}} \ . \tag{4.8}$$

Define announcement frequency efficiency as the ratio between probability of no broken path and announcement frequency

$$\Upsilon(f) = \frac{\eta(f)}{f} \ , \tag{4.9}$$

and the optimum broadcast frequency can be obtained by finding f that maximize Υ:

$$f^* = \arg\max_f \{\Upsilon(f)\} \ . \tag{4.10}$$

We show $\Upsilon(f)$ as a function f for different V_m in Figure 8 with $L = 2\cdot$BA-TTL. V_m is chosen as 2, 5, and 10 m/s. Clear optimum of different announcement frequencies can be seen in Figure 8. For instance, for $V_m = 5$ m/s, announcement frequency efficiency first increases with f and then drops off after $f^* = 0.0214$, which represents roughly an announcement interval of 45 seconds. As V_m increases, the optimum announcement frequency increases as well. In fact, an almost linear relationship between f^* and V_m can be observed and this can also be observed in Figure 12.

4.2 Distribution of Length of Hamiltonian Cycle

First at all, we focus on an area of $r \times r$ and mark four sub-block as illustrated in Figure 3a. Each of these blocks has a size of $r/2 \times r/2$. We name these blocks S_i, $i = 1, 2, 3, 4$, counter-clock wise starting from top left.

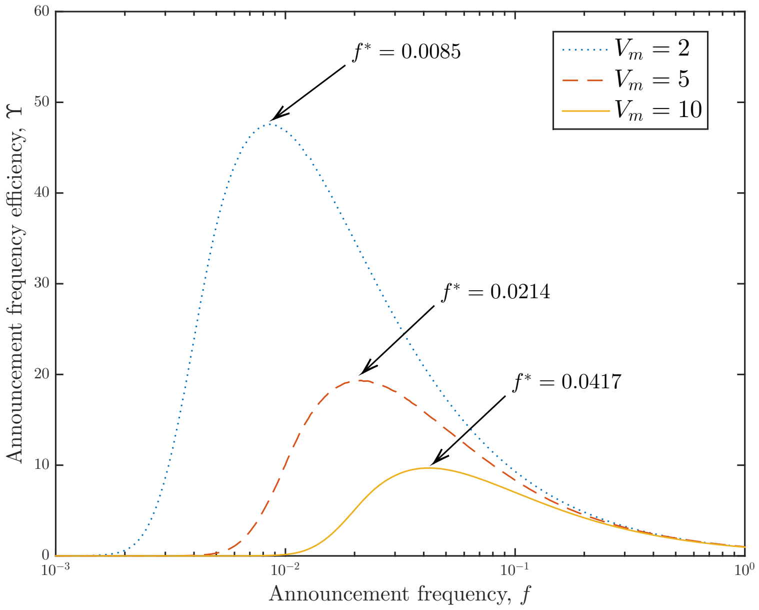


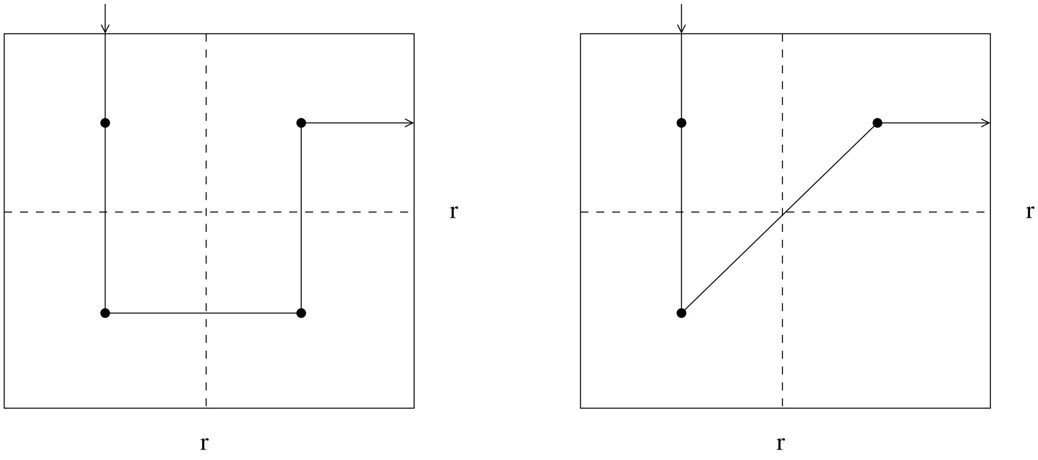
Figure 8. Announcement frequency efficiency for different maximum speed. L is chosen to be $2 \cdot$ BA-TTL.

We introduce a binary tuple $A = (b_1, b_2, b_3, b_4)$ to represent whether each sub-block contains *at least* one node. For example, $A = (1, 1, 0, 1)$ means each of the sub-blocks s_1, s_2, s_4 contains at least one node and the sub-block s_3 does not contain any node. Obviously there are 15 different settings for A, which we define as $\mathcal{A}$, ranging from $(0, 0, 0, 1)$ to $(1, 1, 1, 1)$.

Define $f(r, \rho, n)$ as PDF of number of nodes within a block $r \times r$, given density ρ. Based on Poisson point distribution, we have

$$f(r, \rho, n) = (\rho r^2)^n / n! \cdot e^{-\rho r^2} \ , \tag{4.11}$$

where $M = \rho r^2$.



(a) $A = (1,1,1,1)$ (b) $A = (1,1,0,1)$

Figure 9. Illustrations of $h(A, r)$ with different A.

Define $g(A, r, \rho)$ as the probability that, for an $r \times r$ block with node density ρ, the nodes are distributed as A indicates.

$$g(A, r, \rho) = (1 - f(\frac{r}{2}, \rho, 0))^{|A|} \cdot l(f(\frac{r}{2}, \rho, 0))^{4-|A|}, \tag{4.12}$$

where $|A|$ is the number of "1" in tuple A. Assume each node is at the center of its block. Define $h(A, r)$ as the Euclidean distance needed to visit all the nodes locating at the center of sub-blocks and the path should enter the top-left sub-block and exit the top-right sub-block. For example, when $A = (1, 1, 1, 1)$, then $h((1, 1, 1, 1), r) = 2r$ as illustrated in Figure 9a. When $A = (1, 1, 0, 1)$, then $h((1, 1, 0, 1), r) = (1 + \frac{\sqrt{2}}{2})r$ as illustrated in Figure 9b. There are 15 mappings as listed in Table 4. Define $D(r, \rho)$ as the mean of length of Hamiltonian cycle needed for an $r \times r$ block with node density ρ. It can be calculated recursively.

$$D(r, \rho) = \sum_{A \in \mathcal{A}} \left[h(A, r)(1 - \frac{1}{|A|}) + |A| \cdot D(\frac{r}{2}, \rho) \right] \cdot g(A, r, \rho) \ , \tag{4.13}$$

Table 4. Function $h(A, r)$

A				$h(A, 1)$
0	0	0	1	0.809
0	0	1	0	1.4604
0	0	1	1	1.6514
0	1	0	0	1.6514
0	1	0	1	1.7071
0	1	1	0	1.809
0	1	1	1	2
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1.5161
1	0	1	1	1.7071
1	1	0	0	1.6514
1	1	0	1	1.7071
1	1	1	0	1.809
1	1	1	1	2

in which the term $1 - \frac{1}{|A|}$ reduces the overlap of length between different recursive levels. $|A| \cdot D(\frac{r}{2}, \rho)$ represents the length sum of sub-blocks.

Simulation results based on Algorithm 3.1 are compared with calculation results from Equation 4.13 as plotted in Figure 10. We can observe that the two results are close to each other, which demonstrates our calculation works well.

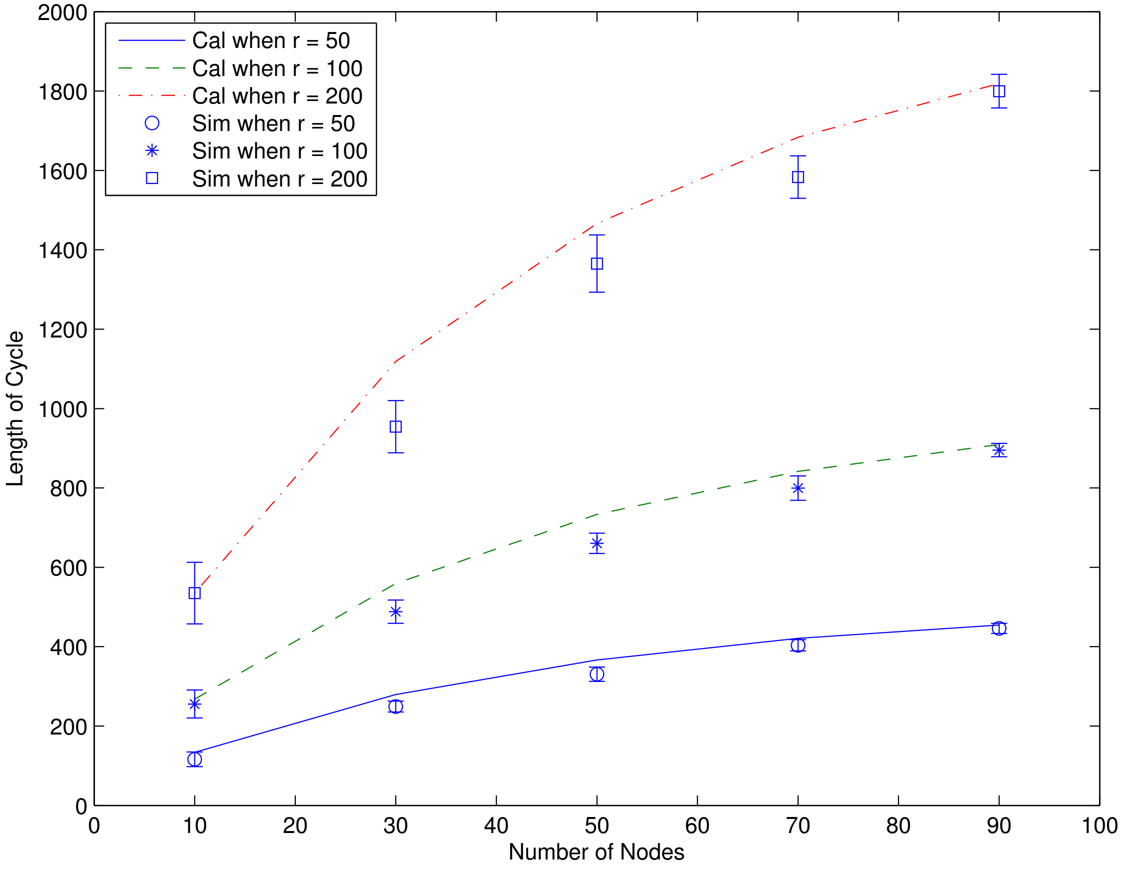


Figure 10. Average Length Comparison between Simulation and Calculation.

CHAPTER V

PERFORMANCE EVALUATION

5.1 Experiment Setup

In order to study the characteristics and evaluate the performance of the PeB scheme, we set up simulation experiment using the ONE simulator [KOK09]. $N = 300$ nodes are uniformly distributed in a field of $2000 \times 2000\ m^2$. The transmission range is 200 meters. Four information servers carrying the same copy of information are randomly distributed throughout the network. More advanced cache placement techniques can be used [MLRR07], but are considered out of the scope of this work. The nodes move with a random way-point mobility model. Queries toward information servers are sent at a normalized rate of one query per second. All simulation results are the average of 20 runs with different seeds and 95% confidence intervals are shown.

We summarize simulation settings in Table 5.

5.2 Performance Metrics

We choose Dynamic Unicast [AWB16] and the scheme by Kim&Ko [KK15] as two state-of-the-art schemes to compare with PeB, by mainly focusing on query and responses (no MAC or cache replacement). Our evaluation focuses on the following metrics: *Overhead*, *Delivery Ratio*, *Delay* and *Overall Efficiency.*

- Overhead, M, is defined as the total number of messages transmitted for each query on an average. Since packet transmission is mostly proportional to en-

Table 5. Simulation Settings

PARAMETERS	VALUES
Simulation time	700 seconds
Number of runs	20
Transmission range	200 meters
Network region	$2000 \times 2000\ m^2$
Number of nodes	300
Number of information servers	4
BA-TTL (TTL of BA messages)	3 hops
LR-TTL (TTL of LR messages)	2 hops
γ	0.2
Announcement frequency, f	0.025
Maximum speed, V_m	5 m/s
Query rate, q	1 query/s

ergy consumption and use of other network resources, such an overhead also represents energy cost to query the information.

- Delivery Ratio, S, is defined as the number of successful queries divided by total number of queries.
- Delay, D, is defined as the sum of time from the moment when a query is generated, to the moment when it is satisfied, divided by total number of successful queries. This metric demonstrates how fast a query can be satisfied.
- Overall Efficiency, Ω, is defined as the number of successful queries divided by number of control packets. This metric demonstrates how efficient the schema can successfully satisfy queries with respect to overhead.

Other evaluation metrics are also possible, such as hop count, but we omit them due to page limit.

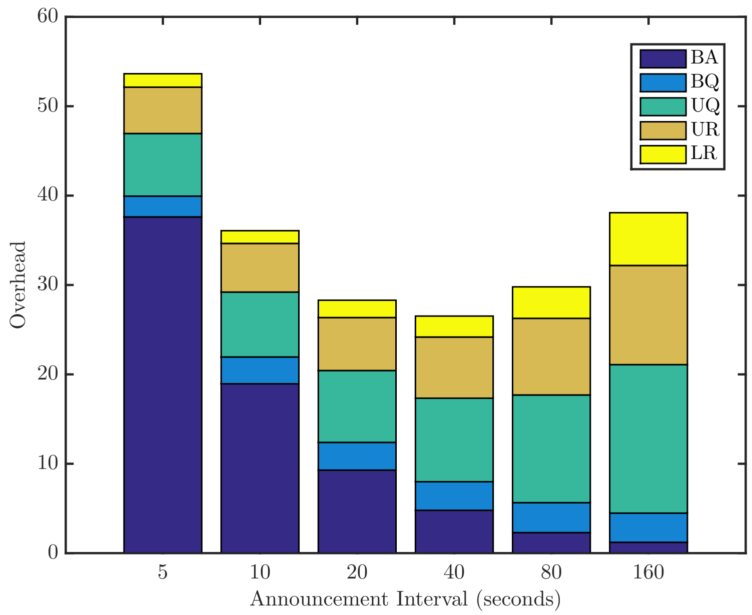


Figure 11. Comparison of Overhead, M, for the PeB scheme with different announcement intervals. The numbers of BA, BQ, UQ, UR, and LR packets are shown. It can be seen that the overall overhead of an announcement interval of 40 seconds is lowest.

5.3 Results and Discussions

The overhead of supporting queries from nodes is presented in Figure 11. In this figure, we show the numbers of BA, BQ, UQ, UR, and LR packets in the PeB scheme with different announcement intervals. When the announcement interval is too high, traces are left outdated, requiring more BQ packets to locate information servers for each query message. On the other hand, short announcement intervals lead to large number of BA packets. The optimum announcement interval can be observed as around 40 seconds. Moreover, the number of LR messages increases when announcement interval increases because of the higher probability of path breaks as indicated in Equation (4.8). We investigate PeB's behavior under different mobility

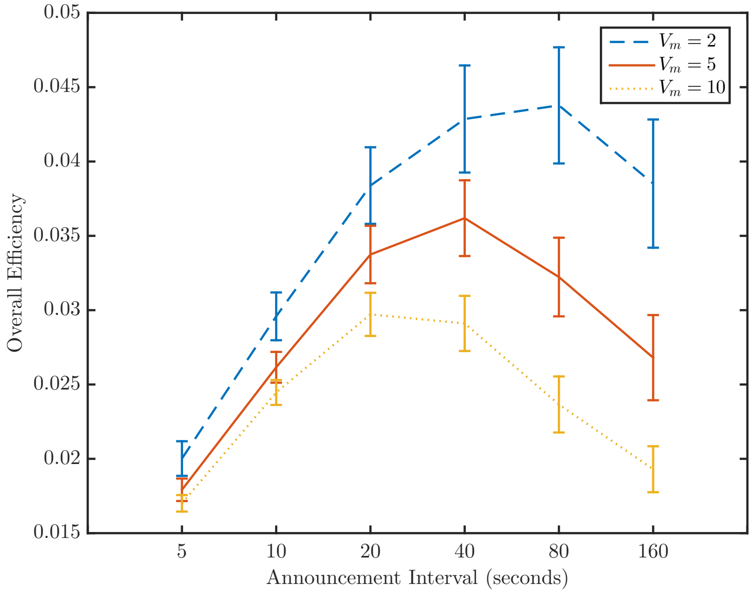


Figure 12. Comparison of Overall Efficiency, Ω, for the PeB scheme with different mobility. As the maximum speed increases from 2 to 10 m/s, the shape of overall efficiency versus announcement interval remains similar. The optimum announcement interval becomes smaller as V_m increases. Furthermore, the overall efficiency of supporting such queries lowers with the increase of V_m.

in Figure 12. With the increase of the maximum speed V_m, from which mobile nodes choose their speed randomly, nodes move faster. From Figure 12, we can observe a natural decrease in overall efficiency as V_m increases, which agrees with the results shown in Figure 8. This is due to higher topology changes caused by higher speed. The best announcement interval can be seen to lower as V_m increases, agreeing with Figure 8 as well. The effect of different announcement scope, BA-TTL, is presented in Figure 13. A large BA-TTL means that the broadcast scope covers a large number of hops (larger region) surrounding the information server. A smaller BA-TTL reduces the scope and the total number of BA messages is reduced. It can be observed that

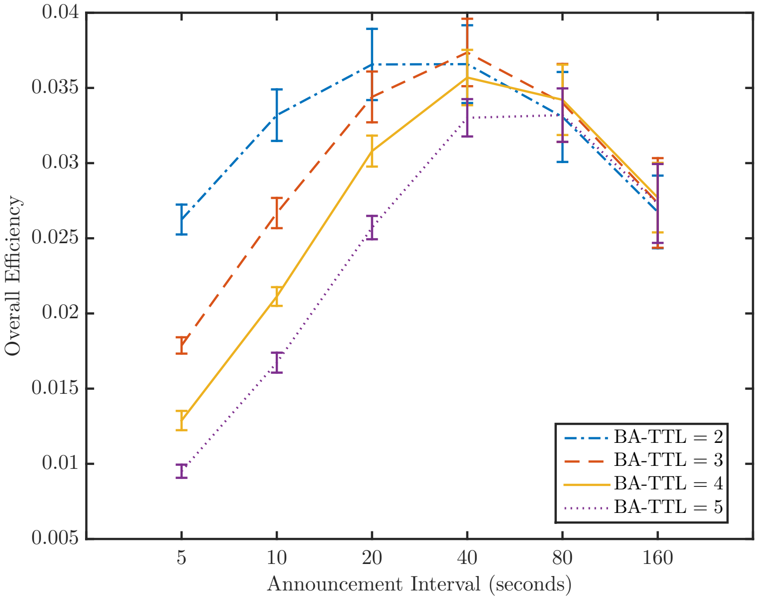


Figure 13. Comparison of Overall Efficiency, Ω, for the PeB scheme with different BA-TTL. This is the scope of the broadcast announcements. An BA-TTL of 3 can be seen to be better than 4 and 5.

the highest overall efficiency is achieved when TTL = 3 for appropriate announcement interval of 40 seconds. Lines become close to each other when interval is 80 and 160, because long announcement intervals make announcement less effective on overhead for all BA-TTL settings. Figure 14 shows the overall efficiency of different BA-TTL under different LR-TTL. In localcast reply approach, LR messages is broadcasted in limited hops. We investigate LR-TTL and BA-TTL in Figure 14 and we can see that when BA-TTL is 3 hops, PeB performs the best in terms of overall efficiency, agreeing with the results shown in Figure 13. Besides, LR-TTL does not have much effect on overall efficiency because, though larger LR-TTL results in more overhead, it contributes to more successful queries as well. We investigate how mobility speed effects overall efficiency under different number of nodes in Figure 15. As maximum mobil-

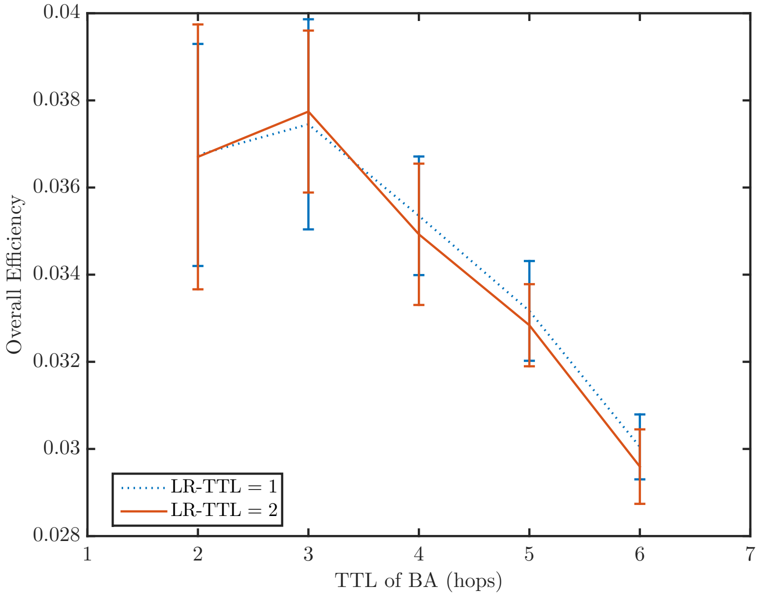


Figure 14. Comparison of Overall Efficiency, Ω, for the PeB scheme with different BA-TTL. A clear optimum of TTL for broadcast announcement of 3 can be observed.

ity speed increases, overall efficiency decreases because higher speed results in higher topology changes, so as path breaks. Denser network topology increases the number of nodes within the scope of announcement, thus more BA messages are duplicated. As a result, overall efficiency decreases. In Figure 16, we investigate the dynamic behaviors of PeB with various settings. In general, we observe an initial increase at the beginning of network operation. This is because of the trace build-up period. Then a more stable delivery ratio can be seen. With a longer announcement interval $(1/f)$, the rate of the initial delivery ratio increase is slower. When systems are running with fewer queries (q), the rate of such an increase is slower too. This is because of the lower chance of "spreading" the trace information toward nodes that are residing outside TTL-hop from the information servers. A similar but naturally reversed trend can be seen in Figures 17 and 18, in which we compare PeB for its overhead

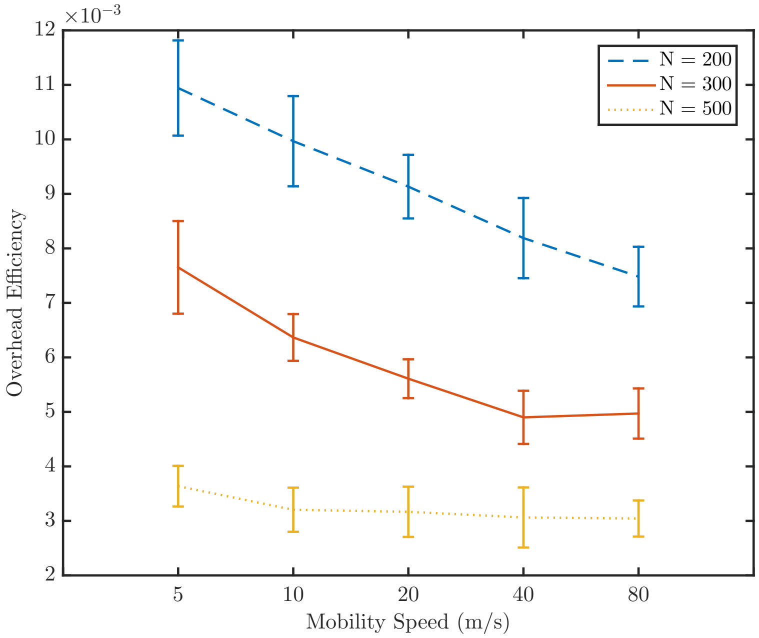


Figure 15. Comparison of Overall Efficiency, Ω, for the PeB scheme with different number of regular nodes.

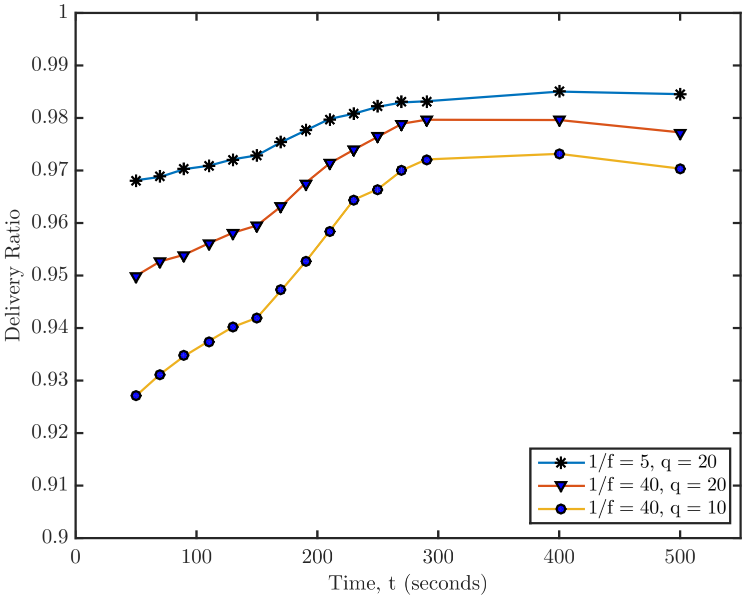


Figure 16. Delivery ratio changes as system run time increases.

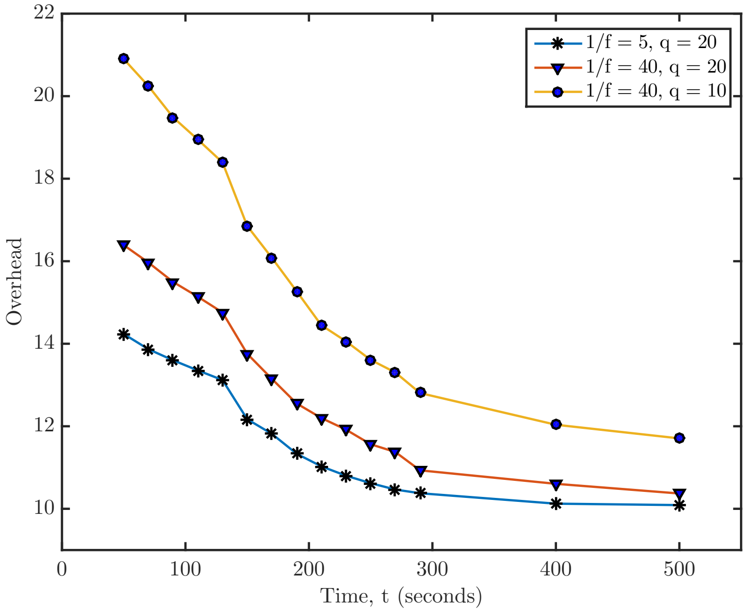


Figure 17. Overhead changes as system run time increases.

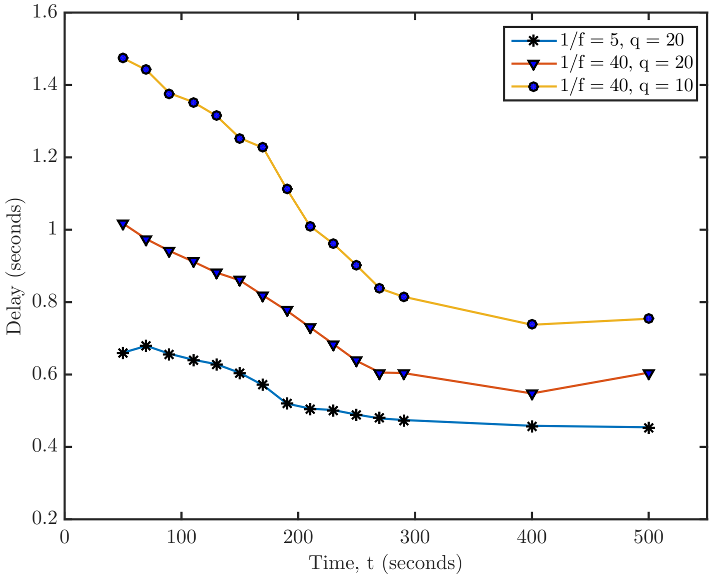


Figure 18. Delay changes as system run time increases.

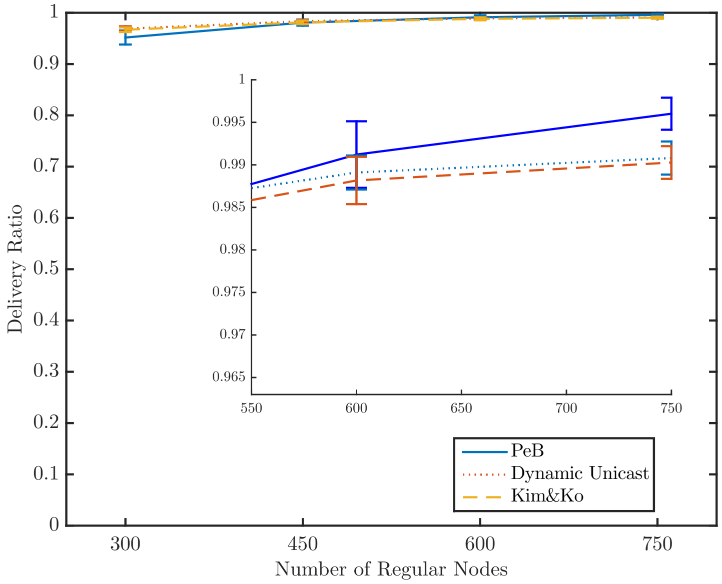


Figure 19. Delivery ratio comparison of different number of regular nodes.

and delay, defined as the time between query submission to response return. We further compare PeB with two state-of-the-art schemes, Dynamic Unicast [AWB16] and Kim&Ko [KK15]. In Figures 19, 20, and 21, we compare delivery ratio, overhead, and overall efficiency among these three schemes, respectively. While all schemes support similar delivery ratios (all increasing with N because of the slightly better connectivity), PeB operates with a much lower overhead (about 20 times lower than Dynamic Unicast and 50 times lower than Kim&Ko scheme). This is more clearly in Figure 21, in which the overall efficiency of the PeB scheme is at least one order better than the Dynamic Unicast and Kim&Ko schemes. A slight increase of overhead and decrease of overall efficiency can be observed as N increases. These should have been caused by the increased cost of maintaining the traces (or broadcast behaviors in the three schemes). Figure 22 shows overall efficiency comparison under different query rates. As query rate increases, traces become more accurate in the PeB scheme, increas-

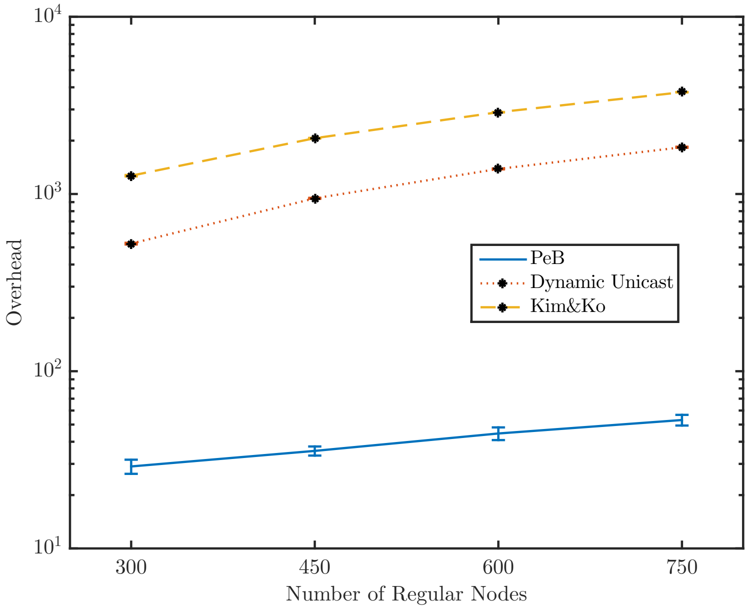


Figure 20. Overhead comparison of different number of regular nodes.

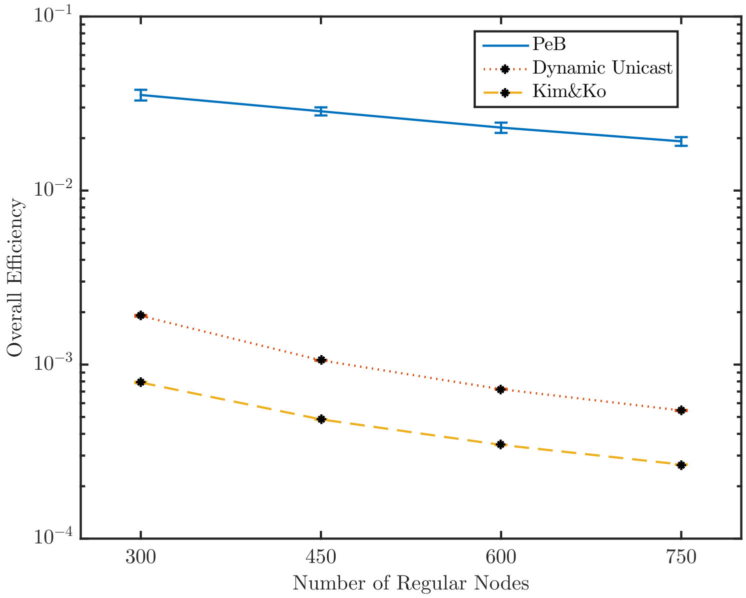


Figure 21. Overall Efficiency comparison of different number of regular nodes.

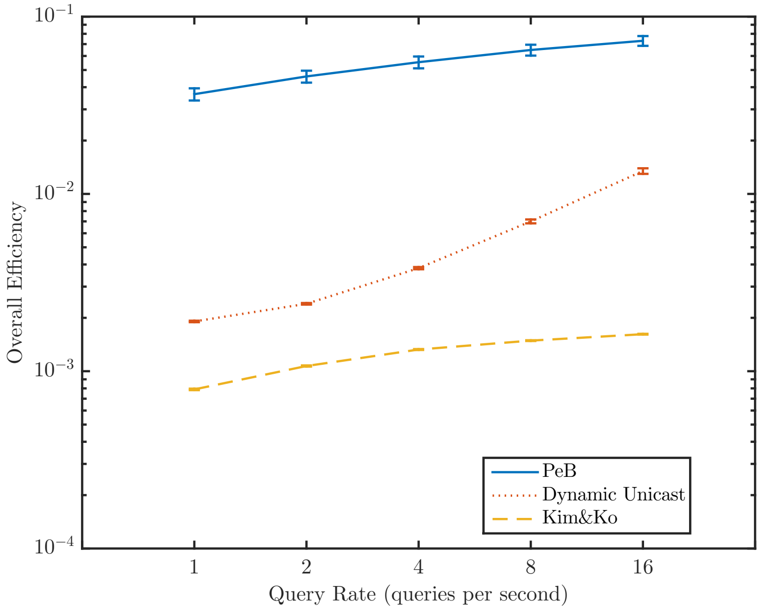


Figure 22. Overall Efficiency comparison of different query rate.

ing its delivery ratio (and overall efficiency Ω). The faster increase of the Dynamic Unicast might have been caused by the increasing number of DATA messages that update the FIB more frequently. Kim&Ko behaves similarly as PeB because the EFS messages are also broadcasted periodically.

CHAPTER VI

CONCLUSION AND FUTURE WORKS

Large wireless networks plays more and more important roles because Internet of Things (IoT) apprears much more frequently in people's daily life. Information distribution and energy concern of such networks are two of those many challenges. In this work, we have investigated these two problems and also introduced two approaches, *PeB* and *Quad*.

PeB is based on the swarm intelligence paradigm. Besides having advantages inherited from swarm intelligence, such as local information possession that does not cause extra overhead of interaction with environment and lowered transmission number in network, our approach announces trace information on those potentially optimal paths that can yield better performance in overhead and delivery ratio, allowing querying nodes to locate requested information more quickly and efficiently both in static and dynamic networks. We also proposed a novel localcast strategy, *Localcast Reply*, that allows reply messages to find returning paths even when there exist broken links. An analysis model has been developed to estimate the best announcement frequency for different network scenarios and confirm it with simulations. Simulation results has shown that PeB works better than two other state-of-the-art schemes.

In the future, PeB can be used for data dissemination and query in other mobile wireless networks such as content-centric networks. Instead of searching for certificates, other essential data or information can be queried.

Quad is a Quadtree-based heuristic algorithm that generates Hamiltonian cycle for WCV to visit those devices that run out of battery. When devices run out of battery, they become static and wait to be recharged. Quad aims to schedule a route for WCV to visits those devices efficiently and finally go back to the starting point. We also gave a distribution of length of such route so that we could know how many WCV we should prepare and how much batteries a WCV should carry. Our numerical results and simulation results are close to each other, meaning that the developed distribution model is precise.

In the future, we will extend Quad to the scenario where there are multiple WCVs ready to function at the same time. We will also improve our distribution model, making it suitable to the multiple WCVs situation and more general. For example, rather than divide a block into 4 ones, we could make it 9 or 16, which may be more efficient and interesting.

REFERENCES

[ABCC11] David L Applegate, Robert E Bixby, Vasek Chvatal, and William J Cook, *The traveling salesman problem: a computational study*, Princeton university press, 2011.

[Aro98] Sanjeev Arora, *Polynomial time approximation schemes for euclidean traveling salesman and other geometric problems*, Journal of the ACM (JACM) **45** (1998), no. 5, 753–782.

[AWB16] Carlos Anastasiades, JÃijrg Weber, and Torsten Braun, *Dynamic unicast: Information-centric multi-hop routing for mobile ad-hoc networks*, Computer Networks **107, Part 2** (2016), 208 – 219, Mobile Wireless Networks.

[BI95] D. Barbara and T. Imielinski, *Sleepers and workaholics: caching strategies in mobile environments (extended version)*, The VLDB Journal **4** (1995), 567–602.

[CH05] Z. Cheng and WB. Heinzelman, *Flooding strategy for target discovery in wireless networks*, Wireless Networks **11** (2005), 607–618.

[DGV09] Yu Du, Sandeep K.S. Gupta, and Georgios Varsamopoulos, *Improving on-demand data access efficiency in manets with cooperative caching*, Ad Hoc Networks **7** (2009), no. 3, 579 – 598.

[DKTM11] Nikos Dimokas, Dimitrios Katsaros, Leandros Tassiulas, and Yannis Manolopoulos, *High performance, low complexity cooperative caching for wireless sensor networks*, Wirel. Netw. **17** (2011), no. 3, 717–737.

[DQA04] Anwitaman Datta, Silvia Quarteroni, and Karl Aberer, *Autonomous gossiping: A self-organizing epidemic algorithm for selective information dissemination in wireless mobile ad-hoc networks*, Semantics of a Networked World. Semantics for Grid Databases (Mokrane Bouzeghoub, Carole Goble, Vipul Kashyap, and Stefano Spaccapietra, eds.), Lecture Notes in Computer Science, vol. 3226, Springer Berlin Heidelberg, 2004, pp. 126–143.

[DRT09] Floriano De Rango and Mauro Tropea, *Swarm intelligence based energy saving and load balancing in wireless ad hoc networks*, Proceedings of the 2009 workshop on Bio-inspired algorithms for distributed systems (New York, NY, USA), BADS '09, ACM, 2009, pp. 77–84.

[GdWFM02] M. Gerharz, C. de Waal, M. Frank, and P. Martini, *Link stability in mobile wireless ad hoc networks*, Local Computer Networks, 2002. Proceedings. LCN 2002. 27th Annual IEEE Conference on, nov. 2002, pp. 30 – 39.

[GP08] Alfredo Garcia and Fernan A. Pedraza, *Rational swarm routing protocol for mobile ad-hoc wireless networks*, Proceedings of the 5th international conference on Pervasive services (New York, NY, USA), ICPS '08, ACM, 2008, pp. 21–26.

[HH09] E. Y. Hua and Z. J. Haas, *An algorithm for prediction of link lifetime in manet based on unscented kalman filter*, IEEE Communications Letters **13** (2009), no. 10, 782–784.

[HHW10] Ren-Hung Hwang, Cheng-Chang Hoh, and Chiung-Ying Wang, *Swarm intelligence-based anycast routing protocol in ubiquitous networks*, Wireless Communications and Mobile Computing **10** (2010), no. 7, 875–887.

[HWH06] Cheng-Chang Hoh, Chiung-Ying Wang, and Ren-Hung Hwang, *Anycast routing protocol using swarm intelligence for ad hoc pervasive network*, Proceedings of the 2006 international conference on Wireless communications and mobile computing (New York, NY, USA), IWCMC '06, ACM, 2006, pp. 815–820.

[IGE$^+$03] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva, *Directed diffusion for wireless sensor networking*, IEEE/ACM Trans. on Networking **11** (2003), no. 1, 2–16.

[JB04] Tao Jiang and John S Baras, *Ant-based adaptive trust evidence distribution in manet*, Distributed Computing Systems Workshops, 2004. Proceedings. 24th International Conference on, IEEE, 2004, pp. 588–593.

[JH08] Y.-J. Joung and S.-H. Huang, *Tug-of-war: An adaptive and cost-optimal data storage and query mechanism in wireless sensor networks*, Proc. of the 4th IEEE international conference on Distributed Computing in Sensor Systems (DCOSS âĂŹ08) (Berlin, Heidelberg), Springer-Verlag, 2008, pp. 237–251.

[JW05] Shudong Jin and Limin Wang, *Content and service replication strategies in multi-hop wireless mesh networks*, Proceedings of the 8th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems, ACM, 2005, pp. 79–86.

[KDT10] Y. Kong, J. Deng, and S. R. Tate, *A distributed public key caching scheme in large wireless networks*, Proc. of IEEE Global Telecommunications Conference - Communication and Information System Security (GLOBECOM '10) (Miami, FL, USA), December 6-10 2010, pp. 1–5.

[Ken06] J. Kennedy, *Handbook of nature-inspired and innovative computing*, Springer US, 2006.

[KK15] Dabin Kim and Young-Bae Ko, *A novel message broadcasting strategy for reliable content retrieval in multi-hop wireless content centric networks*, Proceedings of the 9th International Conference on Ubiquitous Information Management and Communication (New York, NY, USA), IMCOM '15, ACM, 2015, pp. 108:1–108:8.

[KLS16] Michael Krivelevich, Choongbum Lee, and Benny Sudakov, *Compatible hamilton cycles in random graphs*, Random Structures & Algorithms (2016), n/a–n/a.

[KOK09] Ari Keränen, Jörg Ott, and Teemu Kärkkäinen, *The ONE Simulator for DTN Protocol Evaluation*, SIMUTools '09: Proceedings of the 2nd International Conference on Simulation Tools and Techniques (New York, NY, USA), ICST, 2009.

[LLPG10] Uichin Lee, Jiyeon Lee, Joon-Sang Park, and M. Gerla, *Fleanet: A virtual market place on vehicular networks*, Vehicular Technology, IEEE Transactions on **59** (2010), no. 1, 344 –355.

[LLYG14] You Lu, Xiao Li, Yu-Ting Yu, and Mario Gerla, *Information-centric delay-tolerant mobile ad-hoc networks*, Computer Communications Workshops (INFOCOM WKSHPS), 2014 IEEE Conference on, IEEE, 2014, pp. 428–433.

[LT10] Nyree Lemmens and Karl Tuyls, *Stigmergic landmark routing: A routing algorithm for wireless mobile ad-hoc networks*, Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation (New York, NY, USA), GECCO '10, ACM, 2010, pp. 47–54.

[LW02] C. Lindemann and O. P. Waldhorst, *A distributed search service for peer-to-peer file sharing in mobile applications*, Proceedings. Second International Conference on Peer-to-Peer Computing,, 2002, pp. 73–80.

[LWN⁺16] Xiao Lu, Ping Wang, Dusit Niyato, Dong In Kim, and Zhu Han, *Wireless charging technologies: Fundamentals, standards, and network applications*, IEEE Communications Surveys & Tutorials **18** (2016), no. 2, 1413–1452.

[Mar83] Silvano Martello, *Algorithm 595: An enumerative algorithm for finding hamiltonian circuits in a directed graph*, ACM Transactions on Mathematical Software (TOMS) **9** (1983), no. 1, 131–138.

[MDO⁺10] Sudip Misra, Sanjay K. Dhurandher, Mohammad S. Obaidat, Pushkar Gupta, Karan Verma, and Prayag Narula, *An ant swarm-inspired energy-aware routing protocol for wireless ad-hoc networks*, J. Syst. Softw. **83** (2010), 2188–2199.

[MLRR07] H. Miranda, S. Leggio, L. Rodrigues, and K. E. E. Raatikainen, *An algorithm for dissemination and retrieval of information in wireless ad-hoc networks*, Proc. of the 13th International Euro-Par Conference (France), 2007, pp. 891–900.

[NSC03] P. Nuggehalli, V. Srinivasan, and CF. Chiasserini, *Energy-efficient caching strategies in ad hoc wireless networks*, Proc. of the 4th ACM international symposium on Mobile ad hoc networking & computing, 2003, pp. 25–34.

[NTS99] S. Ni, Y. Tseng, and J. Sheu, *The broadcast storm problem in a mobile ad hoc network*, Proc. of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom '99) (Seattle, WA, USA), August 1999, pp. 152–162.

[RLX14] X. Ren, W. Liang, and W. Xu, *Maximizing charging throughput in rechargeable sensor networks*, 2014 23rd International Conference on Computer Communication and Networks (ICCCN), Aug 2014, pp. 1–8.

[Sam84] Hanan Samet, *The quadtree and related hierarchical data structures*, ACM Computing Surveys (CSUR) **16** (1984), no. 2, 187–260.

[SJ05] Chien-Chung Shen and Chaiporn Jaikaeo, *Ad hoc multicast routing algorithm with swarm intelligence*, Mob. Netw. Appl. **10** (2005), 47–59.

[SW04] Prince Samar and Stephen B. Wicker, *On the behavior of communication links of a node in a multi-hop mobile environment*, Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing (New York, NY, USA), MobiHoc '04, ACM, 2004, pp. 145–156.

[VES⁺13] Matteo Varvello, Jairo Esteban, Mark Smith, Lloyd Greenwald, Yang Guo, and Mary Schurgot, *Range queries in manet: Can information-centric networking help?*, 2013 22nd International Conference on Computer Communication and Networks (ICCCN), IEEE, 2013, pp. 1–6.

[XHLL04] Jianliang Xu, Qinglong Hu, Wang-Chien Lee, and Dik Lun Lee, *Performance evaluation of an optimal cache replacement policy for wireless data dissemination*, IEEE Transactions on Knowledge and Data Engineering **16** (2004), 125–139.

[XSH⁺15] Liguang Xie, Yi Shi, Y Thomas Hou, Wenjing Lou, Hanif D Sherali, and Scott F Midkiff, *Multi-node wireless energy charging in sensor networks*, IEEE/ACM Transactions on Networking (ToN) **23** (2015), no. 2, 437–450.

[XSHS12a] Liguang Xie, Yi Shi, Y. Thomas Hou, and Hanif D. Sherali, *Making sensor networks immortal: An energy-renewal approach with wireless power transfer*, IEEE/ACM Trans. Netw. **20** (2012), no. 6, 1748–1761.

[XSHS12b] Liguang Xie, Yi Shi, Y Thomas Hou, and Hanif D Sherali, *Making sensor networks immortal: An energy-renewal approach with wireless power transfer*, IEEE/ACM Transactions on networking **20** (2012), no. 6, 1748–1761.

[YLC11] Fan Ye, Qing Li, and Enhong Chen, *Benefit based cache data placement and update for mobile peer to peer networks*, World Wide Web **14** (2011), 243–259 (English).

[YZS11] Qiong Yang, Jun Zheng, and Lianfeng Shen, *Modeling and performance analysis of periodic broadcast in vehicular ad hoc networks*, Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE, 2011, pp. 1–5.

[ZCP07] Wensheng Zhang, Guohong Cao, and Tom La Porta, *Data dissemination with ring-based index for wireless sensor networks*, IEEE Transactions on Mobile Computing **6** (2007), 832–847.

[ZZCD10] Jing Zhao, Ping Zhang, Guohong Cao, and C.R. Das, *Cooperative caching in wireless p2p networks: Design, implementation, and evaluation*, Parallel and Distributed Systems, IEEE Transactions on **21** (2010), no. 2, 229 –241.