DESIGN AND ANALYSIS OF PLACE BASED OPPORTUNISTIC NETWORKS

by

YANLIANG LIU

Presented to the Faculty of the Graduate School of

The University of Texas at Arlington in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT ARLINGTON

May 2015

To my mother Ming Wan and my father Jianguo Liu
who support me fully during my whole academic life.

ACKNOWLEDGEMENTS

I would like to thank my supervising professor Dr. Yonghe Liu for constantly motivating and encouraging me, and also for his invaluable advice during the course of my doctoral studies. I wish to thank my academic advisors Dr. Mohan Kumar, Dr. Gergely Zaruba and Dr. David Kung for their interest in my research, for taking time to serve in my dissertation committee and for their continuous feedbacks during my Ph.D. studies.

I would also like to extend my appreciation to Dr. Na Li for her helpful discussions and invaluable comments during my doctoral journey. I wish to thank Dr. Hao Che and Dr. Maio Ju for his inspiring advice during the theoretical study of network capacity. I would like to express my thanks to Dr. Shaoshu Sha and Shibo Yu for their sincere help during my dissertation preparation.

I am grateful to all the teachers who taught me during the years I spent in school, first in Nanping China, then in Guangzhou China and finally in the Arlington TX Unites States. I would like to thank my uncle Yaohui Tian and aunt Guiping Liu for encouraging and inspiring me to pursue graduate studies.

Finally, I would like to express my deep gratitude to my mother and father for their sacrifice, encouragement and patience during these years. I feel extremely fortune to meet so many good friends after I attended UT Arlington. Thanks for all the people who help and support me in my career.

April 17, 2015

ABSTRACT

DESIGN AND ANALYSIS OF PLACE BASED OPPORTUNISTIC NETWORKS

Yanliang Liu, Ph.D.

The University of Texas at Arlington, 2015

Supervising Professor: Yonghe Liu

The swift growth and popularization of wireless technology and mobile smart phones have unfolded various opportunities and challenges before researchers' eyes. Opportunistic networks is one of the interesting and challenging topics. Generally speaking, opportunistic networks exploit the potential capability of existing mobile devices carried by people to provide pervasive computing service, such as data forwarding, without pre-planted infrastructures. The movement of mobile devices introduced by human activity plays a crucial role in the functionality of opportunistic networks, since it influences the contacts between different mobile devices. It is well-known that human movement presents these place-centered features: intermittent hops between places and stops at places. And people spend most of their time in different places every day, and their movement inside places are more stable than movement between places. The above relatively stable human activity in each place could provide longer contact time and higher contact possibility between mobile devices. Based on this observation, we propose a new opportunistic network scenario named place based opportunistic networks. In this new type of networks, data forwarding is assumed only to take place in each place to cope with above features.

The purpose of this work is to discuss and study various topics in the proposed place based opportunistic networks, from basic theoretical understanding of place based opportunistic networks to potential applications operating upon these networks. We mainly focused on the following topics: 1) Localization in Place Based Opportunistic Networks. Localization is a required functionality in place based opportunistic networks. It identifies the location of each mobile devices. We proposed a new localization scheme named COAL that takes advantage of surrounding context information in order to reduce the energy consumed by localization service without losing accuracy. 2) Capacity of Place Based Opportunistic Networks. Capacity is a classic and important topic of every network. It indicates the amount of data could be served by the network. Briefly speaking, we proposed a two-layer model to represent this network and solved a maximum flow problem on this network to obtain capacity. 3) Routing in Place Based Opportunistic Networks. Besides capacity, routing is another important topic in network research field and attracts attention from plenty of researchers. Based on the inherent features of place based opportunistic networks, we designed two routing schemes based on popularity and congestion information separately, and setup experiments to compare the performance of several routing schemes. 4)Application Recommendation System in Place Based Opportunistic Networks. As current high penetration of mobile phones, a huge data pool could be built up based on the data sensed and stored by each mobile phone. We proposed to build up an application recommendation system based on these data pool. Mathematical models have been proposed to relate applications and places as well as quantify the attention reward gained by executing each application. An approximate greedy heuristic algorithm and a dynamic algorithm have been designed to compute application recommendation lists. Both simulation and field study showed the feasibility of our proposed recommendation system.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

CHAPTER 1

INTRODUCTION

The first decade of 21th century has witnessed the blossom of wireless communication and mobile devices. Smart phones and personal tablets widely spread and deeply penetrate into every aspect of human daily life. By Q3 2010 [2], the total number of mobile devices purchase has already surpassed the number of personal computers purchase, and this trend still keeps on skyrocketing. These enormous and ubiquitous mobile devices and consequently formed wireless networks introduce plenty of interesting research topics and attract a great many researchers to devote their efforts on those topics. Opportunistic networks is one of the meaningful and challenging topics among them. In this chapter, we begin with briefly introducing the concept and current researches of opportunistic networks. After this, we reveal the main idea and challenges of place based opportunistic networks. In the end, we conclude this chapter with our contributions on place based opportunistic networks.

1.1 Opportunistic Networks

Ubiquitous personal mobile devices and advanced wireless technology have open a gate to a promising network, opportunistic networks, and welcome new opportunities into our everyday life. Opportunistic networks exploit the potential capability of existing mobile devices carried by people to provide pervasive computing service like data forwarding without pre-planted infrastructures. This newborn network setting catalyzes the births of plenty of interesting research topics [3, 4, 5] and real life applications.

Unlike conventional internet and cellular networks that base on and utilize pre-planted infrastructures, in the scenario of opportunistic networks infrastructures usually play an insignificant role, such as backbone network topology built upon them. Lack of pre-known network topology is one of the consequences of this situation; and the other one is the pure peer to peer ad-hoc communication style. Both of them present challenges in front of researchers, and the inherent mobility of mobile devices introduced by human movement makes these challenges even more difficult and interesting.

First of all, lack of pre-existing routing paths place the first challenge on opportunistic networks. In conventional networks, thanks to the pre-planted infrastructures, routing paths from source to destination are usually computed at the beginning of whole forwarding process. The fact that no pre-calculated paths exist means that the routing paths have to be decided simultaneously while the routing is taking place. This real-time online process means that a lot of previous routing schemes and protocols in conventional networks fail to play a part under the scenario of opportunistic networks, such as TCP acknowledge mechanisms. New design strategies and new routing schemes should be proposed to meet the requirements of opportunistic networks. Currently, the main strategy deployed by most of routing schemes in opportunistic networks is to leverage local knowledge provided by nearby mobile devices and then forward the data to the ones that have higher possibility to get closer to the destination [6, 7].

Besides that, one of the crucial problems that arise from opportunistic networks is how to capture the basic features of it, such as capacity. The pure peer to peer ad-hoc communication style and dynamic network topology caused by human movements make it hard to apply conventional methods to opportunistic networks. Some pioneering works on wireless sensor networks [8, 9] have discovered methods in their scenarios. However, they still fail to be directly applied to opportunistic networks because of the inherent human activity centered mobile feature of opportunistic networks. What's more, due to the vast kinds of human movement patterns, the consequent various mobility patterns of mobile devices may require different mathematical models and hence generate different results. Besides above theoretical issues, the real applications of opportunistic networks have been questioned and argued in the academic community for a long time because of their limited practical value. By now the disaster, military and rural environments are the most mentioned scenarios where opportunistic networks could actually play indispensable roles, since their background environments are naturally infrastructure-less. However, the number of mobile devices in these environments are usually small compared to the amount of mobile devices in urban areas where the true potential of opportunistic networks may present. In recent years, the enormous amount of efforts has been contributed to the area of opportunistic networks. Based on these previous research efforts and results, we propose a new scenario of opportunistic networks namely place based opportunistic networks, and will discuss it in the following sections.

1.2 Place Based Opportunistic Networks

In opportunistic networks, mobility is inherent for mobile devices and provides the underpinnings for data backhauling; besides this, we could observe that human movement is intermittent and associated mobile devices will likely stop at a place for some time before continuing. At these places, mobile devices are relatively stationary, which makes the contact periods much longer and contact opportunities more predictable, as compared to those during movement. The new place based opportunistic networks we proposed are illustrated in Fig.1.1. In order to leverage the above human activity centered features of the new networks, data exchanges are assumed only to take place in each place in the networks; and links between two places exist when people move from one place to another during certain period. From these prerequisites, it is obvious that human daily schedules determine where these data exchange could take place. Besides that, these new networks show time varying features in different time during a day, such as the large population during lunch time in restaurants and during evenings in entertainment places. What's more, since the connectivity of different places is created by human daily movement between places, network topology is also determined by human activity. For example, more links from residences to offices during the sunrise time and more links from offices to residences during the sunset time.

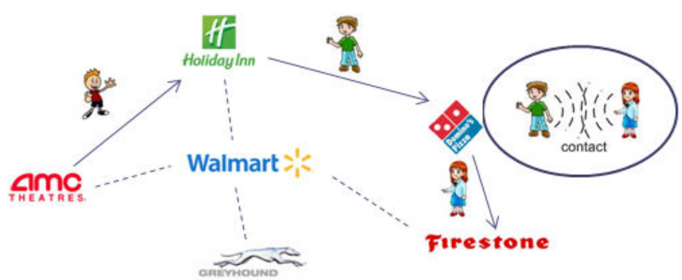


Figure 1.1. Place Based Opportunistic Networks Illustration.

This new highly human activity centered network scenario places several interesting topics in front of us to dig into, which both exist or not exist in conventional opportunistic networks. Firstly, we should know how to enable this network or make it function. For example, it is crucial to identify whether a mobile device is in a place or not in order to make sure data exchange not to happen anywhere. Secondly, to understand the basic characters like capacity of this network could benefit us and later researchers to utilize it better in future. Thirdly, data forwarding, as the key service of opportunistic networks, will be the same topics in this new network. Although the new network possesses most of the characteristics of conventional opportunistic networks, whether the routing schemes in this new network will have some divergences from classic routing schemes or not is an interesting question. Finally, building up real applications upon this network is the same topic as in conventional opportunistic networks, and those applications could better exhibit the usefulness of the new network.

1.3 Contributions

From the above discussions, we divide our research work into 4 research parts to study: (1)localization, the key enabling functionality of place based opportunistic networks; (2)capacity, the basic characteristic of place based opportunistic networks; (3)routing design, data forwarding service provided by place based opportunistic networks; (4)applications recommendation system, potential applications built upon place based opportunistic networks. From theoretical study to practical application, we hope to better understand this new network scenario. The followings are our current achievements:

1.3.1 Energy Efficiency Localization in Place Based Opportunistic Networks

In this part, we proposed a novel localization scheme termed COAL (for COntext Aware Localization) to target at both energy efficiency and accuracy. Our key idea was to leverage users' context information such as an ongoing event to facilitate the localization scheme. By employing these context information, we could significantly reduce localization frequency to conserve energy while maintaining the high degree of accuracy. Noteworthily, our scheme is complementary to existing location schemes using any wireless signal and can be implemented readily as their enhancement. We implemented our scheme in iPAQ smart phones using WiFi signals and performed extensive field studies. The results showed that the scheme can save energy up to 90 percent without sacrificing too much accuracy.

1.3.2 Capacity of Place Based Opportunistic Networks

In this part, we focused on investigating the capacity of place based opportunistic networks. Owing to the opportunistic nature of mobile device contacts and time varying nature of human activity, the capacity of both places and links are time varying. We proposed a two-layer model to derive the capacity of place based networks. In the first layer, a mixed form queueing network model was constructed to compute the population in each place and also the visitor arrival rate as well as visiting probability to each place. After this, in the second layer we transformed the results from the first layer into node capacity by leveraging the results from [8] and derived link capacity. Finally, based on these two results, we solved a time-varying maximum flow problem to obtain the network capacity.

1.3.3 Routing of Place Based Opportunistic Networks

In this part, our contributions mainly include the following two aspects: 1. Based on new place-based opportunistic networks scenario we proposed in our previous work, we presented two new routing schemes specifically for this new network scenario. One deployed popularity indicator and the other one utilized congestion information to form a competitive game. Both of the routing schemes worked well in new network scenario. 2. We designed experiments to run several routing schemes in this new network. And based on comparisons on the results, we showed some insights to design routing schemes for this new network scenario.

1.3.4 Applications Recommendation in Place Based Opportunistic Networks

In this part, we proposed and designed an application scenario based on place based opportunistic networks: applications recommendation system for suggesting new applications and relieving waiting anxiety. Our contributions include: 1) We proposed a new application recommendation system that utilizes human activity information in different places in a city, which has not been studied in other works or existing applications recommendation systems. This system included place/application matching model and two applications list recommending algorithms. 2) We implemented the recommendation application on real mobile phones and conducted various field studies to show the feasibility of our recommendation system. 3) This system provided application developer an new option to quickly and conveniently advertise their new applications instead of the long-waiting process through App Stores. Besides that, our new application recommendation system was a distributed one. It provided fault tolerance and privacy protection over the centralized design.

CHAPTER 2

LITERATURE

In this chapter, we will discuss the related research works on localization, routing, capacity and recommendation systems. Basically, we briefly review the contributions of previous works and discuss the relationship between these works and our works.

2.1 Localization

An extensive set of localization schemes have been developed in the literature [10, 11, 12, 13, 14, 15]. In general, these studies can be classified into two categories: radio signal strength indicator (RSSI) based [16, 17, 15] and background information combination based [18].

Traditional RSSI based approaches perform triangulation based on signal-distance conversion. More advanced schemes often use fingerprinting [15] to increase accuracy. In these schemes, signal pattern in a place is collected beforehand and pattern matching is performed during runtime to pinpoint a person's location. Obviously significant efforts are needed to collect and update these data [19, 20, 14]. Regardlessly, these schemes often require intensive periodical localization operation, which can drain precious battery energy from a mobile device. Besides that, approaches employing background information often take advantage of additional sensory data such as sound and light to identify a location [21]. Learning techniques are often used to train the system and increase accuracy. Examples include EEMSS [10] and CenceMe [22]. Besides above two main categories, some other works also proposed interesting localization methods. In [23], they leveraged the channel state information of OFDM systems instead of RSSI to pinpoint indoor location.

A subset of existing works has specifically focused on high accuracy with constraint on energy consumption [10, 12]. Combination of multiple sensors is a common approach in this regard, where different sensors are chosen to handle different situation in order to conserve energy. The accuracy of this approach relies on how to optimally assign different sensors in different places. It often employs either history trace data or ambiance fingerprinting, such as background sound. Different from them, our localization scheme will use the basic RSSI and contextual event information to conserve valuable battery energy and achieve room-level accuracy at the same time.

2.2 Capacity

Capacity of computer networks has been a classic subject of extensive networking research [24, 8, 9, 25, 26, 27]. The most well-known and fundamental achievement is the capacity of a weighted network that was discussed in [24]. It has been shown that this problem can often be transformed into a linear programming problem and solved by the simplex algorithm. Capacity of static ad-hoc wireless network was investigated in [8], followed by further work on that of mobile and mix-form wireless networks [9, 25]. Several papers also discussed the capacity of opportunistic networks. For example, it was shown in [26] that contact time and inter-contact time are two important factors to affect the capacity of opportunistic networks. Capacity of an urban transportation networks was discussed in [27] and obtained by solving linear programming problem with the help of available origin-destination (O-D) demand pattern of transportation network. Besides that, [28] proposed to combine a stochastic model and encounter duration analysis to try to induce the space-time capacity of a given delay tolerant network.

In place centered opportunistic networks, mobile devices only exchange data when they encounter each other inside a place. This is different from conventional opportunistic networks whose focus is often on transient contacts, and we proposed a new way to compute the capacity of the place centered opportunistic networks.

2.3 Routing

Routing has always been an interesting and challenging topic in networking research area, which mainly handles how to select paths in a network in order to better send data traffic from source to destination. [29] presented one of the most famous and fundamental algorithms that computes the shortest path from one node to another in a network. However, shortest path is usually not the only concern during route selection. Quality of service(QoS), such as congestion and fairness, is also an important issue when designing routing schemes. Better QoS could satisfy other requirements of the network. In another way, those QoS-oriented routing schemes can be thought as finding shortest paths that satisfy some predefined constrains. [30] designed an algorithm to make routing decision according to the requirements of multimedia applications. In Mobile Ad-hoc Networks (MANET), energy consumption and bandwidth are also important QoS aspects when designing routing schemes. [31] proposed a Swarm-based Distance Vector Routing scheme based on ant colony optimization to satisfy the delay, jitter and energy constraints together.

Routing in delay tolerant networks(DTN) and opportunistic networks(OPNET) work differently from the conventional networks, since unstable links and long delay in these networks could result in dynamic network topology and consequently affect the route selection. Epidemic based routing scheme is one of the most important routing schemes in DTN [32]. Basically in this type of routing schemes, data are made into several copies and disseminated over the network. Besides that, context information and social information can also be taken into account during routing decision [6, 7]. This extra information could increase the delivery probability, since they could provide data with more opportunity to meet better relay node. We are interested in the routing design in place based opportunistic networks and how various parameters could affect the performance of routing schemes. The followings are some routing schemes we plan to use as references during our experiments.

2.3.1 Geographic-based(Short) Schemes

This is a type of classic routing schemes based on the famous shortest path routing introduced by [29]. Basically, data are transferred from the source to destination by strictly following the computed shortest path. In this kind of routing schemes, partially network information (distance between two nodes) will be utilized, and routing path will be computed from the source to destination based on that information. We decided to deploy the classic shortest path algorithm in our experiments study, data would be transferred to the next place according to the pre-computed shortest path. However, the shortest path routing scheme that we used was slightly different from the original one: when data were carried to a new place, the shortest path should be recomputed since some links might disappear in this dynamic network topology. As we know, human flow in a city changes with time during a day, which causes the link between two places could change by time — either it exists or not. In a result, shortest path for that data needs to be recalculated when it arrives at one place; and only the next hop is useful, due to the fact that the whole network topology could change when data reaches next place. An example is shown in Fig.2.1 to illustrate the dynamic network topology.

Figure 2.1. Time Varying Network.

In this example, data in place A wants to reach place C, the shortest path ($A \rightarrow B \rightarrow C$) obtained in place A is different from the later shortest path ($B \rightarrow D \rightarrow C$) obtained when it arrives at place B due to dynamic network topology.

2.3.2 Flooding-based(Flood) Schemes

Flooding is another classic routing algorithm, which has been well studied in previous works [33]. Basically, flooding schemes have no knowledge about the network and try to make several copies of the data and broad/multicast them to increase the data delivery probability. The good side about flooding scheme is that more data copies make that specific data more possible to be successfully delivered to the destination. On the other side, it introduces inefficiency and high congestion into the network, since same data copies occupy treasurable network resources, which means fewer resources and high drop rate for other data. In later experiments, we chose multi-copy routing algorithms with path finding. Basically fixed number of copies of data would be disseminated to the network [33]. Besides this, same flooding scheme without path precalculation(PureFlood) would also be experimented and compared.

2.3.3 Geographic-congestion-based(Drop) Schemes

Congestion is a common issue in every network; enormous efforts have been contributed on the congestion control topic in wireless network [34, 35]. Briefly, congestion happens when the data arrival rate exceeds the existing network process capability, and in consequence the excessive data have to be dropped. In congestion-based algorithms, congestion information (mainly data drop rate) would be taken into account besides basic path finding computation. Briefly, we found places with paths to the destination, chose the place with the least drop rate as the next hop, and transferred data to pedestrian who would visit this place in his next stop. The good part about this routing schemes is that it prevents a central place to be overwhelmed with too many data and provides more choices for data transfer. However on the other side, it cannot guarantee the shortest path for data, since it treats the congestion information with higher priority than the shortest path.

2.4 Recommendation system and package recommendation

Most common recommendation systems are designed to recommend the results in the form of a list of items to customers, in which there are no special relationships between those items. However, applications that require packages of items in forms of sets or sequences are emerging and revealing new challenges that classic recommendation ways cannot be directly applied to solve them. This kind of applications includes travel package recommendation and courses recommendation.

Some methods were proposed to solve the package recommendation problems through different perspectives. [36, 37] tried to use optimization way to obtain travel packages that satisfy customers' demands. Basically, each item (hotel/ scenic/ transportation) in the travel package is associated with a value (rating) and cost. With the total cost limitation that each customer usually budgets, they proposed to map the travel package problem into a knapsack problem. Based on classic 0/1 knapsack problem, they developed heuristic algorithms to find top-N solutions for each knapsack problem and let user make the final selection. [38] proposed to use multiagent system to find appropriate travel packages to customers. Briefly, they proposed a distributed multiagent recommender system that includes two different kinds of agents: Components agents(CP) are responsible for searching appropriate items by either its own data or communicating with other agents; and travel agents(TA) combine the search results and produce the final recommendations. [39, 40] proposed a model to consider the existing travel package during recommendation reasoning. Basically, they developed tourist-area-season and tourist-relation-area-season topic models to represent travel packages and tourists by different topics distributions, and then proposed cocktail approach (hybrid strategy) to recommend personalized travel packages. Besides that, [41] took some basic concepts from object-oriented programming as reference to facilitate the recommendation process. Briefly, they used objects to represent features and value pairs of travel packages and tourists, and proposed two models: Object-oriented Topic Model was designed to discover the hidden travel interests; and Object-oriented Bayesian Network model was used to infer the co-travel probability between tourists.

Besides recommendation systems, personalization systems and expert systems could also offer suggestions to people. However, in our problem scenario, due to privacy issues, the only available information is the activity information while waiting in different places. With only this information, personalization systems could not come into play, since they aim at providing recommendation to specific individuals instead of a group of people, which requires a lot of context and behavioral data that are missing in our scenario. Expert systems also fail to handle this problem. It is difficult to find related expert knowledge about waiting information in each place and how to give suggestions in place-based opportunistic networks, and those suggestions usually lose the flexibility to let users make choices among several options.

CHAPTER 3

CONTEXT AWARE ENERGY EFFICIENT LOCALIZATION

3.1 COAL: Context Aware Localization Scheme

In this section, we propose a context aware localization scheme namely COAL that leverages context of mobile terminals to facilitate localization and hence achieve energy efficiency and high accuracy. As the context is obtained by the backend server through analyzing public data, neither a prior training data from the field nor its periodic update is necessary. Without losing generality, in the remainder of this chapter, we will use WiFi network to illustrate the proposed scheme. It can be easily extended into any environment with identifiable physical reference such as cellular base stations.

The overall idea of COAL can be illustrated using the following scenario. Suppose a group of students is on campus attending classes and other events. While in-between these events, they are in transient state and their locations are changing, they will likely remain at a specific location while they attend an event. We can use available context information such as seminar schedule and/or a student's personal agenda to facilitate user's localization in addition to any conventional localization method. Furthermore, with the context information, localization can be done in a significantly more energy efficient way and more precise manner. Below, we formally present our assumptions and approaches.

3.1.1 Landmarks and Signal Strength

We assume that a mobile device can receive signals from multiple WiFi access points (APs), denoted by an AP vector

$$AP_{rec} =< AP_{rec1}, AP_{rec2}, ..., AP_{recn} >$$

A corresponding signal strength vector is defined as

$$S_{rec} =< S_{rec1}, S_{rec2}, ..., S_{recn} >$$

where S_{reci} is the received signal strength for AP_{reci}.

While a mobile device can receive signals from a large number of APs in a certain environment and hence introduce a long AP vector, some of the APs can be weak. This usually indicates the APs are not within physical proximity of the user and not reliable for the purpose of localization. We filter the AP list based on the signal strength, reserving those with desirable strong signals. The new access point vector can be denoted as

$$AP_{str} =< AP_{str1}, AP_{str2}, ..., AP_{strn} >$$

and the corresponding new signal strength vector is

$$S_{str} =< S_{str1}, S_{str2}, ..., S_{strn} >$$

With the AP list and signal strength vector, we can localize the mobile device using methods already proposed in the literature, for example triangulation or fingerprinting proximity matching [15]. However, for a known environment such as a campus, we can simplify and enhance the localization process. In our design, we divide the absolute signal strength collected by a mobile device into groups by applying K-Clustering algorithm [42], as shown in Figure 3.1. The result is then used to generate a grouped AP table. Both mean value and deviation of the signal strength are considered.

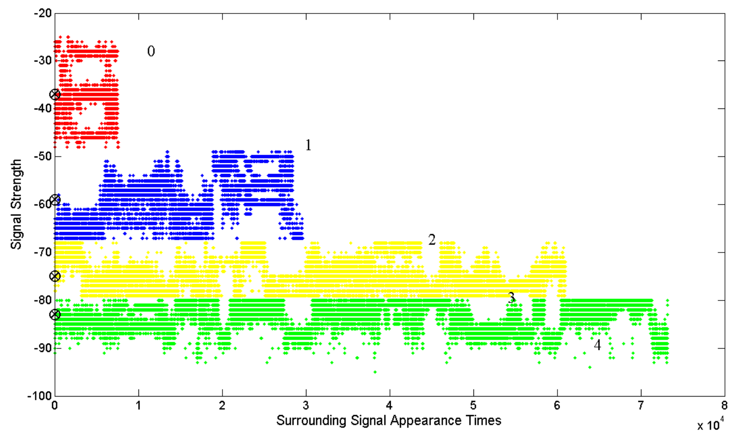


Figure 3.1. Signal Strength of APs in Different Groups.
Based on signal strength, APs are clusterized into different groups (0-4) and used to enhance localization results.

Assume that the received signal strengths of surrounding APs fall into the range n is s^n. Based on our field test in a campus environment, this distribution of the APs signal strength can effectively localize a mobile terminal. For example, if at least one of all signals is in the group s^0, i.e., strongest one, it indicates the mobile is in a closed environment such as a class room. On the other hand, if multiple s^1 signals are present without s^0, it usually indicates the mobile is in an open space within a building such as a hallway. If there are no dominant signals and the strengths are in group s^3 or lower, the mobile is most likely outside a building on campus. The above observations are used to enhance existing localization scheme for example triangulation for more accurate results.

3.1.2 Context Enhancement for Reduced Localization Frequency

Given the list of access points, a plethora of localization schemes can be utilized to determine the location of the mobile terminal. However, to avoid outdated location information, conventional localization schemes have to periodically refresh the access point list, either perform mobile based computation or communicate with localization server for facilitation. These periodic operations can potentially be energy hungry and drain precious battery energy.

In our scheme, we employ user context information to reduce the frequency of these periodic operations and hence conserve energy. We can outline the basic idea using the following example. Assume that a mobile user is attending a seminar. It is highly likely that the user will be in the same location for an extended period. This information of ongoing seminar in the classroom, combined with the seminar's starting and ending time, can be considered as user's context. This context can be used to dramatically reduce localization frequency: during the seminar, the user is highly likely to remain at the same location and localization can be performed at very low frequency; as the seminar is about to end, localization frequency can be increased in order to capture potential user movement.

3.1.2.1 Overall Operation

We first compute a user's location based on received WiFi signal using triangulation and then enhance the localization result (either in a classroom, hallway, or outside a building) using the grouped AP list, as discussed before. If the determined location, together with current time, concur an event such as a class in the system, we will reduce the localization frequency (including rescanning WiFi signals) based on the starting and ending time of the event.

Triangulation for localization is performed on the backend server. Determining the localization frequency, on the other hand, can be performed on both the backend server and mobile device. The advantage of this task partition is two-fold. Localization algorithm usually is computation intensive and energy hungry. Therefore, offloading this task to backend server can conserve energy on the mobile terminal. The backend server will analyze the context information based on the mobile's location and determine proper localization frequency for the mobile. This frequency is then communicated back to the mobile. The mobile will analyze this frequency and enhance or correct it with its own local context information (such as the user's local calendar). For example, if a meeting schedule is in the user's calendar but not considered by the backend server, the mobile can reduce the localization frequency by itself.

3.1.2.2 Context Information Gathering

While numerous research results on context modeling and event prediction, particularly in pervasive computing domain [10, 12, 21], we employ a more straightforward approach in order to focus on the localization part itself. The context information we utilize is gathered from the web and further analyzed at the backend server. This information include public class schedule information available through the university, public personal calenders such as Google/Outlook calendar, or even a seminar announcement available on a mailing list.

Gathering the context information can be performed offline for a particular environment, for example, university campus. It can also be enhanced with real time data analysis [43].

Currently, context information gathering is only implemented at the backend server. The mobile device uses only those directly available locally on the device itself, such as personal calendar.

3.1.2.3 Localization Frequency Determination

Once the context information is obtained and matched to the user's current location, it can be used to determine required localization frequency. Without losing generality, we will use class schedule and classroom as examples for the event and location.

When the context of a user is a class, the backend server will consider the starting and ending time of the class. If the class will last an extended period, for example 1 hour, the user is likely to remain at the same location during this time. A much lower localization frequency can be used here, for example, every $T_1 = 30$ minutes, during this period. As the class is near its end, the frequency can be increased in order to timely detect the user's movement.

3.1.3 Individual Localization Schedule

Although context information such as class schedule can be reliable sources to predict upcoming location changes when an event ends, it is also risky to use it solely as ending of an event is not fixed in real life. The challenge here is to timely detect location changes toward the end of an event.

A trivial solution is to increase the localization frequency of each user toward the end of the event. However, this will result in higher energy consumption. In our design, we assign localization task to individual device in a group gathering *alternatively*: if one user detects a location change, indicating possible end of the gathering, other users will be alerted by the backend server and the localization frequency then will be increased for all. We implement two approaches to assigning localization task to users: random allocation and fixed allocation.

Assume that in a group gathering of n users, localization frequency is relaxed to every T_2 period. In random allocation, the starting time of each user's location operation, denoted by t_s is randomly selected according to the number of total users in this place.

$$t_s = \frac{T_2}{n} * randint(1, n-1) \tag{3.1}$$

For fixed allocation, user will recursively bisect the T interval to ensure their localization time is evenly distributed in the T_2 time interval. Formally,

$$t_s = \frac{T_2}{2^{\lfloor \log n \rfloor + 1}} * ((n - 2^{\lfloor \log n \rfloor}) * 2 + 1) \tag{3.2}$$

Regardless which allocation scheme we choose, from the backend server's point of view, localization for the group is performed at a needed higher frequency: every T_2/n time interval. For each user, on the other hand, they still benefit from the context facilitation as localization is performed every T_2 interval individually. This way we not only can detect changes in the location but also conserve energy.

3.1.4 Context Prediction

In addition to scheduled events such as those from public calendars, temporary or ad-hoc group or personal event can happen at a specific location for an extended period as well. These can serve as context information to facilitate our localization scheme.

The approach we employ to detect a temporary group gathering is to determine the similarity of behavior for a number of users at a fixed location. For example, if two or more users are in the same place already for a period, we can predict that the users are in a meeting and likely will remain at the place. Formally, if multiple users share the same AP_{str} and the similar range S_{str}, we can predict with high confidence that a temporary group meeting is in place and localization frequency can be reduced.

3.2 System Design

Our system is based on a client-server architecture, including the mobile client side and the backend server.

3.2.1 Mobile Client Side

Operation on the mobile client side has four components: 1) Collect WiFi APs' signal information; 2) Communicate the signal information to backend server; 3) Wait for the localization result from server; and 4) Setup different localization frequency/time scheme based on received result. The operation is detailed in Algorithm 1.

3.2.2 Server Side

The backend server will analyze the signal strength information sent from a mobile client, perform localization, determine the localization frequency and communicate it back to the client. The operation is detailed in Algorithm 2.

Here, "Further process" denotes how the server generates the localization scheme based on each user's new and previous status. This process is detailed in Algorithm 3.

Algorithm 4 determines how the server generates the reduced frequency based on different situations.

Result: Set next localization time

Collect and send AP list information to the server;

Wait for server response;

if *server response shows frequency reduction* **then**

  if *any event going on now* **then**

    if T_1 *minutes* $<$ *remaining time* **then**

      Set next localization time = T_1 minutes later;

    else

      Set next localization time = T_2 minutes later;

    end

  else

    Set next localization time = T_2 minutes later;

  end

else

  Do normal periodic localization every T_3 seconds;

end

Algorithm 1: Client Side Algorithm

3.3 Experiment and Evaluation

Our implementation was based on open source LAMP architecture. We also used open source social network platform Elgg [44] to handle interaction with each user and for future expansion. On the mobile side, we simply used Google gears to collect the available WiFi AP information and feeded it into our own server for further processing.

The client side was HP iPAQ 910 with pre-compiled Google gears. The server side was a Lenovo T500 laptop. We deployed 5 mobile devices on our campus envi-

Data: AP list information from the client side

Result: Localization scheme

Wait for AP list information from the client side;

if *this client's localization times* $\geq N$ **then**

  Calculate average signal strength, find AP_{str} and store the result;

  Compute location by RSSI triangulation;

  Remove existing N times AP list information;

  if AP_{str} *exists* **then**

    Generate energy-efficient localization scheme;

  else

    Generate normal localization scheme;

  end

else

  if *old status exists and old* AP_{str} *exists* **then**

    Further process (Algorithm 3);

    Discard received access points information;

  else

    Store received access points information;

    Generate normal localization scheme;

  end

end

Send localization scheme back to client;

Algorithm 2: Server Side Algorithm

ronment. The focused test field was the computer science building of 6 floors. The location of each access point was obtained from the campus IT team.

Data: Old status(including old AP_{str}, new AP list information

Result: Localization scheme

if *old status is in an event or ad-hoc group meeting* **then**

  if *old* $AP_{str} \equiv$ *new* AP_{str} *and old* $S_{str} \cong$ *new* S_{str} **then**

    if $ratio\ of\ users\ left \geq M\%$ **then**

      Generate normal localization scheme;

    else

      Generate energy-efficient localization scheme;

    end

  else

    Notify system this user leave this place;

    Generate normal localization scheme;

  end

else

  Generate normal localization scheme;

end

Algorithm 3: Further Process Algorithm

We performed real life experimental study to verify our scheme. We also performed the simulation study based on synthetic data to overcome the limitation on device numbers.

3.3.1 Synthetic Study

Firstly, we collected signal strength information in real life from different classrooms and constructed a model for the signal strength distribution. Evidently, distribution of signal strength is different for different situations [45]. In our study, the

Data: New AP list information

Result: Localization scheme

if AP_{str} *contains ap from range 0* **then**

  find room associated with this ap;

  if *event going on in this room now* **then**

    return event remaining time;

  else

    if $Num\ of\ users \geq K$ **then**

      return T_2 interval division and start time t_{start};

    end

    Generate normal localization scheme;

  end

else

  if $Num\ of\ users \geq K$ **then**

    return T_2 interval division and start time t_{start};

  else

    Generate normal localization scheme;

  end

end

Algorithm 4: Generate Energy Efficient Localization Scheme Algorithm

lognormal distribution fit our collected data well and hence was selected as the model for the signal strength from different APs. We computed the mean and standard deviation of real life collected signal strength and input them into the lognormal model to generate simulated signal strength from different WiFi access points.

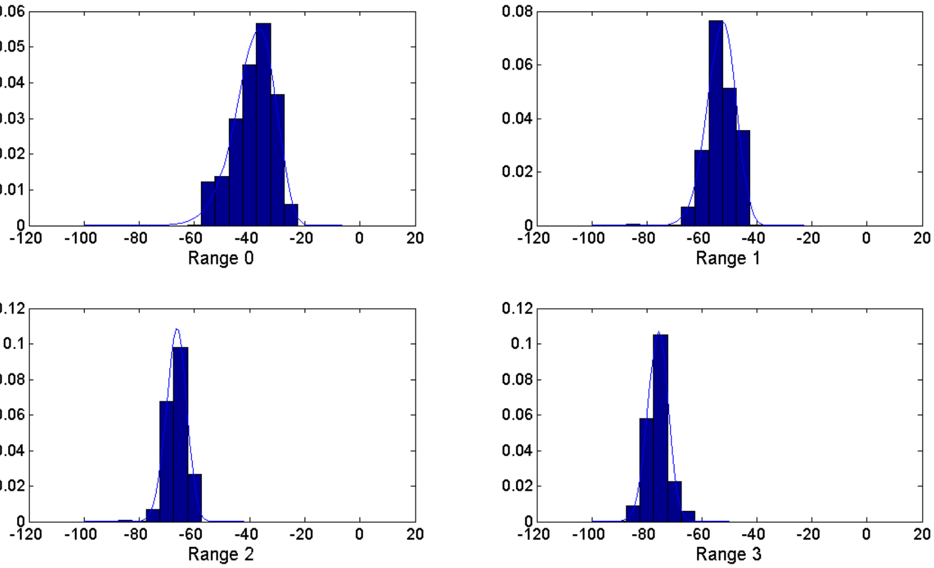


Figure 3.2. Signal Strength Distribution.

Figure 3.2 is 4 snapshots of the signal strength distributions in different ranges we used in the study.

We will focus on the number of times that localization is performed as the key comparison metric of different schemes. The results are summarized in Table 3.1 for a normalized 80 minutes period.

Table 3.1. Localization Frequency Comparison: Synthetic Data

	user 1	user 2	user 3	user 4	user 5
Normal scheme	160	160	160	160	160
COAL(Class)	14	15	16	15	16
COAL(Group Meeting)	20	19	17	17	18

3.3.2 Analysis

On average, in both class and group meeting cases, COAL outperformed the Normal Periodic Localization scheme. The total localization times was reduced by around 90%. In another word, 90% energy for localization operation could be saved. According to our measurement on the iPAQ, each localization request cost around 0.017% battery energy on the mobile phone. For an 80-minute event, periodic localization scheme cost 2.72% of the battery energy; and COAL, on the other hand only cost 0.238% battery energy. COAL could save around 2.482% battery energy during an 80-minute event.

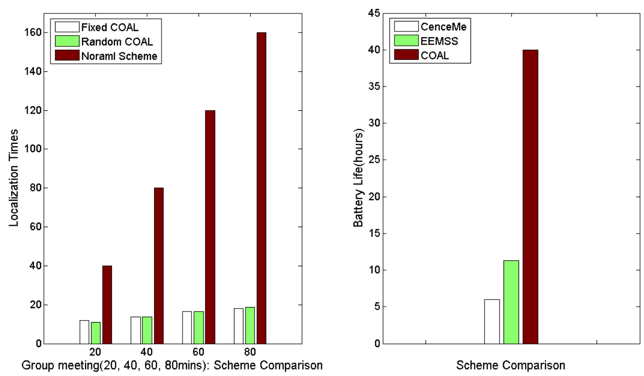


Figure 3.3. Localization Times and Battery Life Comparison.

We also studied the performance of the fixed allocation scheme and random allocation scheme regarding users' localization time (termed Fixed COAL and Random COAL respectively). As shown in Figure 3.3, both Fixed COAL and Random COAL outperformed Normal Periodic Localization scheme, especially when the group meeting is longer. Actually Fixed COAL and Random COAL performed similarly, as during, for example, a 10-minute interval, both of them on average only perform one localization request. The difference came at the end of the meeting. Random COAL might require more localization than Fixed COAL. The expected localization times in the whole system is shown in Fig. 3.4.

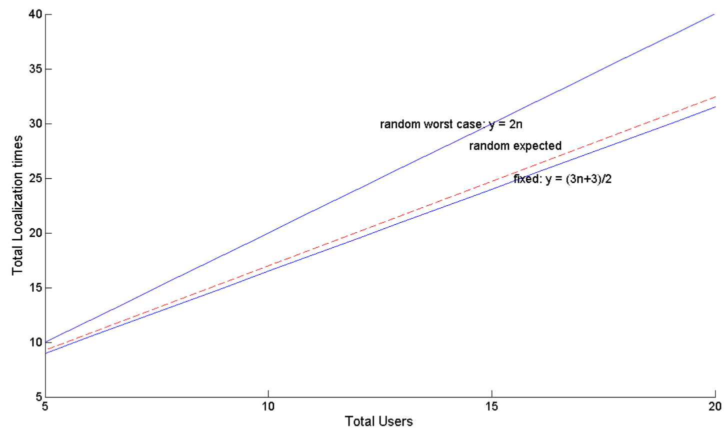


Figure 3.4. Comparison for Fixed and Random Allocation Schemes of Localization Starting Time.

We also compared COAL with schemes using different sensors including EEMSS [10] and CenceMe [22]. According to [12], given $T_{probe} = 30$ *seconds*, the WiFi in Nokia N95 can sustain 40 hours. In COAL, the localization period is at least 30 seconds, resulting in an average localization period $T_{COALprobe} \gg 30$ *seconds*. If we consider only WiFi functions for the mobile devices, then the total battery life for COAL in N95 is at least 40 hours, better than EEMSS and CenceMe.

3.4 Summary

In this chapter, we proposed COAL, a context aware localization scheme for mobile networks. By exploiting user context obtained either through online information or runtime determination such as classes or events, COAL can significantly reduce localization frequency and hence conserve energy for mobile devices. Additionally, we designed schemes that can quickly detect transient period when user locations change. We implemented COAL on iPAQ smart phones. Both real life experiments and simulations on synthetic data were performed. The results showed that COAL can effectively reduce localization frequency needed to determine a mobile device's location.

CHAPTER 4

CAPACITY OF PLACE BASED OPPORTUNISTIC NETWORKS

4.1 Network Model

While mobility is inherent for mobile devices and provides the underpinnings for data backhauling in opportunistic networks, human movement is intermittent and associated mobile devices will likely stop at a place for some time before continuing. As we can observe, mobile devices are relatively stationary when their owners are inside places, which makes the contact periods much longer and contact opportunities more predictable, as compared to those during movement. And this relatively stationary feature of mobile devices gives out the potential to make these mobile devices form ad hoc networks inside places. In this chapter, our target is the type of place centered opportunistic networks that possess the above features. In this kind of networks, a mobile device with information to be delivered will carry the information and travel among places. At that place, if the mobile device discovers other suitable mobile devices for relaying the message toward the destination, messages can be exchanged using short range communication interfaces such as Bluetooth or WiFi. A message relaying mobile device will be selected according to its likelihood of successfully delivering the information toward target places, determined by specific routing protocols. Messages are stored on mobile devices and transferred from one place to another when associated people are traveling between places. In this chapter, the total amount of data could be transferred in this network when sources transmit data only at an exact time point during some given period is defined as the capacity of this network, supposed that mobile devices in any places send out traffic to other places by uniformly distributed fashion.

As we know, human activity shows place-centered features: traveling between places and staying at places. Due to these features, to study the capacity of these networks, we separate the network into two layers as shown in Fig.4.1. The first layer is the human hopping movement among places, essentially determining the number of devices at a place and the human flow between places. This layer is built up from three queuing models detailed in later sections. The second layer is the message exchange among mobile devices at a place by constructing a wireless ad hoc network in that place.

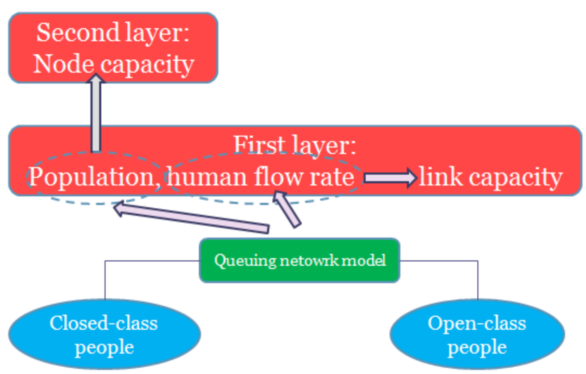


Figure 4.1. Two Layer Network Model.

4.1.1 First Layer Model

The first layer discusses and models the first feature mentioned above. Imagining public places like McDonald's in a city as nodes in the opportunistic network, every day people are traveling around these places. It is known that each person has his/her own schedule and will spend different amount of time in each place [46]. These daily activity and movement of people among places could be considered as the flow in the above network. Therefore, we propose a queuing model to capture the characteristics of this network formed by moving people, including population distribution in each place and flow between two places at different time during a day. This is the first layer in the network model.

Owing to different factors, including duration length of visiting time or types of people movement, we employ the following three queuing models that are first introduced in [25].

4.1.1.1 Open Form Queueing Network

Among all the people that visit the opportunistic network, a certain amount of them will spend a relatively short time in the network. In another word, they will visit one or two places and leave the network. For example, tourists could be considered as this kind of people. We consider these people as open-class jobs in the network and name them as open-class people. For these open-class people, we propose to model each place as an infinite server queue ($\infty/G/\infty$), and the whole network is an open form network of infinite-server queues. When a person, as a job, arrives at a place, he/she will be served immediately, and this process is independent of other people also in this place.

To formally present this queueing model, we introduce the following notations.

- U total number of open-class people in the network
- P total number of places
- U_i number of open-class people in place P_i
- $1/\mu_i$ the expected residence time of a person in place P_i
- λ_i arrival rate to place P_i
- ρ_i load of P_i, where $\rho_i = \lambda_i/\mu_i$
- p_{ij} empirical probability of an person move from P_i to P_j

- γ_i exogenous arrival rate to P_i
- Δt_{ji} the travel time from P_j to P_i
- T the whole observation period, from t_{begin} to t_{end}, T simulation clocks in total

In a common queueing network, we have the aggregate arrival rate as

$$\lambda_i(t) = \gamma_i(t) + \sum_{j \neq i} \lambda_j(t - \Delta t_{ji}) p_{ji}(t - \Delta t_{ji}),\ 1 \leq j \leq P \tag{4.1}$$

The probability that a person leaves the network is given by

$$p_{i0}(t) = 1 - \sum_{j=1}^{P} p_{ij}(t) \tag{4.2}$$

The marginal distribution of the total number of people in P_i is given by

$$P(U_i = u_i, t) = e^{-\rho_i(t)} \frac{\rho_i^{u_i}(t)}{u_i!} \tag{4.3}$$

4.1.1.2 Closed Form Queueing Network

Besides above open-class people, we also observe that some people will keep on traveling among different places for an extended period as closed-class jobs in the network. For example, residents of a city could be considered as this kind of people. For these closed-class people, we model each place as an infinite server queue ($\infty/G/\infty$), and the whole network is a closed form network of infinite server queues. When a person like a job arrives at a place, he/she will be served immediately, and this process is independent of other people also in this place.

To formally present this queueing model, we introduce the following notations.

- W total number of closed-class people in the network
- P total number of places
- W_i number of closed-class people in place P_i

- $1/\mu_i$ the expected residence time of a person in place P_i
- ρ_i load of P_i, where $\rho_i = \lambda_i/\mu_i$
- p_{ij} empirical probability of an person move from P_i to P_j
- v_i fraction of time of a person visit P_i

The aggregate arrival rate is given by

$$\lambda_i = \sum_{j \neq i} \lambda_j p_{ji}, \ 1 \leq j \leq P. \tag{4.4}$$

The marginal distribution of the total number of people in P_i is given by

$$P(W_i = w_i) = \binom{W}{w_i} v_i^{w_i}(1 - v_i)^{W - w_i}. \tag{4.5}$$

4.1.1.3 Mixed Queueing Network

By combining the above two kinds of people, we obtain the mixed form queuing network with two types of jobs, open-class and closed-class. Suppose there are L people in the network in total.

The aggregate arrival rate of the mixed form network is given by

$$\begin{aligned} \lambda_i(t) = {} & \gamma_i(t) \\ & + \sum_{j \neq i} \lambda_j^{open}(t - \Delta(t)_{ji}) p_{ji}^{open}(t - \Delta(t)_{ji}) \\ & + \sum_{j \neq i} \lambda_j^{closed} p_{ij}^{closed}, \\ & 1 \leq j \leq M \end{aligned} \tag{4.6}$$

The probability that an open-class person leaves the network is described as

$$p_{i0}^{open}(t) = 1 - \sum_{j=1}^{P} p_{ij}^{open}(t). \tag{4.7}$$

The marginal distribution of total number of people in P_i is described as

$$P_i(L_i = u_i + w_i, t) = e^{-\rho_i(t)} \frac{\rho_i^{u_i}(t)}{u_i!} \binom{w}{w_i} v_i^{w_i} (1 - v_i)^{w - w_i}. \tag{4.8}$$

As we can observe from above equations, the following parameters are crucial to calculate P_i. And we can obtain them by analyzing the real trace data, and we will discuss how to compute these parameters in the experiments section.

Table 4.1. Parameters to be collected

$1/\mu_i(t)$	the expected residence time of a person in place P_i
$p_{ij}^{open}(t)$	empirical probability of an open-class person from P_i to P_j
p_{ij}^{closed}	empirical probability of a closed-class person from P_i to P_j
$\gamma_i(t)$	exogenous arrival rate to P_i
v_i	fraction of time of a person visit P_i

4.1.2 Second Layer Model

Here we will study the second feature mentioned above and establish the second layer model that focuses on message exchange at a place. The second layer captures the human hangout activity and the resulting data exchange among mobile devices in a place, which is similar to data exchange in a wireless ad hoc network. Therefore, we could model each place and mobile devices inside it as a wireless ad hoc network and study the capacity of this network. Besides that, human activity varies in different places at different time. For example, in movie theaters or restaurants, people are stationary at most of the time. On the contrary in a shopping center, most of the people are in the movement at most of the time. Therefore, according to the types of human activity (stationary and mobile) in each place, they could be modeled as different types of wireless ad hoc networks. Fortunately, there are well-known results [8, 9] in the literature that fit into different scenarios in terms of network capacity in our place centered opportunistic network.

Assume that each device is capable of transferring at S bits/sec. According to [8], the throughput of a wireless ad hoc network with n static nodes is $\Theta(\frac{S\sqrt{n}}{\sqrt{\log n}})$. Also, when considering the mobility in the wireless network, the throughput could be improved to $\Theta(Sn)$ [9]. Besides that, [47] discussed the capacity of hybrid wireless networks, which depends on the growing speed of number of the base stations. If the number of base station grows fast enough, the capacity of whole hybrid networks is $\Theta(Sm)$, where m is the number of base stations. Since our place centered opportunistic networks are pure ad hoc networks formed only by mobile devices, we will not consider the hybrid networks containing base stations. Results from [8, 9] can be leveraged to calculate the node capacity at each place, i.e., the total amount of data that can be exchanged per unit time at a place. We will formally introduce the capacity calculation details in the later section.

We remark that the number of people in a place normally will vary from time to time. For example, in a restaurant, there are usually more customers during lunch and dinner time than the morning and afternoon. Therefore, the capacity of each place will vary with time. Previous results given by [8, 9] are under the condition that the network is stable (number of nodes is fixed). A time-varying extension of these results should be developed for our scenario. This problem is detailed in the following section.

4.1.3 Combination of First Layer and Second Layer

As stated above, the first layer is based on human movement pattern in place centered network, and the second layer is based on the wireless ad hoc network in each place. We are interested in finding out the capacity of the resulting network. The challenge here is to effectively combine the first and the second layers together, and transform them into basic node and link capacity of a network. Firstly, we begin with formally constructing the mathematical formula of node capacity and link capacity.

4.1.3.1 Link Capacity

Assuming that each person has one mobile device, and each mobile device provides fixed amount of memory to facilitate the data exchange in the network, we then can transform the first layer human flow into links in our network. Basically, a link is built upon the human movement from one place to another and the total memory provided by their mobile devices. Suppose that each mobile device provides B bits memory from its sd card, we have the link capacity from place m to place n at time t as

$$\begin{aligned} C_{(m,n)}(t) \quad =& \lambda_m^{open}(t) * p_{mn}^{open}(t) * B + \\ & \lambda_m^{close}(t) * p_{mn}^{close}(t) * B, \quad (m,n) \in E \end{aligned}$$

Here E represents all links in the network. Briefly, the link capacity between two places indicates the amount of data could be transported from one place to another by all mobile devices traveling between these places during given period. Since the links in this network are built based on human flow, they also vary with time.

4.1.3.2 Node Capacity

We can also derive the node capacity. Node is basically each place, and its capacity is related to the total number of mobile devices or population in itself. Firstly, as we have discussed in the first layer section, a distribution about total number of people in each place is given; then at time t, we can use the mean of this distribution μ to approximate the actual number of people in that place, since human population hardly changes dramatically given a short time period. Secondly, we image that all the mobile devices carried by people form a wireless ad hoc network. According to [48], several schedule algorithms could be utilized to guarantee linear convergence speed to the stable state of the wireless network. Since in our scenario people usually spend a much longer time than this convergence time in each place, we consider that most of the time the wireless network in each place is stable and we use the number of people at that time as input value ($P_n = P_n(t)$). Then at the time t, we can use the second layer model to derive the node (each place) capacity at place n at time t. What's more, as mentioned before human activity can vary in different places, which will result in different wireless ad hoc network environment and in turn affect the induced node capacity. Suppose the node capacity is C_n, and we will discuss them in the later independent sections with their own experiments results.

4.2 Network Capacity of Places with Static People and Experimental Results Analysis

4.2.1 Node Capacity

The first one is the type of places where people barely move around or move in very limited boundary during most of the staying time like offices and movie theaters. In this kind of places, we consider each mobile device that associated with each person is static in its own territory compared to other mobile devices. In reality, there may be a few people moving around sometime; however, the moving time is instantaneous without patterns to leverage, and is too little compared to the static time. Therefore, we treat all mobile devices to be static in this type of places. All these mobile devices inside a place together could be imaged as a static wireless ad hoc network. Assuming that each device is capable of transferring at S bits/sec. By keeping the same assumption in [8], we derive the node capacity of the first type of place n at time t as (Here N indicates the set of places in the network, and we use $\mu(P_n)$ to approximate $P_n(t)$):

$$C_n = \Theta(\frac{S * \sqrt{\mu(P_n)}}{\sqrt{\log \mu(P_n)}}), \quad n \in N$$

4.2.2 Network Capacity

In this section, we detail the derivation of the capacity of the network. We consider one source and one destination case at first.

First of all, we define the following notations to better explain the idea. Let $G = (N, E)$ be the whole network.

- G whole network composed of places and links
- N the set of places in the network
- E the set of links between different places
- $C_{(m,n)}$ for link $(m, n) \in E$, the capacity of this link, defined by the first layer human flow and the memory provided by each mobile devices
- C_n the capacity of place n, defined by the capacity of wireless ad-hoc network formed by people in that place
- Q_T the capacity of the whole network during the period T

- $f_{(m,n)}$ flow from place m to place n
- f_s flow from source s
- t_{begin} the beginning time point of observation
- t_{end} the end time point of observation
- T the whole observation period T simulation clocks in total
- s source node
- d destination node

As noted earlier, the capacity of a place centered network is normally time-varying, mainly due to the change of the number of people presenting at and moving among different places at different time. In other words, C_{ij} and C_i are time-varying. Time decomposition approaches are proposed in [49, 50] to transform the network into a multistage one to solve this problem. Two techniques are usually considered: pure time-expanded network (inter-link capacity independent) and scenario-based network (inter-link capacity dependent). Due to the complexity introduced by the second method, in this chapter we will employ pure time-expanded network one firstly, and plan to use the second one in future when more accurate models are needed to characterize the more detailed network scenario. Fig.4.2 is an example to transform a time-varying network into time-expanded network. Briefly, the left part is the original network with a tuple ($distance, time$) assigned to each link; and the right part is the consequent network when we expand the original network by time.

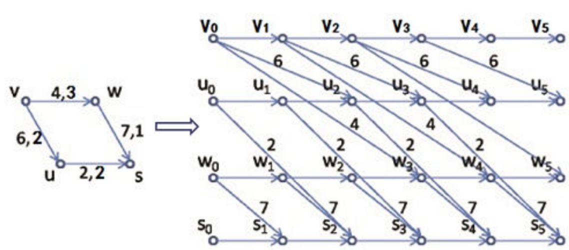


Figure 4.2. Time Expanded Network.

4.2.2.1 Network Capacity Definition

We formally define network capacity here: 1) The capacity of the network during some period $Q_{t_{end}-t_{begin}}$ (source could start to transmit data at any time during this period). 2) The capacity of network start exactly at some time point and during given period $Q_{[t,t_{end}-t]}$. The first one is the cumulative summation during some period $t_{end} - t_{begin}$ of the second one.

$$Q_{t_{end}-t_{begin}} = \sum_{t_{begin} \leq t \leq t_{end}} Q_{[t,t_{end}-t]} \tag{4.9}$$

We will focus on how to calculate $Q_{[t,t_{end}-t]}$ in the followings and use T to represent the time duration $t_{end} - t$. And also we use this as the network capacity during the experimental results analysis.

4.2.2.2 Network Capacity without Node Capacity

Firstly, we consider the situation without node capacity constraint at a given time point t. This is a maximum flow problem, and we can solve it by applying a simplex algorithm [51] or using a classic maximum flow algorithm [24]. The following equations illustrate the maximum flow problem constructed based on our network scenario in linear programming form:

$$\begin{aligned}
&\underset{Q_{[t,T]}}{\text{maximize}} && Q_{[t,T]} \\
&\text{subject to} && Q_{[t,T]} \geq 0 && (4.10) \\
& && Q_{[t,T]} = f_s && (4.11) \\
& && -f_s + \sum_{m:(m,d)\in E} f_{(m,d)} - \sum_{m:(d,m)\in E} f_{(d,m)} \leq 0 && (4.12)
\end{aligned}$$

$$m \in N, (m,d) \in E$$

$$0 \leq f_{(m,n)} \leq C_{(m,n)} \tag{4.13}$$

$$\sum_{m:(m,n)\in E} f_{(m,n)} - \sum_{m:(n,m)\in E} f_{(n,m)} \leq 0 \tag{4.14}$$

$$m, n \in N - \{s, d\}, (m,n) \in E$$

$$f_s + \sum_{m:(m,s)\in E} f_{(m,s)} - \sum_{m:(s,m)\in E} f_{(s,m)} \leq 0 \tag{4.15}$$

$$m \in N, (m,s) \in E$$

4.2.2.3 Network Capacity with Node Capacity

In the case that a node has node capacity constraint, let C_n denote the node n's capacity, as defined in the previous section. We can expand the above network, and split a node into two nodes and create a link with capacity C_n between these two nodes, as depicted in Fig.4.3. These two nodes are input node and output node, and the new link is from the input node to the output node. The input node is connected to the incoming links that connect to the original node, and the output node is connected to the outgoing links that connect to the original node. Then this expanded network becomes new network with extra nodes and links instead of the original one.

Under the new node set is N^* and the new edge set E^*, the maximum flow problem in linear programming form becomes

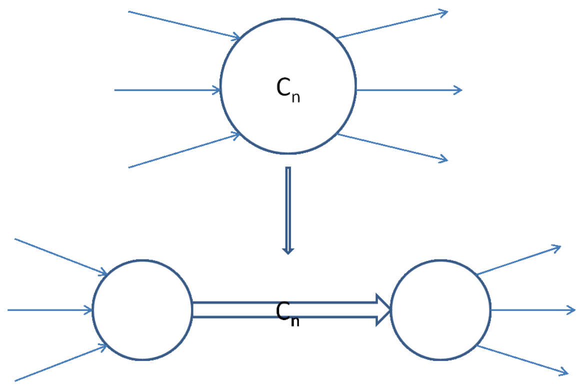


Figure 4.3. Node Capacity Transformation.

$$
\begin{aligned}
\underset{Q_{[t,T]}}{\text{maximize}} \quad & Q_{[t,T]} \\
\text{subject to} \quad & Q_{[t,T]} \geq 0 && (4.16) \\
& Q_{[t,T]} = f_s && (4.17)
\end{aligned}
$$

$$- f_s + \sum_{m:(m,d)\in E^*} f_{(m,d)} - \sum_{m:(d,m)\in E^*} f_{(d,m)} \leq 0 \qquad (4.18)$$

$$m \in N^*, (m,d) \in E^*$$

$$0 \leq f_{(m,n)} \leq C_{(m,n)}, \qquad (4.19)$$

$$\sum_{m:(m,n)\in E^*} f_{(m,n)} - \sum_{m:(n,m)\in E^*} f_{(n,m)} \leq 0 \qquad (4.20)$$

$$m, n \in N^* - \{s, d\}, (m,n) \in E^*$$

$$f_s + \sum_{m:(m,s)\in E^*} f_{(m,s)} - \sum_{m:(s,m)\in E^*} f_{(s,m)} \leq 0 \qquad (4.21)$$

$$m \in N^*, (m,s) \in E^*$$

For multiple sources and multiple destination cases, we can transform them into one source and one destination case by introducing a super source connected to all sources and a super destination connected to all destinations. Then given any observation time frame from a specific time point, network capacity can be calculated by solving the above linear programming problem.

4.2.3 Experiments

4.2.3.1 Experiment Design

We designed an experiment scenario to test our proposed model. Basically in a city containing P places, L pedestrians were traveling among these places, including U open-class pedestrians and W closed-class pedestrians. We only used pedestrian here since the different traveling speeds were not major element that affect network capacity. Each person would have randomly prepared schedule, and they visited different places by following the schedule. After running the simulation, as we discussed before, parameters in Table.4.1 should be obtained based on the simulation results. The followings are some important notations and detailed equations to calculate those parameters:

- U_{ij} number of open-class pedestrians travel from P_i to P_j
- U_{i0} number of open-class pedestrians leave the network
- W_{ij} number of closed-class pedestrians travel from P_i to P_j

We calculate $p_{ij}^{open}(t)$ by:

$$p_{ij}^{open}(t) = U_{ij}^{open}(t)/(sum_j U_{ij}^{open}(t) + U_{i0}(t))$$

We obtain the $p_{ij}^{closed}(t)$ by:

$$p_{ij}^{closed}(t) = W_{ij}^{closed}(t)/sum_j W_{ij}^{closed}(t)$$

We acquire v_i by:

$$v_i = sum_w t_w(i)/sum_p sum_w t_w(p)$$

And for $1/\mu_i(t)$ and $\gamma_i(t)$, since they can be easily calculated by averaging corresponding simulation results, in this chapter we plan to predefine these two parameters in the experiment setup.

An experimental scenario that contained 10 places and predefined paths between them in a 1000*1000 map was setup. Pedestrians moved along the paths between two places by their own schedules at speed $15/simclock$. Place visiting schedules were uniformly distributed. The number of closed-class pedestrians was set to be 1000. Open-class pedestrians visited the network at the rate $10/simclock$, which was 1 for each place. When data were carried to their destinations, they were considered to be successfully delivered and the memory they occupied in the host mobile device would be released. The transmission rate of mobile devices was set to be $1MB/simclock$, and message size was 1KB. We also tested other values and similar results were obtained. Table.4.2 lists all the default parameter values chosen for this experiment (time unit is the simulation clock step).

And we were interested in finding out the possible effect on network capacity caused by following parameters in Table.4.3:

Table 4.2. Experiment Parameters Setup

Total places	10
Total closed pedestrians	1000
Open pedestrian arrival rate γ_i	1
Open pedestrian stop arriving time	N/A
Place stay time	$uniform(10, 20)$
Pedestrian speed	15
Buffer size	$1MB$
Message size	$1KB$
Mobile device transmission rate	$1MB$
Map size	$1000 * 1000$
Place coordinates	(x, y) $x, y \in uniform(0, 1000)$
Total simulation time	400

Table 4.3. Selected Parameters To Be Analyzed

Buffer size	How much buffer each mobile device provides
Open arrival rate γ_i	How frequently people arrive in the network
Open arrival stop time	When open-class people stop arriving
Place staying time	How long people spend in a place on average

4.2.3.2 Result Analysis

Based on the results of the experiments, given any time frame, we can compute the capacity for this network. Fig.4.4 showed stabilized network capacity in a long term about networks formed by purely first type of places. Fig.4.5(a),4.5(b),4.5(c),4.5(d) were depictions of the time-varying capacity $Q_{[t,100]}$ comparison between several specific parameter values from different simulation time point within 100 simulation clocks. In those figures, the left part is the network capacity from the first type of places, and the right part is from the second one.

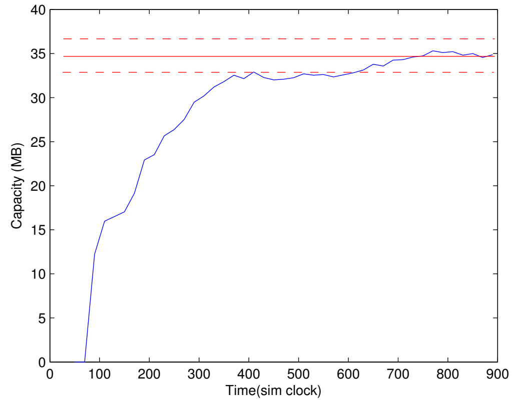
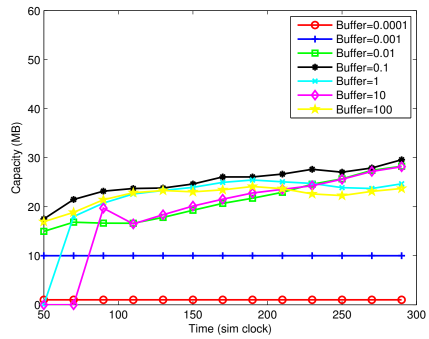


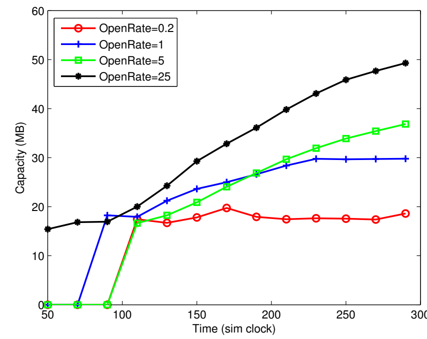
Figure 4.4. Network capacity between 50-900 sim clock.

From above data, some valuable conclusions could be observed. In Fig.4.4, we could see the network capacity was increasing at the beginning of the whole simulation and eventually stabilized around a point. As we know, the number of closed-class people was fixed and the number of open-class people was changing with time; in the beginning, the total number of people increased due to arrival of open-class people, which caused the network capacity to increase as node and link capacity depend on the number of people. However, after some simulation time, the open-class people started to leave the network, which caused the total number of people to become stable, and this induced the network capacity to stabilize around a point. And the fluctuation we observed in the results was due to the dynamic network topology. In order to show the stabilized network capacity, we ran the simulation for 1000 simulation clocks time. In later figures, we would only focus on showing the effects introduced by different parameter values and not on the stable part of the network capacity. Therefore, we would stick to the default 400 simulation clocks running time.

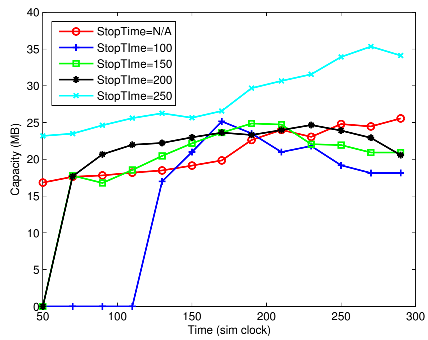
From Fig.4.5(a), network capacity increased with larger buffer size provided by each mobile device due to the fact that it is one of the factors that affects link capacity. However, link capacity is not the only parameter which determines the net-



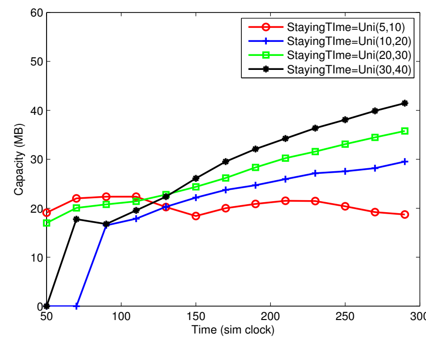
(a) Network capacity with different buffer size provided by pedestrian



(b) Network capacity with different open-class pedestrian arrival rate



(c) Network capacity with different time point when open-class pedestrian stop arriving



(d) Network capacity with different place staying time

Figure 4.5. Network capacity with different parameters setup, between 50-290 sim clock.

work capacity. When the buffer size reaches a certain number, the network capacity is limited by the "node capacity" in each place, which depends on the amount of pedestrians in each place and will be illustrated and explained later.

From Fig.4.5(b) and Fig.4.5(c), the network capacity was changing along with the human flow. When the number of closed-class pedestrians was stable and the open-class pedestrian arrival rate increased, the capacity of whole network increased since the increasing population in the network caused the growing node capacity and

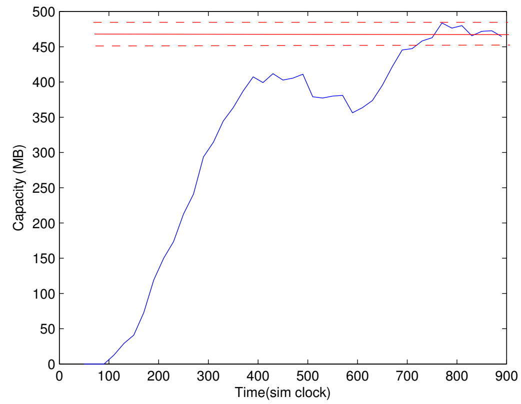


Figure 4.6. Network capacity between 50-900 sim clock.

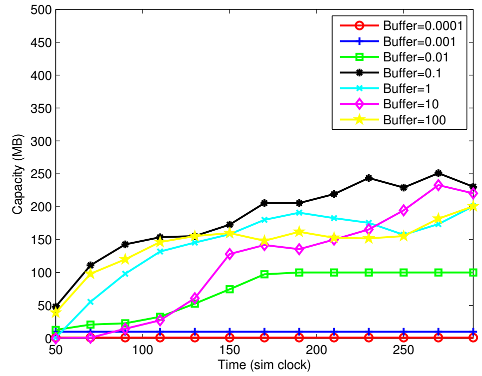
link capacity; when open-class pedestrians stopped entering the network, the network capacity would decrease after some time due to the decreasing population.

From Fig.4.5(d), we observed that if pedestrians spent more time at each place, the network capacity would increase. The reason is that given the same arrival rate, longer staying time implies more people in that place statistically. According to our node capacity definition, node capacity increases with more pedestrians, which in turns causes growing network capacity.

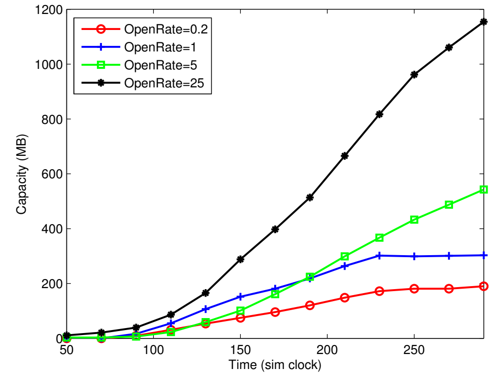
4.3 Network Capacity of Places with Mobile People and Experimental Results Analysis

4.3.1 Node Capacity and Network Capacity

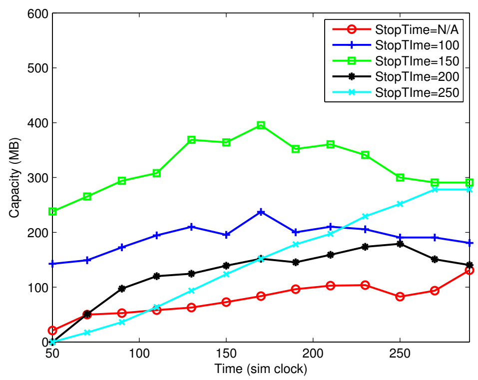
Besides the first type of places, it is easy to observe that in some places like shopping centers and theme parks, people are moving around inside these places at most of the time, and their movement is independent of each other. This is different from the situation in the first type of places. As we know, the mobility always introduces more uncertainty and more opportunity. Therefore, we are interested in



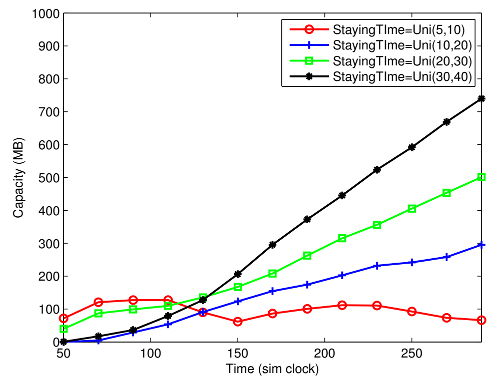
(a) Network capacity with different buffer size provided by pedestrian



(b) Network capacity with different open-class pedestrian arrival rate



(c) Network capacity with different time point when open-class pedestrian stop arriving



(d) Network capacity with different place staying time

Figure 4.7. Network capacity with different parameters setup, between 50-290 sim clock.

studying the characteristics of this kind of places and group them into the second type namely "mobile" places aside from the first type namely "static" places. In this kind of places, mobile devices that are associated with their owners are also moving around with no boundary. Briefly, we consider all these mobile devices inside a place together as a mobile wireless ad hoc network. According to previous work, the mobility introduced by human in the second type of places could increase the number of contacts between mobile devices, which infers to increase the data exchange chance. All these potential factors could affect the node capacity and resulting network capacity. Assuming that each device is capable of transferring at S bits/sec. People may have different movement pattern in different places, such as predefined-path movement in museum and random-path movement in the shopping mall. However, according to [52], two-dimensional mobility pattern is not a necessary condition for the result to hold. Therefore, the movement of people in each place could satisfy the requirements of assumptions in [9]. By taking the same loose delay constrains assumption in [9], we derive the node capacity of the second type of place n at time t as (Here N indicates the set of places in the network, and we use $\mu(P_n)$ to approximate $P_n(t)$):

$$C_n = \Theta(S * \mu(P_n)), \quad n \in N$$

4.3.2 Comparison between "mobile" places and "static" places

In section III, results from [8] are leveraged to obtain node capacity for "static" places. As we know, the conclusions in [8, 9] are based on the prerequisites of unit area $1m^2$ and self-chosen transmission range/power by each node. To both better apply their conclusions and meet reality requirements, in our network scenario the unit area is considered to be maximum transmission range that could be reached by current P2P wireless technology. For the places are so large that their areas exceed the maximum transmission range under current wireless technology, we could just scale the capacity results by relative constant. However, since this scale is independent of the population in each place, for simplicity we will not consider it individually for each place and treat each place as unit area without discrimination. Due to the same setup, in "static" places too much bandwidth consumed by relaying and bad channel condition caused by signal interference result the insufficient data transferring, which in turns lower network throughput.

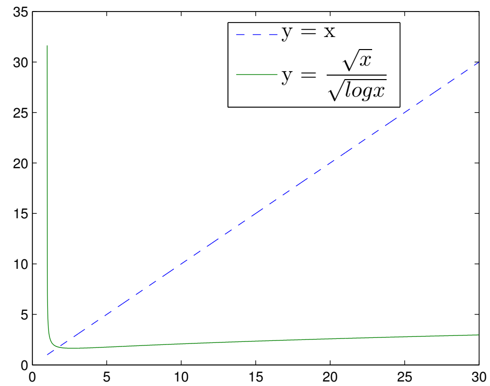


Figure 4.8. Comparison between mobile and static capacity function.

While in "mobile" places, as discussed in [9], mobility and resulting multiuser relaying possibility could improve the network throughput if we keep the same assumption (unit area and maximum transmission range). Two keys things are required to distinguish and improve network throughput from static wireless networks: 1) mobility of wireless nodes 2) relaying functionality provided by wireless nodes. Both of them could be satisfied in our "mobile" places. Firstly, people and associated mobile devices are naturally moving in these places. Secondly, their mobile devices could be easily served as relaying nodes with currently equipped advanced wireless functionality. When these two key components are enabled, the multiuser means could be utilized to make concurrent successful transmission, which is impossible in "static" places. What's more, to follow specific paths are not a requirement to hold the capacity results [52]. And the time scale to achieve network capacity improvement is also tolerable due to the fairly enough human walking speed and relatively short and reachable transmission range in each place.

From mathematical perspective, we could plot the node capacity curve of these two types of places in Fig.4.8:

The intersection of two capacity functions falls into interval $[1, 2]$. However, it is an extreme situation, which is not applicable in reality. The reason is that the x axis indicates the number of people in a place, and a wireless network needs at least 2 wireless devices to form. Except above special situation, in most of the cases (interval $[2, +\infty]$), the network capacity generated from "mobile" places is larger than that from "static" places. Even in the optimal situation (with perfect node placement and scheduling) in "static" wireless network (capacity is $\Theta(S * \sqrt{\mu(P_n)})$), the capacity increase still cannot catch up with the gain obtained in "mobile" places as population increases. These are cooperation outcomes owing to the mobility and relaying mechanism introduced in "mobile" places. We can apply the network capacity results of mobile wireless networks like this to better approach real scenario: when the population is very small, we can easily reach the network throughput without worrying about the negative effect caused by interference; when the population is large, relatively fairly spread population density by everyday experience and human mobility could step in and ensure the network capacity.

For places that possess neither purely mobile nor static feature such as airports, we will not include them in current network model. As we know, only relaying could not improve the network capacity [8], so does the mobility only [9]. Both of them need to be enabled on every mobile device in the network to ensure $\Theta(n)$ concurrent successful transmissions in order to obtain capacity results in [9]. For above places, since not all the nodes are mobile, we plan to find suitable solutions for them in future. And we follow the same steps in the last "static place" section with above new node capacity formula to compute network capacity.

4.3.3 Experiments and Result Analysis

The experiments setups were prepared similarly as in the last ”static place” section. Besides that, we assigned each person with a moving speed σ inside each place, the default value 3 per sim clock.

Based on the results of the experiments, given any time frame, we can compute the capacity for this network. Fig.4.6 showed stabilized network capacity in a long term about networks formed by purely the second type of places.

Fig.4.7(a),4.7(b),4.7(c),4.7(d) were depictions of the time-varying capacity $Q_{[t,100]}$ comparison between several specific parameter values from different simulation time point within 100 simulation clocks.

Similar conclusions as described in the ”static place” section could also be obtained from 4.6,4.7(a),4.7(b),4.7(c),4.7(d), except the saturated point network capacity for buffer size. Besides that, we could observe the capacity difference between network formed by purely the first type of places and purely the second type of places: the network capacity increased dramatically in the network formed by the second type of places due to the mobility of mobile devices. For any networks formed by the combination of these two types of places, the network capacity will reveal the similar pattern. What's more, in Fig.4.7(c), we observed that the network capacity showed the big difference under same parameters setup except stop time of the arrival of open-class pedestrians. As we further analyzed, this was mainly due to the different network topology resulting from human movement in the whole city.

4.4 Summary

In this chapter, we studied the capacity characteristic of PopNet, a type of opportunistic networks centered on places. Briefly, we proposed a two-layer model to calculate the capacity of this network. Places in the real world were considered as the nodes in this network, and human daily activity formed the links between different nodes. In the first layer, we deployed three queuing models to compute the population in different places in this network as well as the human flow rate among places, and formulated the link capacity from these human flow. In the second layer, by mapping the network formed by mobile devices in each place to a wireless ad hoc network, we formulated the node capacity. In the end, we derived the capacity of the network by solving a time-varying maximum flow problems. As our experiments showed, buffer size provided by each device, open-class pedestrians arrival rate to each place and also the staying time in each place could affect the capacity of the network.

CHAPTER 5

ROUTING DESIGN AND ANALYSIS OF PLACE BASED OPPORTUNISTIC NETWORKS

5.1 Proposed Routing Schemes

We studied the capacity of place based opportunistic networks in the last chapter, and will continue to discuss the possible routing schemes for this new network scenario. Briefly, in this chapter we propose two new routing schemes according to the features of this new place based opportunistic network. New routing schemes designed specifically for this new scenario will be introduced in this chapter. Because of the repeating pattern of human daily activity, history information about each place such as population and travel schedule could be easily gathered from online location-based service like Foursquare. Gossip protocol [53] and WiFi/Bluetooth P2P technique will serve as the basic blocks to share important data such as average buffer occupation rate among mobile devices. Detailed discussion about how to achieve this will be skipped, since we are more interested in how various information could effect the performance of routing schemes.

5.1.1 Routing Scheme Design

In conventional opportunistic networks and delay tolerant networks, contact possibility between two mobile devices in anywhere is one of the important concerns during routing scheme design. Basically, contact possibility is an indicator about how possibly a mobile device candidate could forward data to the destination mobile device. However, in this chapter we consider routing design in a more macro level with less focus on contact possibility. For example, if data need to be routed from one place to another, this process does not require any particular mobile device to do the transfer job. In another word, any mobile devices that will visit that place could assist this process. Therefore, the routing decision is made based on the information among all mobile devices in each place, such as the visiting schedule in each mobile device.

5.1.1.1 Geographic-Population-Based(Rank) Scheme

It is known that history of a place is valuable information, which could be leveraged in routing scheme design. In this section, population history of a place is the basic information we choose to take advantage of. Population information could reflect how popular a place is, since people have the tendency to visit popular places. And as we know, if a person can meet more people in a popular place, it implies more interactions between mobile devices carried by people in that place. In another word, the more mobile devices in a place, the larger relay candidates pool and the higher chance data could be exchanged. In a result, places with more people will be assigned a higher priority, and data will be forwarded to that place with higher probability. Besides that, places with more outgoing connections to other places could provide more path options. And this out-degree property is based on the diversity of schedules of people in that place, since links between places are built upon human flow between places. The more diverse the schedules in a place are, the larger out-degree it has. And the diversity of schedules in a place is usually in proportional to population in that place. This out-degree information could be considered as another popularity information. Briefly, we treat this popularity information as extra metrics added into the shortest path routing schemes. The following Alg.5 illustrates this routing schemes in details:

Data: Data await transfer, population history, out-degree history

Result: Next place where data will travel to

$NextHop = P_{random}$

Calculate top k shortest paths list $PathList$

for $Path_i$ *in* $PathList$ **do**

  calculate the next place popularity indicator Pop_i

end

Find the maximum Pop_{max}

Set $NextHop =$ First place of $Path_{max}$

Algorithm 5: Data next hop selection in population based routing scheme

We present how to calculate the popularity indicator in the rest part of this section. Popularity indicator is composed of two elements: out-degree rank and population rank. For out-degree rank, we use PageRank[54] similar technique to assign a value to each place. Firstly, we introduce a vector $B = (b_1, b_2, ..., b_m)$, which indicates the out-degree rank of each place, and a link matrix ($a_{i,j}$ indicates whether a path exists from i to j) into popularity indicator calculation.

$$A_{m,m} = \frac{1}{\sum a_{ij}} \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,m} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,m} \end{pmatrix}$$

B is set to the initial value $(1/m, 1/m, ..., 1/m)$, and we keep on recomputing the following formula until the convergence vector B_{cvg} is obtained (basically the difference between B_i and B_{i-1} is less than a threshold value δ).

$B_i = B_{i-1} * A_{m,m}$

Secondly, we calculate the normalized vector *Popu* of population in each place. P_i indicates the number of people in place i

$Popu = (P_1, P_2, P_3, \cdots, P_m)/\sum_i P_i$

The final popularity indicator Pop_i will be computed by multiplying the population indicator and centrality (out-degree rank) indicator

$Pop_i = B_{cvg}(i) * Popu_i$

In a result, the place with highest popularity indicator in the next hops of top k shortest paths to the destination will be chosen as the actual next hop.

5.1.1.2 Game-Theory Competitive(Game) Routing Schemes

Besides population history, congestion history could also be utilized to facilitate the routing scheme design. The reason to take congestion history into account is that network congestion could cause data drop and consequently result in longer delivery time, which jeopardizes the effect of shortest path routing scheme. In another word, the shortest path scheme only considers distance factor, and this could cause a lot of data drops if we keep on sending data to the nodes in paths that are central ones with limited resources and already congested. In reality, network resources are limited. If data are routed to a busy router, they will be dropped with highly probability. Then the whole routing process may take more time. Based on this observation, we decide to introduce congestion information into routing scheme, and try to find shortest paths with the least congested next hop. However, instead of solely attempting to control the incoming data rate from start place in order to reduce congestion at end place, we propose a new routing scheme that takes available congestion history information into account and also looks after the fairness among several competitive flows to the same end place. Fairness could make sure data from different start places get fair opportunities to be relayed. Briefly, our idea is to introduce a competitive game in routing scheme to guarantee that each link gets fair network resources and decreases

Table 5.1. Competitive Game Reward Table, service rate is M

	A Obey($1/2M$)	A Deny(M)
B Obey($1/2M$)	$1/2M$, $1/2M$	$1/3M$, $2/3M$
B Deny(M)	$2/3M$, $1/3M$	$1/2M$, $1/2M$

data drop rate at the same time. Details about this routing scheme are described in the rest part of this section.

Firstly, the following game Table 5.1 will be utilized as an example to illustrate how this routing scheme keeps fairness and reduces congestion. Supposed that place D can serve M data per unit time, and the utility gained by serving M data is M. Place A and B send data to D at the same time.

As we can observe from Table 5.1, if one place sends more data than it does when both of them are unselfish (top-left), the reward for selfish one is higher, so he has incentive to send more data. But this selfish action harms the other one's reward; and the other one could penalize the selfish one by increasing its own data sending rate, and make its competitor run in a low reward mode. In this situation, congestion is increasing and the whole network will become unstable and finally the reward for both will be actually lower than $1/2M$. From the perspective of the whole network, if everyone follows the original data rate, the whole network gains the best reward, and the network congestion is low. The table is one time game; however, according to Nash Equilibrium in infinite game [55], we know that in the infinite version of this game, both players will become unselfish and this situation is the equilibrium state for the whole system. Therefore, each node should obey the unselfish sending rate. This two-player game could be easily extended to multiple players cases, and the state that all of them keep unselfish is the Nash Equilibrium for the new game.

The following competitive game routing scheme Alg.6 is proposed, and routing decision will be made based on the next hop history information. The history about the number of pedestrians in that place will be utilized to calculate node capacity. Suppose there are N people in place A, and then we can calculate the theoretical capacity C of the place by the way introduced in the previous section. Besides that, the amount of data delivered from current place to that place will be used to check and keep the equilibrium of the competitive game in order to maintain fairness among flows. In the end, data will be delivered to the place that possesses the most available resource.

5.2 Experiments

5.2.1 Experiments design

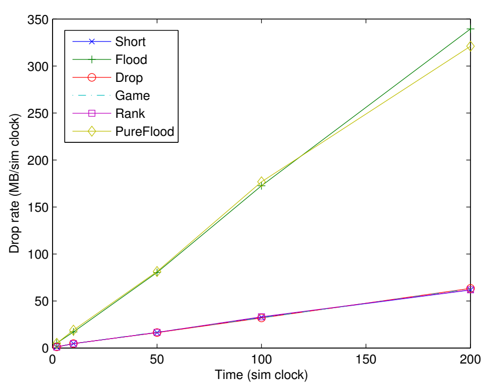
We designed an experiment scenario to test the proposed routing schemes and compared experiment results obtained after deploying different routing schemes mentioned previously in the network. Briefly, in this scenario, there were M places, X open-class (temporally visiting the network) pedestrians and Y closed-class (always staying in the network) pedestrians were traveling among these places (Z pedestrians in total). Each person would have randomly prepared schedule, they visited different places by following their own schedules. After running the simulation, we are interested in analyzing the following results in Table.5.2:

An experiment scenario that contained 10 places in a 1000*1000 map is setup. Pedestrians moved along the paths between two places by their own schedules at speed $15/\textit{sim clock}$. The number of closed-class pedestrians was set to be 500. Open-class pedestrians visited the network at the rate $2/\textit{sim clock}$, which was 1/5 for each place. When data were carried to their destination, it was considered to be successfully delivered and the memory they occupied in the host mobile device would be released. The transmission rate of mobile devices was set to be $1MB/sim\ clock$, and message size was 100KB. We also tested other values and similar results were obtained. Table.5.3 lists all the default parameter values chosen for this experiment (time unit is the simulation clock step).

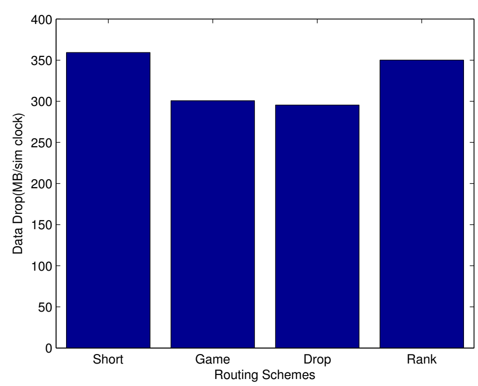
5.2.2 Result analysis

Given any time frame, we can obtain results for each routing schemes, such as the delivery rate in different simulation time. With default value setup in Table 5.3, the followings Fig.5.1(a),5.2,5.3 are the result pictures of comparison among above listed metrics in different simulation time within 100 simulation clocks time frame.

5.2.2.1 Drop rate



(a) Data Drop Rate (MB / sim clock)



(b) Data Drop Details (exclude flooding-based routing schemes, data generation rate 1000MB / sim clock)

Figure 5.1. Data Drop Results.

As Fig.5.1(a) shows, most of the routing schemes excluding flooding ones had similar data drop rate. The reason is that multiple copies of data were spread over the network in flooding based routing schemes. Given the same network resources, only limited amount of data could be served, and all extra data would be dropped due to unavailable network resources. The detailed data drop difference among the other four routing schemes can be observed in Fig.5.1(b): when the whole network was saturated (fast enough data generation rate, e.g. $1000MB/sim\ clock$), schemes that took congestion information into account during routing decision phase such as game-theory based one showed less data drop rate.

5.2.2.2 Delivery rate

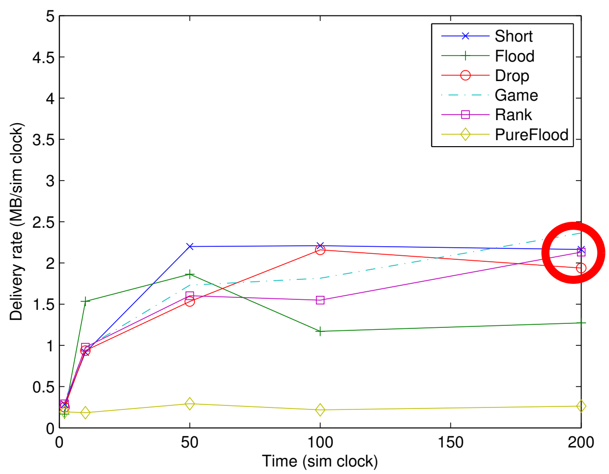


Figure 5.2. Data Delivery Rate (MB / sim clock).

As we can see in the Fig.5.2, flood routing schemes with pathfinding could reach higher delivery rate when the data generation rate was low. The reason is that several copies of data were dispensed over the whole network. The more copies the data had, the better chance they could be delivered; and these copies might travel different paths to arrive at the destination. However, when the data generation rate was high, as we know the whole network had capacity limit, all the other routing algorithms would converge to the throughput limit point as shown in the Fig.5.2. And flood routing schemes would have less delivered data, since duplicated data consumed precious network resources, which caused new data dropped due to unavailable network resources.

5.2.2.3 Delivery delay

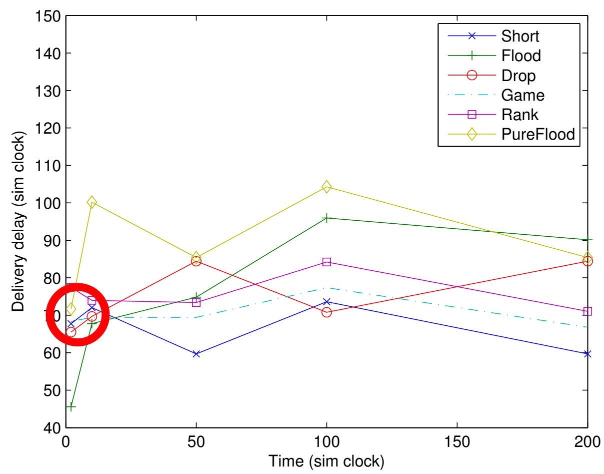


Figure 5.3. Data Delivery Delay (sim clock).

In this place-based opportunistic network, data are transferred among several places. A routing path will be a crucial part facilitating data delivery. In our routing design, pre-computed paths are the basis for most of the routing schemes except the PureFlood one, because they provide guaranteed routes for data to reach the destination. Without shortest path or pre-calculated path, data will travel around random places, which causes high delivery delay and wastes precious network resources. As shown in Fig.5.3, when the data generation rate was low, all the routing schemes with path precalculations had similar delivery delay except flooding routing scheme with pathfinding, since more data copies increased the delivery probability and consequently decreased delivery delay. However, when data generation rate was high, they presented different data delivery delay results. Except shortest path routing scheme, which always achieved shortest data delivery delay, game-theory based routing scheme had best overall delivery delay, since it took whole congestion information into account in order to allow every data flow to be served as well as reduce the data drop rate. Flooding scheme without path pre-computation showed the worst delivery delay results, since the data flowed along random paths to reach their destinations.

From above results, some valuable conclusions can be obtained: 1. A pre-calculated path is a crucial part for data delivery in this place-based opportunistic network, since data are guaranteed to be delivered if they follow the path. Without pre-computed path, data delivery rate will be low and data drop rate will be high, and the network resources could not be efficiently utilized. 2. Simulation throughput is still low compared to the theoretical capacity value, and better routing schemes are desired to fit into this network in future. 3. Game-theory based routing scheme can provide better congestion situation without sacrificing huge data delivery delay, since they utilize congestion history information to reduce data drop rate and ensure each flow has opportunity to transfer data to its ending node.

5.3 Summary

In this chapter, we proposed two routing schemes Rank and Game to fit specifically into the new place-based opportunistic networks scenario. And then designed experiments to observe how routing schemes with various knowledge about the network perform in this network. Finally, compare and analyze their results to find out what can affect the performance of routing schemes. From the simulation results, we can conclude that precalculated routing path is the basis for good data delivery, since it guaranteed the data could be delivered if they flow along the path. Besides that, we can observe that the game-theory based routing scheme can provide better congestion situation without sacrificing too much data delivery delay.

Data: Data await transfer, pedestrian history, data history

Result: Next place where data will travel to

$NextHop = P_{random}$

$DR_{best} = 1$

$Path_{best}.length = \infty$

for P_i *in* $PlaceList - P_{present}$ **do**

    existing data from $P_{present}$ to P_i $D_{exist} = \infty$

    if P_i *in* $NextPlaceList$ **then**

        update existing data amount D_{exist}

    end

    calculate P_i capacity C based on pedestrian population history N

    count number of links L to P_i

    current utilized ratio $DR_i = D_{exist}/(C/L)$

    if $DR_i >= 1$ **then**

        continue

    end

    if $DR_i > 0$ *and* $DR_i < DR_{best}$ **then**

        $DR_{best} = (DR_{exist} + datasize)/(C/L)$

        $NextHop = i$

    else if $DR_i == 0$ **then**

        if $Path_i.length < Path_{best}.length$ **then**

            $DR_{best} = (DR_{exist} + datasize)/(C/L)$

            $NextHop = i$

        end

    end

end

Algorithm 6: Data next hop selection in game-theory based routing scheme

Table 5.2. Proposed Metrics

Delivery rate	How much data are actually delivered/time unit
Drop rate	How much data are dropped/time unit
Delivery delay	How long does it take to deliver data on average
Throughput	Delivery rate, will be compared with network capacity

Table 5.3. Experiment Parameters Setup

Total places	10
Total closed-class pedestrian	500
Open arrival rate γ_i	1/5
Place stay time	$uniform(10, 20)$
Pedestrian speed	15
Buffer size	$1MB$
Message size	$100KB$
Node transmission rate	$1MB$
Map size	$1000 * 1000$
Place coordinates	$(x, y)\ x, y \in uniform(0, 1000)$
Total simulation time	400
Flooding copy	4
Top k	5

CHAPTER 6

APPLICATION RECOMMENDATION IN PLACE BASED OPPORTUNISTIC NETWORKS

6.1 Place centered application recommendation system

We introduced the concept of place based opportunistic networks in [56]. Briefly speaking, place based opportunistic networks is defined as a type of opportunistic networks where data exchanges only take place inside places. Human daily movement in this type of networks exhibits highly place-centered features: intermittent hops between places and long stop at places. In another word, people exhibit highly repetitive activity patterns in their daily life, which are often centered around certain places like restaurants, shopping centers, and banks [57, 58]. Visiting these places often incurs certain periods of waiting time, which smart devices nowadays have become the de facto means to kill painful waiting time. In this paper we design a place-centered application recommendation system that offers better experience for the users through the term attention reward we define. Briefly, attention reward is the amount of attention people pay on something during certain time period. Our proposed recommendation system considers the unique characteristics of a particular place such as its waiting time as well as waiting pattern and suggests appropriate applications with possible best reward for a user. For example, if a user is at the movie theater waiting for ordering pop corns, movie (trailer) related applications most likely will generate people's interests. Our system includes two components, namely *application pool discovery* and *application list recommendation*.

6.1.1 Application pool discovery

The first step of our recommendation scheme is to establish a pool of potentially attractive candidate applications. This pool is composed of two parts, popular application pool and place-related application pool. Applications from above two pools will be equally considered as candidates to form the final application recommendation list.

6.1.1.1 Popular applications

We conducted a 100-people survey regarding application usage during waiting periods on multiple online social networks(Facebook, WeChat and QQ), the result of which is shown in Table 6.1. We can see that "Social" is the most popular application category during waiting period, followed by reading-related categories ("News/Magazine" and "Books/Reference"), and "Game".

Table 6.1. Popular Application Categories

Social	News&Magazine	Game	Books&Reference	Others
60	28	25	14	24

From above survey results, we can observe that the most popular applications fall into the top 4 categories (84%). Briefly, we will choose top N applications from top M categories in application store to form popular application pool. Since the recommendations are conducted in mobile devices with limited computational capability and battery energy, to keep our final application pool relatively small and diverse, in our system we choose 5 as N and 4 as M to form the first part P_{popu} of final application pool P_{final}. Other numbers could also be chosen, and they will only effect the application candidates pool but not the recommendation algorithms.

6.1.1.2 Place-related applications

The second part of the applications are place-related, which will vary along with the current location of the user. To generate this set of applications, we first need to relate applications to a place. To achieve this, we assume that the place information is available to our system (e.g, if it is a movie theater). This can be easily satisfied by various location related services including Google or Foursquare. The relatedness between applications and a place is computed by joining the following three components, *category similarity, hand motion similarity and attention level similarity.*

6.1.1.2.1 Category similarity The first potential matching point between places and applications is their respective categories. The category information of a place is essentially the type of that place, and can be obtained from an online location service such as Foursquare. Given the categories of applications, we can use feature-based matching algorithms to calculate the relatedness between applications and places. For example, both the application "Flixster" and movie theaters fall into "movie" related categories, and hence they are highly related. We use the concept of WikipediaMiner [59] to measure the relatedness between two "category" words. In WikipediaMiner,

$$sr(a,b) = \frac{log(max(|A|,|B|) - log(|A \cap B|)}{log(|W|) - log(min(|A|,|B|))} \tag{6.1}$$

where a and b are the two articles in Wikipedia directly related to the two "category" words. A and B are the sets of articles in Wikipedia related to a and b

respectively. And W is the set of articles in the entire Wikipedia. In our system, the relatedness $R1$ between a place and an application is similarly defined as

$$\begin{aligned} R1 &= SimCat(Cat_{place}, Cat_{app}) \\ &= sr(Article(Cat_{place}), Article(Cat_{app})) \end{aligned} \tag{6.2}$$

6.1.1.2.2 Hand motion similarity While waiting in line at different places, owing to different corresponding place-related activities, people's hand motion exhibits different characteristics. For example, at shopping malls, people usually hold items to be purchased in one hand or hold shopping cart as they wait in the check-out line. This constrains the application to be played by only one hand, therefore applications like "Angry Birds" are not appropriate for this situation. Applications themselves also display different hand motion characteristics[60]. Therefore the previous example implies that one-hand applications could be better choice than two-hand ones in that case. Based on these facts, we define the above possible similarity of hand motion in a place and an application as the second relatedness component to capture the potential connection between them. And each waiting activity is further characterized by different sub-activities: seating, walking, and standing. Mathematically, we define this hand motion similarity as followings:

$$\begin{aligned} R2 &= SimHand(HM_{place}, HM_{app}) \\ &= Seat_{perc} + Walk_{perc} * \frac{T_{App1H}}{T_{App}} && (6.3) \\ &+ Stand_{perc} * R_{Hand} && (6.4) \end{aligned}$$

$$R_{Hand} = \begin{cases} 1.0 : T_{App2H}/T_{App} <= StandTwoHand_{perc} \\ StandTwoHand_{perc} + T_{App1H}/T_{App} : else \end{cases} \tag{6.5}$$

Briefly, we divide the hand usage of applications and activities into two-hand period, one-hand period and busy period(no hand available), and compute each period percentage in order to get final relatedness. In above formula, hand-motion similarity $R2$ depends on the activity characteristics (percentage of different activity and related hand motion percentage) at a place during waiting and hand motion characteristics while engaging in an application. In the above equation, T_{App} denotes the average session time of an application. T_{App1H} and T_{App2H} denote the time periods people use this application by one hand and two hands separately. $Seat_{perc}$, $Walk_{perc}$ and $Stand_{perc}$ denotes the percentage of time each activity occupies during waiting at that place. R_{Hand} denotes the hand motion similarity between stand activity and a selected application. Currently, we obey the following rules: 1) While seating, both hands are available. 2) While walking, only one hand is available. 3) While standing, the two-hand period is suitable for any periods for any applications; the one-hand period is only suitable for applications' one-hand period. Approaches in obtaining these percentage data will be discussed in our experiment part.

6.1.1.2.3 Attention level similarity What's more, people pay different levels of attention on different applications. Previous work [61] discussed how to divide attention levels according to human body languages. As a result, we can assign the possible attention range of each application. For example, the attention level for "Temple Run" could be $(8-10)$ due to the full concentration during this game based on our daily experience. Currently in our system, we quantify the attention level of each application according to their categories, mainly based on the attention level description in [61]. Detailed and accurate attention level assignments to each application requires tedious and long-term data collecting, which is beyond our current budget, and we plan to carry it out in future.

Besides that, the levels of attention that people pay on waiting in different places are also different. We now define the attention level while waiting at a place. Usually, there are three types of waiting queues. The first type is a real waiting line like the check-out queue in a shopping center, and we name it Mall-type waiting. The second type is a virtual waiting line like at a hospital named Hospital-type waiting. In this type of waiting, people usually pick up a number, seat down and wait. The third one is an unforeseen waiting queue in bus station. They usually stand/seat in that place for a while and frequently check the surrounding environment even when visible bus schedule and sometimes some sort of waiting time estimation notification are available. This is mainly due to unpredictable traffic situation during this waiting type.

We define the attention levels of three types of waiting in Table 6.2.

Table 6.2. Waiting Attention Level

Type	Bus-type	Mall-type	Hospital-type
Attention Level	$6-9$	$3-7$	$1-4$

With the attention levels of waiting types and applications, we define the third part relatedness $R3$ between places and applications as following:

$$R3 = SimAL(AL_{place}, AL_{app}) = \frac{|(10 - AL_{place}) \cap AL_{app}|}{|(10 - AL_{place}) \cup AL_{app}|} \tag{6.6}$$

By combining the above three relatedness $R1, R2, R3$ together, we obtain the final relatedness between applications and places as

$$Relatedness(app, place) = a_1 R1 + a_2 R2 + a_3 R3, \tag{6.7}$$

where a_1, a_2, a_3 are the coefficients. Finally, for any application with final relatedness value above threshold value λ, it will be included in the second part $P_{related}$ of final application Pool P_{final}. Right now we predefine the value of these coefficients in our experiments, and plan to tune these coefficients based on users' feedback data by machine learning techniques in future.

6.1.2 Application list recommendation

We use the relatedness value defined in last section to choose the best place for new application, and then build up the application pool P_{final}. Based on P_{final}, we are interested in finding out the best application list from the pool and recommend it to people in that place. This is a challenging task, since we need to take the unexpected actions from people into account, such as they may spend much less time on the recommended applications than the average time due to the bad or boring suggestion. To handle these issues, we model this recommendation problem as a stochastic knapsack problem. Briefly, we will build up a list of applications within fixed total time(average waiting time in a place). The total waiting time in a place is considered as the total size of the knapsack. And the attention reward and session time of each application are mapped to the value and teh cost of an item separately. Various methods like online checkin service Foursquare or [62] could be used to collect the waiting time in a place. In this work, we assume that the time spent on each application follows a normal distribution. Suppose the function of attention level along time is $f(t)$, the attention reward (accumulated attention amount) could be formulated as following equation:

$$ar|_0^t = \int_0^t f(t)dt \tag{6.8}$$

The expected attention amount yielded by application A is $E_{attention}^{app_A} = ar|_0^{E(t)}$. Each application contributes a portion of attention reward of the whole list. Besides that, we also consider the context switch cost between applications due to the time wasted during application switch. We model the task switch cost as a function of the total time needed to switch from old application to new one based on the loading time of new application and the relatedness between these two applications. The reason is that the more difference between two applications, the more time people will need to get involved into the new context. In current state, application download time is not taken into account since it depends on a lot of unpredicted factors such as mobile phone hardware and surrounding signal strength. The following is the switch cost function that we propose:

$$\begin{aligned} time(app_A, app_B) =& AppsAvgLoadT(app_B) + \\ & \beta * 2^{(1-Relatedness(app_A, app_B))} \end{aligned} \tag{6.9}$$

$$cost(app_A, app_B) = ar(app_A)|_{t_{appAend}}^{t_{appAend}+time(app_A, app_B)} \tag{6.10}$$

The first part of Equation 6.9 is the average loading time for the new application. The second part is defined as warm-up time. According to [63], the resumption time of a task from interruption is about 1250-1700ms. We suppose that the time to focus on a new task would be double of the resumption time. In another word, the minimum warm-up time β is assumed to be 2.5s. In the end, we define the total attention reward that results from a sequence S of applications as below:

$$AR(S) = \sum_{1 \leq i \leq |S|} ar_i - \sum_{1 \leq i < |S|} cost(app_i, app_{i+1}) \tag{6.11}$$

Based on above attention reward model, we propose the following two algorithms to satisfy different problem scenarios.

6.1.2.1 Greedy Heuristic Best application Sequence (GHBS)

Given a pool of appropriate application candidates generated from the application pool discovery step, we need a new solution to address the variant knapsack problem due to the cost between two sequential items. As we know, classical 0/1 knapsack problem is an NP problem [], it is easy to show the NP-hard characteristic of the new problem by just simply defining the cost between two items to be a constant all the time.

The first approach to attack the above NP-hard knapsack problem would be doing brutal force to permutate all possible sequences from the application pool and pick out the sequence that yields maximum attention reward. However, this is time-consuming and not appropriate for mobile devices even for a small amount of application pool due to their limited and precious battery energy. For example, to select 10 applications out of 20 ones would result 67 billion possible sequences, which requires over one minutes intensive calculation and is evidently not applicable. With this concern, we design a greedy heuristic approximate solution. From the practical point of view, greedy design is an appropirate choice since commonly people have tendency to terminate executing the recommendation list earlier. As a result, recommending applications with highest attention rewards will have high chance to indeed yield high attention reward in practice.

We define the following terms.

- P_{final} application pool
- T total time
- n total number of applications in P_{final}
- m the number of applications in the final recommendation list $AList_s$
- ar_i the attention reward caused by application i

- t_i the session time of application i
- $v_i = E[min\{t_i, 1\}]$ the mean truncated time of application i, if the total time is 1
- $u_{ij} = time(app_i, app_j)/T$ the switch time from app_i to app_j
- $c_{ij} = cost(app_i, app_j)$ the switch cost from app_i to app_j
- $time(AList_s) = \sum_{i \in AList_s} t_i$
- $v(AList_s) = \sum_{i \in AList_s} v_i$
- $ar(AList_s) = \sum_{i \in AList_s} ar_i$

The heuristic solution that we propose is a greedy algorithm shown in Alg. 7.

Data: A, ar_i, v_i, c_{ij}, u_{ij}, $1 \leq i, j, k \leq n$

Result: Application list $AList_s$

1. Order the applications by considering rewards by

$$\frac{ar_{i_1}}{v_{i_1}} \geq \frac{ar_{i_2}}{v_{i_2}} \geq \frac{ar_{i_3}}{v_{i_3}} \geq ...; \tag{6.12}$$

2. Select App a_{j_0} with largest value in (6.12) and add it into pool $AList_s$, $k = 0$;

3. Order the applications by considering both rewards and costs

$$\frac{ar_{i'_1} - c_{j_k i'_1}}{v_{i'_1} + u_{j_k i'_1}} \geq \frac{ar_{i'_2} - c_{j_k i'_2}}{v_{i'_2} + u_{j_k i'_2}} \geq \frac{v_{i'_3} - c_{j_k i'_3}}{ar_{i'_3} + u_{j_k i'_3}} \geq ... \tag{6.13}$$

Select $a_{j_{k+1}}$ with largest value in the sequence starts with new selected application a_{j_k} in last step based on value from (6.13) and add it into pool $AList_s$;

4. Repeat step 3 until $\sum_{a_l \in AList_s} v_{a_l} + u_{a_{l+1} a_l} >= 1$.

Algorithm 7: Greedy Heuristic Best Application Sequence (GHBS)

The time complexity of our greedy algorithm is $O(mnlogn)$. m, the expected number of applications played in this time period can be calculated by T/t_{avg}, given that t_{avg} is the average application session time. This algorithm is much faster than the original brutal force algorithms with complexity $O(n!)$. We can prove the $1/2 - approximation$ feature of GHBS which we which we will discuss later.

6.1.2.2 Average Best application Pool (ABP)

The above solution GHBS works with a strong assumption that people execute recommended applications following the list one by one. However, in real life given a list of recommended applications, it is common to observe that people have high probability to open applications in a different order due to several factors such as personal preference. For example, assume that the recommendation application list for a place is $\langle "PulseNewsReader", "AngryBirds", "WeChat" \rangle$. Originally, "Pulse News Reader" should be exectued first. However, when a 12-year old boy who is a big fan of "Angry birds" receives this list, he will probably skip the "Pulse News Reader" and open "Angry Birds" directly. In this case, the resulting attention reward will be different from the attention reward yielded by the original sequence (we will consider personal preference in recommendation in our future work). Based on this observation, besides the above approximate optimal application sequence, we are also interested in discovering the application pool that yields the highest average expected attention reward. When receiving this recommendation application pool, people have the options to use the application out of the original order, and the consequential application execution order is expected to yield the average highest attention reward compared to other application pool. Here we assume that people's application selection falls into the uniform distribution.

Given total time T and the average application session time t_{avg}, we plan to schedule $k = T/t_{avg}$ applications out of the recommended application pool. The formula to compute average reward of the application pool $P = \{app_1, app_2, app_3, ..., app_n\}$ is given by

$$Total_{avg} = \sum_{i \in P} ar_{app_i} - \frac{1}{n} * (\sum_{i,j \in P, i \neq j} c(app_i, app_j)). \tag{6.14}$$

The above formula could be used to construct the average best application pool. However, we can further improve the performance by introducing dynamic programming into the computation. It is easy to observe that the above formula can be divided into two parts, the reward part and the cost part.

$$Reward_{app_{a_1 \dots a_{n+1}}} = \sum_{a_i \in S} ar_{app_{a_i}} \tag{6.15}$$

$$Cost_{app_{a_1 \dots a_{n+1}}}$$
$$= \sum_{a_i, a_j \in S, i \neq j} (cost(app_{a_i}, app(a_j)) + cost(app_{a_j}, app_{a_i})) \tag{6.16}$$

From above Equation 6.15,6.16, we can observe the transition functions from n state to $n+1$ state and corresponding formulate a dynamic programming algorithm to compute the average best application pool. This is shown below.

Its time complexity is $O(nkC_n^k)$, and space complexity is $O(kC_n^k)$. We can prove the $1/4 - approximation$ feature of ABP which we will discuss next.

6.1.2.3 Mathematical foundation of GHBS and ABP

In this section, we discuss the approximate feature of above proposed GHBS and ABP algorithms. It is well known that attention paid on things decreases gradually along with the time when no extra stimulation is introduced into audiences as illustrated in Fig. 6.1 [1]. The periodically attention growth in this figure is caused

Data: A, ar_i, v_i, c_{ij}, u_{ij}, $1 \leq i, j, k \leq n$

Result: Application Pool $APool_s$

1. Begin from all possible combinations of basic application pool with only 2 different applications, we calculate $Reward, Cost, Total$ from above for these pools

$$\begin{aligned} & Total_{app_{i_0}, app_{i_1}} \\ =& ar_{app_{i_0}} + ar_{app_{i_1}} - \frac{c(app_{i_1}, app_{i_0}) + c(app_{i_0}, app_{i_1})}{2} \\ =& Reward_{app_{i_0,i_1}} - \frac{1}{2} * Cost_{app_{i_0,i_1}} \end{aligned} \tag{6.17}$$

2. Increase the size n of application pool by 1, and calculate $Reward, Cost, Total$ for new application pools

$$\begin{aligned} & Total_{app_{a_1 \ldots a_{n+1}}} \\ =& Reward_{app_{a_1 \ldots a_{n+1}}} - \frac{1}{n+1} * Cost_{app_{a_1 \ldots a_{n+1}}} \\ =& Reward_{app_{a_1 \ldots a_n}} + ar_{app_{n+1}} - \\ & \frac{1}{n+1} * (Cost_{app_{a_1 \ldots a_n}} + \\ & sum_{t \in a_1 \ldots a_n}(c(app_t, app_{n+1}) + c(app_{n+1}, app_t))) \end{aligned} \tag{6.18}$$

3. Repeat step 2 until $n >= k$.

Algorithm 8: Average Best Application Pool (ABP)

by new stimulations, such as the intermediate conclusions shown in the figure. The quickly rising attention level in the end of the curve is common since people usually notice themselves that they can end this activity (talk/presentation) and leave soon.

6.1.2.3.1 Approximate ratio of GHBS We assume A^* is the optimal solution to the original 0/1 knapsack problem, and A_g is the solution generated by GHBS. By

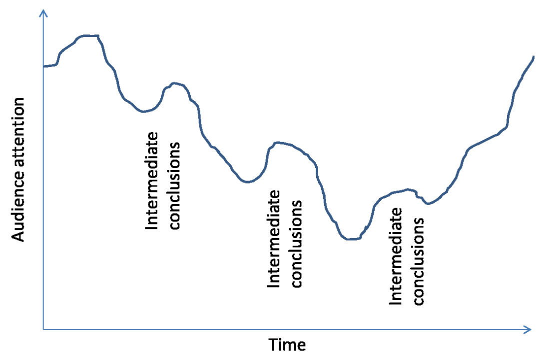


Figure 6.1. Attention Graph Along Time [1].

constructing a fractional knapsack problem based on the above 0/1 knapsack problem, we can conveniently obtain the optimal solution FA^* by ordering the items according to their $TotalReward/Time$ ratio and picking up items greedily. Suppose there are k items in the solution FA^*, A_{g_i}, FA_i^*, A_i^* represent ith item in corresponding solutions, and $value(x), cost(x, y)$ indicate the attention reward and cost switch of a solution or items. Then we have the following equations and inequations

$$A_{g_i} = FA_i^*, i < k \tag{6.19}$$

$$\begin{aligned} &Total(A^*) \\ &= value(A_1^*) + \sum_{1<i<k-1} -cost(A_i^*, A_{i+1}^*) + value(A_{i+1}^*) \\ &< value(FA_1^*) + \\ &\sum_{1<i<k-1} -cost(FA_i^*, FA_{i+1}^*) + value(FA_{i+1}^*) \\ &<= Total(FA^*) \end{aligned} \tag{6.20}$$

$$= Total(A_g) - cost(A_{g_{k-1}}, FA_k^*) + value(FA_k^*). \tag{6.21}$$

In our scenario, the item switch cost (applications switch time) is far less than the length of waiting period. We have

$$\begin{aligned} -cost(A_{g_{k-1}}, FA_k^*) + value(FA_k^*) &< Total(FA_k^*) \\ &< 1/2 * Total(FA^*) \\ &< Total(A_g) \end{aligned} \tag{6.22}$$

Therefore, based on 6.21 and 6.22, we can obtain the following inequations and prove that GBHS is $1/2 - approximate$.

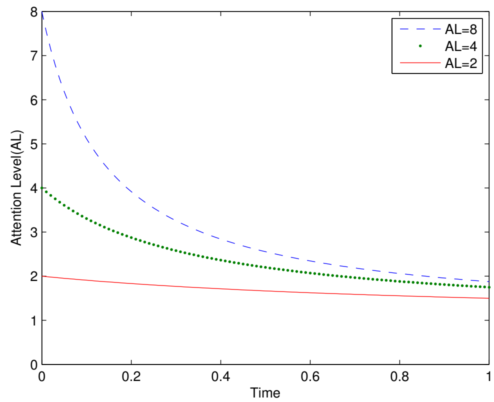


Figure 6.2. Attention Function with Different Original Attention Level.

$$\begin{aligned} Total(A^*) &<= Total(A_g) + Total(A_g) \\ &= 2 * Total(A_g) \end{aligned} \quad (6.23)$$

$$(6.24)$$

6.1.2.3.2 Approximate ratio of ABP Based on the above attention level, we could simplify it without intermittent stimulation and apply our 10-level attention definition in previous sections. By considering the average attention level for each attention range(for example using 8 to represent $[7, 9]$), we have the Fig.6.2.

$$
\begin{aligned}
value(A^*_{avg}) \quad & <= 4 * value(A_{abp_{avg}}) && (6.25)\\
Total(A^*) \quad & = value(A^*) - cost(A^*) \\
& = k * value(A^*_{avg}) - k * cost(A^*_{avg}) \\
& = k * value(A^*_{avg}) - k * \frac{\alpha}{\beta} * value(A^*_{avg}) \\
& < 4k * (1 - \frac{\alpha}{\beta}) * value(A_{abp_{avg}}) \\
& = 4 * Total(A_{abp}) && (6.26)
\end{aligned}
$$

As we know, the derivative of reward function is a monotonic decreasing function. From the above figure and our attention level definition, it is easy to observe that the ratio between the area below "–" curve and the area below "-" from time 0 to 1 is less than 4. This ratio is less than 2 between areas covered by "–" and "+" curves as well as "+" and "-" curves. Suppose A^* always takes the highest reward-yielding applications, and A_{abp} always takes the lowest reward-yielding applications. The average applications switch time is α, and the average application session time is β. We also assume $value(A^*_{avg}), cost(A^*_{avg})$ represents the average attention reward and cost for solution A^*, similarly for $A_{abp_{avg}}$. We can obtain the the above equations and inequations. Based on 6.26, we can conclude that ABP is at least 1/4-approximate.

6.2 Simulation and field study

We perform simulation and field study in this section.

6.2.1 Simulation

All the default value of parameters chosen for this simulation (time unit is the simulation clock) is similar as [56]. There are two types of data in this network, one is the activity feature data about each place, and the other one is the recommendation results. Every mobile phone in this network exchanges first type of data with each other, execute average aggregation on data and wait for next encounter to exchange again. Basically, gossip aggregation protocols[64, 53] are deployed in the network in order to provide aggregation service. The second type of data is considered successfully delivered when they are carried to their own source place and the memory they occupied in the host mobile device will be released. And we choose 1/3 as the default value for a_1, a_2, a_3, 0.3 as the threshold λ value. Tuning on these values is planned on future version of system with machine learning technique enabled.

Besides above parameters setup, we utilize the data from previous works [65, 60] to preprocess some important data that require a lot of time to collect, such as the application session time, the application hand usage statistic data and activity hand usage statistic data. Due to space limitation, we will omit the details. Due to the distributed characteristic of our recommendation system, it is possible for people in a place to receive multiple application recommendation list. In our system, we have the following mechanisms for selection 1) List with highest reward always wins, and 2) List with higher reward has better chance to win.

We are interested in results on final attention rewards of application lists and pools as well as the activity feature data aggregation process in the whole network. As the second row of Figure 6.3 show, our GHBS algorithm achieves at least half of the final reward when compared to the optimal solution (brutal force optimal best sequence) in all three waiting type scenarios. And ABP algorithm also produces high attention reward. In the first row, the execution required for GHBS is dramatically less than the optimal one and the ABP. Additionally, the results show large execution time deviation of the optimal best sequence and the ABP algorithm in different simulation runs. As we analyze, this is due to the the number of applications in application pool (43, 16, 34 for each setup) discovered in different simulation setup

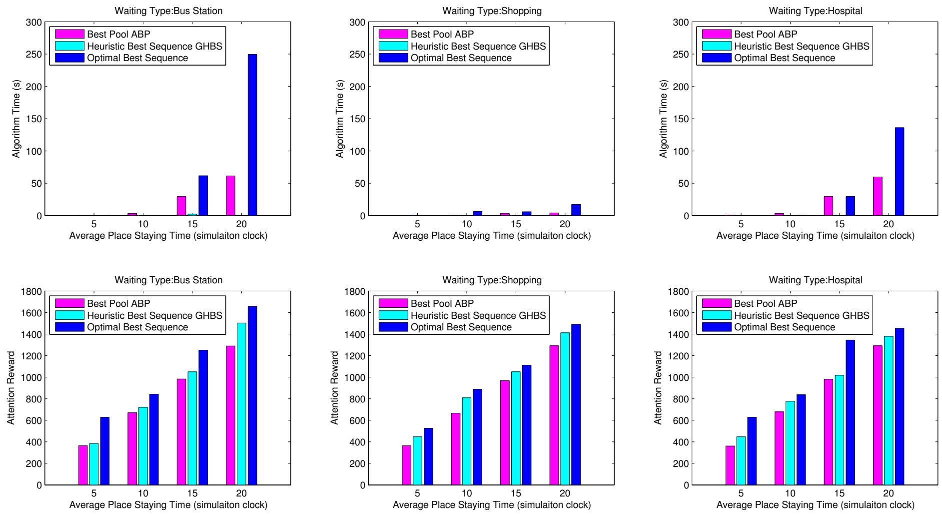


Figure 6.3. Recommendation Algorithms Time and Attention Rewards.

runs. The time complexity of the optimal best sequence and the ABP algorithm greatly depend on the size of application candidates pool.

Figure 6.4 and 6.5 illustrate the convergence of activity feature information (walk/seat/stand proportion) about a place under different parameter setups along the time. We choose a place with hospital waiting type to exemplify the aggregation process. For hospital waiting type, we preset the activity ratio as walk:stand:seat (0.2 : 0.4 : 0.4) in the simulation, and also have specific setups for other waiting types. The practical value of above data could be obtained by turning help to sensors like accelerometer on mobile devices or using similar ways and analyzing signal strength pattern (stable and transition status) as in [62]. The first fact we can observe is that the activity feature aggregation values converge around the setup values after some time in all sub-figures. Secondly, given different pedestrian moving speed (5/simclock and 15/simclock) but the same other parameters, it is easy to observe that in all sub-figures of Figure 6.5 the activity aggregation results under 15/simclock moving

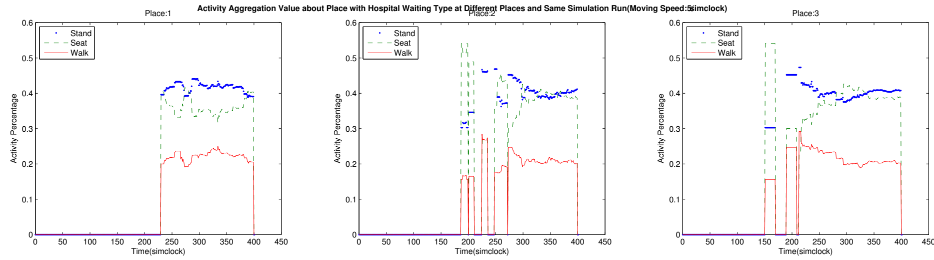


Figure 6.4. Activity Information Convergence Process (Value in Different Places within Same Simulation Run, Speed 5/simclock).

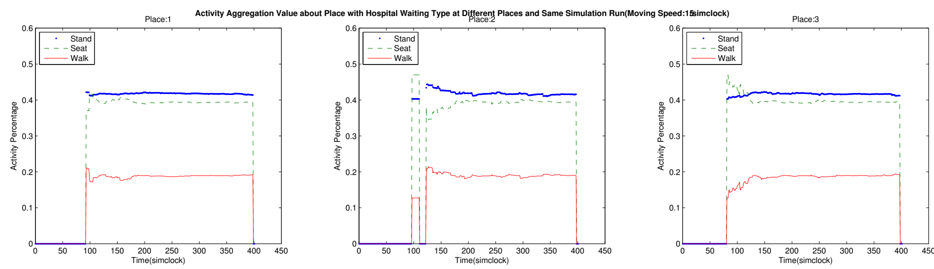
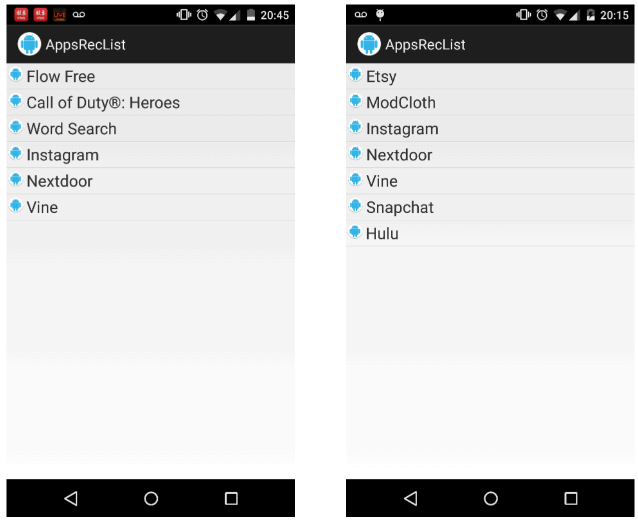


Figure 6.5. Activity Information Convergence Process (Value in Different Places within Same Simulation Run, Speed 15/simclock).

speed quickly converge and oscillate in fairly small range compared to the zigzag slow convergence curve for 5/simclock moving speed in all sub-figures of Figure 6.4. These facts imply that the faster the pedestrians move in the network, the faster the activity information will converge and stabilize. In the end, we can observe that in all different places the activity feature aggregation values about the same place converge, which indicates that these activity feature data are well distributed in the whole network.

6.2.2 Field study

We have implemented the recommendation system in real mobile devices and conducted field studies in real life. We choose Nexus 5 and Nexus 7 as our test



(a) ABP results in Walmart

(b) GHBS results in Walmart

Figure 6.6. On-field Application Recommendation Results.

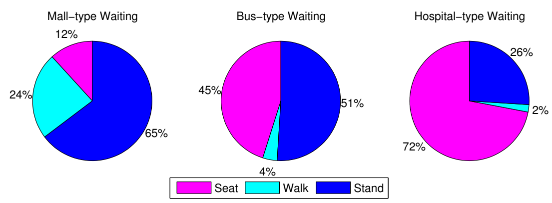


Figure 6.7. Activity Information of 3 Different Types of Waiting.

devices and use WiFi direct as underlying P2P communication method. Fig.6.6 are two screenshots of resulting application pool and application sequence while waiting in Walmart. In the field studies, firstly we are interested in energy consumption data since large scale field experiments are hard to conduct due to the limitation we face. With 1/min localization frequency and 1/min activity sensing frequency setup, the localization(81%) and activity sensing(16%) components consumed most of the energy during the lifetime of recommendation system. For balancing between energy consumption and sensing accuracy, please refer to our previous work in [66].

Table 6.3. Sample Applications in Fig.6.6

Category	Applications
Game	Flow Free, Call of Duty: Heroes, Word Search
Social	Instagram, Nextdoor, Snapchat, Vine
Media	Hulu
Shopping	ModCloth, Esty

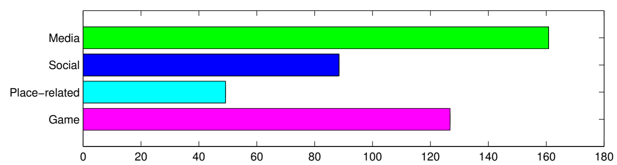


Figure 6.8. The Usage Time of Different Application Categories.

Besides the energy consumption, we also collected some on-field usage statistics to provide us with some insights given that no large-scale experiments have been conducted at current stage. In our 5 people on-field study (in local Walmart, hospital, train station), on average 3.5 recommended applications were executed given that the waiting period was around 12 minutes. Sample participants had high tendency to execute the applications out of the original order with high probability to run the game applications at first (5 out of 5). As we can observe from above sample recommendation results in Fig.6.6 and Table 6.3, recommended applications include those from "Social" and "Games" categories, which aligns with our everyday experience. Besides that, those applications with high category relatedness (Esty and ModCloth in Walmart, both of them fall into "Shopping" category) were successfully recommended; and they indeed attracted people's attention to some extent (executed 2 out of 2). We also plotted the activity statistics figure of different types of waiting Fig. 6.7 and application usage time figure Fig. 6.8. However, the overall results such as application execution ratio were still undesirable. There are several factors that could cause current on-field results, inlcuding the users' personal preference (recommendation results aim at all people without discrimination at the same place), the quality of the user interface of our system and the content of the recommended applications. Furthermore, people's reaction to the application recommendation list can be crucial feedback data not considered now. In future, we plan to introduce machine learning technique into our system toward this end.

6.3 Summary

In this paper, we have proposed a new application recommendation system that utilizes human activity information in different places. Specifically, we have designed a approximate greedy heuristic best attention reward sequence algorithm GHBS and a best expected attention reward pool algorithm ABP to construct application recommendation list. Our simulation and field studies show the feasibility of our recommendation system.

REFERENCES

[1] J. Hartley and I. K. Davies, "Note-taking: A critical review," *Innovations in Education & Training International*, vol. 15, no. 3, pp. 207–224, 1978. [Online]. Available: http://dx.doi.org/10.1080/0033039780150305

[2] (2011) How the smartphone is killing the pc. [Online]. Available: http://echidnainc.com/five-critical-ux-app-analytic-metrics/

[3] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. S. Peh, and D. Rubenstein, "Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with zebranet," *SIGOPS Oper. Syst. Rev.*, vol. 36, no. 5, pp. 96–107, Oct. 2002.

[4] A. Pentland, R. Fletcher, and A. Hasson, "Daknet: rethinking connectivity in developing nations," *Computer*, vol. 37, no. 1, pp. 78–83, 2004.

[5] A. Doria, "Providing connectivity to the saami nomadic community," in *In Proc. 2nd Int. Conf. on Open Collaborative Design for Sustainable Innovation*, 2002.

[6] S. Nelson, M. Bakht, and R. Kravets, "Encounter-based routing in dtns," in *INFOCOM 2009, IEEE*, 2009, pp. 846–854.

[7] C. Boldrini, M. Conti, J. Jacopini, and A. Passarella, "Hibop: a history based routing protocol for opportunistic networks," in *World of Wireless, Mobile and Multimedia Networks, 2007. WoWMoM 2007. IEEE International Symposium on a*, 2007, pp. 1–12.

[8] P. Gupta and P. Kumar, "The capacity of wireless networks," *Information Theory, IEEE Transactions on*, vol. 46, no. 2, pp. 388–404, 2000.

[9] M. Grossglauser and D. Tse, "Mobility increases the capacity of ad hoc wireless networks," *Networking, IEEE/ACM Transactions on*, vol. 10, no. 4, pp. 477–486, 2002.

[10] Y. Wang, J. Lin, and M. Annavaram, "A framework of energy efficient mobile sensing for automatic user state recognition," in *Proceedings of the 7th international conference on Mobile systems, applications, and services. MobiSys '09*, 2010, pp. 179–192.

[11] K. Lin, A. Kansal, D. Lymberopoulos, and F. Zhao, "Energy-accuracy trade-off for continuous mobile device location," in *Proceedings of the 8th international conference on Mobile systems, applications, and services. MobiSys '10*, 2010, pp. 285–298.

[12] I. Constandache, S. Gaonkar, M. Sayler, R. R. Choudhury, and L. Cox, "Enloc: Energy-efficient localization for mobile phones," in *INFOCOM 2009*, 2009, pp. 2716–2720.

[13] Z. Zhuang, K.-H. Kim, and J. P. Singh, "Improving energy efficiency of location sensing on smartphones," in *Proceedings of the 8th international conference on Mobile systems, applications, and services. MobiSys '10*, 2010, pp. 315–330.

[14] H.-Y. Wang, V. Zheng, J. Zhao, and Q. Yang, "Indoor localization in multi-floor environments with reduced effort," in *IEEE International Conference on Pervasive Computing and Communications (PerCom '10)*, 2010, pp. 244–252.

[15] A. Taheri, A. Singh, and A. Emmanuel, "Location fingerprinting on infrastructure 802.11wireless local area networks (wlans) using locus," in *Proceedings. 29th Annual IEEE International Conference on Local Computer Networks (LCN 2004)*, 2004, pp. 676–683.

[16] E. Chan, G. Baciu, and S. Mak, "Using wi-fi signal strength to localize in wireless sensor networks," in *WRI International Conference on Communications and Mobile Computing (CMC '09)*, 2009, pp. 538–542.

[17] D. L. Lee and Q. Chen, "A model-based wifi localization method," in *Proceedings of the 2nd international conference on Scalable information systems*, vol. 304, 2007.

[18] M. B. Kjærgaard, J. Langdal, T. Godsk, and T. Toftkjær, "Entracked: Energy-efficient robust position tracking for mobile devices," in *Proceedings of the 7th international conference on Mobile systems, applications, and services. MobiSys '09*, 2009, pp. 221–234.

[19] M. Batty and P. Kyaw, "Vector-based location finding for context-aware campus," in *Fifth International Conference on Wireless and Mobile Communications 2009, ICWMC '09*, 2009, pp. 116–121.

[20] J. Astrain, J. Villadangos, J. Garitagoitia, J. G. de Mend'ivil, and V. Cholvi, "Fuzzy location and tracking on wireless networks," in *Proceedings of the 4th ACM international workshop on Mobility management and wireless access (MobiWac '06)*, 2006, pp. 84–91.

[21] M. Azizyan, I. Constandache, and R. R. Choudhury, "Surroundsense: Mobile phone localization via ambience fingerprinting," in *Proceedings of the 15th annual international conference on Mobile computing and networking (MobiCom'09)*, 2009, pp. 261–272.

[22] E. Miluzzo, N. D. Lane, K. Fodor, R. P. H. Lu, and M. M. S. B. E. Z. T. Campbell, "Sensing meets mobile social networks: The design,implementation and evaluation of the cenceme application," in *Proceedings of the 6th ACM conference on Embedded network sensor systems. SenSys '08*, 2008, pp. 337–350.

[23] K. Wu, J. Xiao, Y. Yi, M. Gao, and L. Ni, "Fila: Fine-grained indoor localization," in *INFOCOM, 2012 Proceedings IEEE*, 2012, pp. 2210–2218.

[24] L. R. Ford and D. R. Fulkerson, "Maximal flow through a network," *Canadian Journal of Mathematics*, vol. 8, no. 3, pp. 399–404, 1956.

[25] Y.-C. Chen, J. Kurose, and D. Towsley, "A mixed queueing network model of mobility in a campus wireless network," in *INFOCOM, 2012 Proceedings IEEE*, 2012, pp. 2656–2660.

[26] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott, "Pocket switched networks: Real-world mobility and its consequences for opportunistic forwarding," Tech. Rep., 2005.

[27] H. Yang, M. G. Bell, and Q. Meng, "Modeling the capacity and level of service of urban transportation networks," *Transportation Research Part B: Methodological*, vol. 34, no. 4, pp. 255 – 275, 2000.

[28] P. Jacquet, B. Mans, and G. Rodolakis, "On space-time capacity limits in mobile and delay tolerant networks," in *INFOCOM, 2010 Proceedings IEEE*. IEEE, 2010, pp. 1–9.

[29] D. E. Knuth, "A generalization of dijkstra's algorithm," *Information Processing Letters*, vol. 6, no. 1, pp. 1–5, 1977.

[30] Z. Subing and L. Zemin, "A qos routing algorithm based on ant algorithm," in *Local Computer Networks, 2000. LCN 2000. Proceedings. 25th Annual IEEE Conference on*, 2000, pp. 574–578.

[31] A. N. Asokan, R. and A. Nivetha, "A swarm-based distance vector routing to support multiple quality of service (qos) metrics in mobile adhoc networks," *Journal of Computer Science*, vol. 3, no. 1, pp. 700–707, 2007.

[32] R. Ramanathan, R. Hansen, P. Basu, R. Rosales-Hain, and R. Krishnan, "Prioritized epidemic routing for opportunistic networks," in *Proceedings of*

the 1st international MobiSys workshop on Mobile opportunistic networking, ser. MobiOpp '07. New York, NY, USA: ACM, 2007, pp. 62–66. [Online]. Available: http://doi.acm.org/10.1145/1247694.1247707

[33] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and wait: an efficient routing scheme for intermittently connected mobile networks," in *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, ser. WDTN '05. New York, NY, USA: ACM, 2005, pp. 252–259. [Online]. Available: http://doi.acm.org/10.1145/1080139.1080143

[34] Z. Fu, P. Zerfos, H. Luo, S. Lu, L. Zhang, and M. Gerla, "The impact of multihop wireless channel on tcp throughput and loss," in *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, vol. 3, 2003, pp. 1744–1753 vol.3.

[35] K. Ramakrishnan and S. Floyd, "A proposal to add explicit congestion notification (ecn) to ip," United States, Tech. Rep., 1999.

[36] M. Xie, L. V. Lakshmanan, and P. T. Wood, "Breaking out of the box of recommendations: From items to packages," in *Proceedings of the Fourth ACM Conference on Recommender Systems*, ser. RecSys '10. New York, NY, USA: ACM, 2010, pp. 151–158. [Online]. Available: http://doi.acm.org/10.1145/1864708.1864739

[37] M. Xie, L. V. S. Lakshmanan, and P. Wood, "Comprec-trip: A composite recommendation system for travel planning," in *Data Engineering (ICDE), 2011 IEEE 27th International Conference on*, April 2011, pp. 1352–1355.

[38] F. Lorenzi, A. Bazzan, and M. Abel, "An architecture for a multiagent recommender system in travel recommendation scenarios," vol. 1, pp. 88–91, 2006.

[39] Q. Liu, E. Chen, H. Xiong, Y. Ge, Z. Li, and X. Wu, “A cocktail approach for travel package recommendation,” *Knowledge and Data Engineering, IEEE Transactions on*, vol. 26, no. 2, pp. 278–293, Feb 2014.

[40] Q. Liu, Y. Ge, Z. Li, E. Chen, and H. Xiong, “Personalized travel package recommendation,” in *Data Mining (ICDM), 2011 IEEE 11th International Conference on*, Dec 2011, pp. 407–416.

[41] C. Ashcraft and R. G. Grimes, “Spooles: An object-oriented sparse matrix library.” in *PPSC*, 1999.

[42] T. Kanungo, T. Kanungo, and T. Kanungo, “An efficient k-means clustering algorithm: Analysis and implementation,” *IEEE Transactions on Pattern Analysis AND Machine Intelligence*, vol. 24, no. 7, pp. 881–892, 2002.

[43] I. Trestian, S. Ranjan, A. Kuzmanovic, and A. Nucci, “Googling the internet: Profiling internet endpoints via the world wide web,” *IEEE/ACM Transactions on Networking*, vol. 18, no. 2, pp. 666–679, 2010.

[44] D. Tosh and B. Profitt. Elgg - open source social networking platform. http://www.elgg.org/.

[45] K. Kaemarungsi, “Distribution of wlan received signal strength indication for indoor location determination,” in *1st International Symposium on Wireless Pervasive Computing*, 2006.

[46] X. Wu, D. H. Bennett, K. Lee, D. L. Cassady, B. Ritz, and I. Hertz-Picciotto, “Longitudinal variability of time-location/activity patterns of population at different ages: a longitudinal study in california,” *Environmental Health*, vol. 10, no. 1, p. 80, 2011.

[47] B. Liu, Z. Liu, and D. Towsley, “On the capacity of hybrid wireless networks,” in *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, vol. 2, 2003, pp. 1543–1552 vol.2.

[48] E. Modiano, D. Shah, and G. Zussman, "Maximizing throughput in wireless networks via gossiping," in *Proceedings of the joint international conference on Measurement and modeling of computer systems*, ser. SIGMETRICS '06/Performance '06, 2006, pp. 27–38.

[49] J. E. Aronson, "A survey of dynamic network flows," *Ann. Oper. Res.*, vol. 20, no. 1-4, pp. 1–66, Aug. 1989.

[50] G. D. Glockner, G. L. Nemhauser, and C. A. Tovey, "Dynamic network flow with uncertain arc capacities: Decomposition algorithm and computational results," *Comput. Optim. Appl.*, vol. 18, no. 3, pp. 233–250, Mar. 2001.

[51] T. H. Cormen, C. Stein, R. L. Rivest, and C. E. Leiserson, *Introduction to Algorithms*, 2nd ed. McGraw-Hill Higher Education, 2001.

[52] S. Diggavi, M. Grossglauser, and D. Tse, "Even one-dimensional mobility increases ad hoc wireless capacity," in *Information Theory, 2002. Proceedings. 2002 IEEE International Symposium on*, 2002, pp. 352–.

[53] D. Kempe, A. Dobra, and J. Gehrke, "Gossip-based computation of aggregate information," in *Foundations of Computer Science, 2003. Proceedings. 44th Annual IEEE Symposium on.* IEEE, 2003, pp. 482–491.

[54] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web." Stanford InfoLab, Technical Report 1999-66, November 1999, previous number = SIDL-WP-1999-0120. [Online]. Available: http://ilpubs.stanford.edu:8090/422/

[55] J. Nash, "Non-cooperative games," *Annals of Mathematics*, vol. 54, no. 2, pp. pp. 286–295, 1951. [Online]. Available: http://www.jstor.org/stable/1969529

[56] Y. Liu, S. Lu, and Y. Liu, "Capacity of place based opportunistic networks," in *Mobile Ad-hoc and Sensor Networks (MSN), 2013 IEEE Ninth International Conference on.* IEEE, 2013, pp. 1–8.

[57] D. H. Maister, *The psychology of waiting lines.* Harvard Business School, 1984.

[58] S. Taylor, "Waiting for service: the relationship between delays and evaluations of service," *The Journal of Marketing*, pp. 56–69, 1994.

[59] I. Witten and D. Milne, "An effective, low-cost measure of semantic relatedness obtained from wikipedia links," in *Proceeding of AAAI Workshop on Wikipedia and Artificial Intelligence: an Evolving Synergy, AAAI Press, Chicago, USA*, 2008, pp. 25–30.

[60] A. Karlson, B. Bederson, and J. Contreras-Vidal, "Understanding single-handed mobile device interaction," *Handbook of research on user interface design and evaluation for mobile technology*, pp. 86–101, 2006.

[61] B. Gollan, B. Wally, and A. Ferscha, "Automatic human attention estimation in an interactive system based on behaviour analysis," *Proc. EPIA 2011*, 2011.

[62] Y. Wang, J. Yang, Y. Chen, H. Liu, M. Gruteser, and R. P. Martin, "Tracking human queues using single-point signal monitoring," in *Proceedings of the 12th Annual International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '14. New York, NY, USA: ACM, 2014, pp. 42–54. [Online]. Available: http://doi.acm.org/10.1145/2594368.2594382

[63] G. Mark, D. Gudith, and U. Klocke, "The cost of interrupted work: More speed and stress," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '08. New York, NY, USA: ACM, 2008, pp. 107–110. [Online]. Available: http://doi.acm.org/10.1145/1357054.1357072

[64] M. Jelasity, A. Montresor, and O. Babaoglu, "Gossip-based aggregation in large dynamic networks," *ACM Transactions on Computer Systems (TOCS)*, vol. 23, no. 3, pp. 219–252, 2005.

[65] M. Böhmer, B. Hecht, J. Schöning, A. Krüger, and G. Bauer, "Falling asleep with angry birds, facebook and kindle: a large scale study on mobile application

usage," in *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services.* ACM, 2011, pp. 47–56.

[66] Y. Liu, S. Lu, and Y. Liu, "Coal: Context aware localization for high energy efficiency in wireless networks," in *Wireless Communications and Networking Conference (WCNC), 2011 IEEE*, March 2011, pp. 2030–2035.

BIOGRAPHICAL STATEMENT

Yanliang Liu was born in Nanping Fujian China in 1985. He received his B.S. degree from Sun Yat-sen University in 2007 and his Ph.D. degrees from The University of Texas at Arlington in 2015 respectively, all in Computer Science. From 2009 to 2015, he was with the department of Computer Science and Engineering at the University of Texas at Arlington as a research and teaching assistant in Center for Research in Wireless Mobility and Networking (CReWMaN) and Wireless Networks and Systems (WINS) Labs. His current research interest is in the area of Mobile Computing and Mobile Recommendation System. He is a member of several IEEE societies.