A NOVEL SCHEME FOR CONTACT PREDICTIONS IN OPPORTUNISTIC NETWORKS

by

SUJOY KUMAR BHATTACHARYA

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

MASTER OF SCIENCE IN COMPUTER SCIENCE

THE UNIVERSITY OF TEXAS AT ARLINGTON

December 2012

ACKNOWLEDGEMENTS

I would first like to thank Dr. Kumar for accepting me under his supervision and supporting , encouraging me to pursue research. I would also like to thank Dr. Matthew Wright and Dr. Yonghe Liu for agreeing to be a part of my committee.

I would like to thank all the members of the faculty of the CSE department from whom I have learnt a lot. Some of the courses that I have taken at UTA has really broadened my perspective on theoretical as well as practical aspects of Computer Science and my love for the subject has increased .

I would like to thank all of my friends at UTA for making this journey even more enjoyable. Finally I would like to thank my mother, my elder sister, my brother in law and last but not the least my wife Sukalpa .Without their support, I could not have made this journey alone.

November 21, 2012

ABSTRACT

A NOVEL SCHEME FOR CONTACT PREDICTIONS IN

OPPORTUNISTIC NETWORKS

Sujoy Kumar Bhattacharya, M.S

The University of Texas at Arlington, 2012

Supervising Professor: MOHAN KUMAR

In the opportunistic network (ON) paradigm information is exchanged between two devices as they encounter each other. For such information exchange to take place the devices must know about the presence of other devices in the neighborhood.

A very fundamental problem in ONs is to be able to predict the occurrence of a future opportunistic contacts that are highly dynamic. An accurate predictor which takes into account the long time history switches to the data transfer mode can benefit from multiple objectives. Such a predictor switched to the data transfer mode from idle mode in anticipation of a contact. Also it maximizes the number of opportunistic contacts while spending minimal energy.

In this thesis, we have designed a predictive framework that uses data mining methodologies to accurately predict opportunistic contacts. For evaluation of the scheme, we have used Bluetooth traces collected by University of Illinois at Urbana Champaign, using their movement (UIM) framework using Google Android phones for a period of 3 weeks[14].Extensive simulation of our scheme using these real life traces show that the precision and recall values are close to 50% higher compared to the previous schemes. Also the energy usage, is 35% lower for KFP making it an attractive option for predicting opportunistic contacts,

to obtain efficient routing as well as swift information dissemination in ONs in an energy efficient manner.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

Figure Page

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Opportunistic Networking (ON) is a networking paradigm where wireless devices , primarily mobile, interacts with other devices in their neighborhood . As the mobile devices move about in its environment it meets with different devices. It is a highly dynamic and uncertain environment in which these devices operate as ON does not assume any dependence on pre-existing networking infrastructure or prior knowledge of the location of the devices.

In an ON, when a connection is established between two devices, data exchange takes place from one device to the other. In our work we consider smartphones/PDA's carried by humans as the devices, hereafter simply referred to as devices. In the future, a device would imply the above mentioned devices. Modern devices are increasingly equipped with multiple network interfaces with complementary characteristics. WiFi, Blue-tooth, cellular networks and NFC are examples[13]. In this work we consider only Blue-tooth based opportunistic contacts.

Data transfer is however an energy hungry process. When two devices are exchanging data, the energy consumption for both is many times more than when no data is exchanged. As in traditional ONs there is typically no knowledge of the location of the devices more often than not the devices are in the discovery mode of Bluetooth when no device is around as a result wasting precious energy and lowering the battery life. An accurate predictor of future contacts, will thus maximize the number of opportunistic contacts at the same time extending the battery life of the device.

In this work, we exploit the regularity of the human walking pattern to come up with a predictive scheme. Lately, various measurement studies of human walk traces have discovered several significant patterns of human mobility. Data exchange is an energy hungry process.

When two devices are exchanging data, the energy consumption for both is many times more than when no data is exchanged. In traditional ONs there is typically no knowledge of the location of the devices more often than not the devices are in the discovery mode of Bluetooth when no device is around as a result wasting precious energy and lowering the battery life. An accurate predictor of future contacts, that utilizes the regularity of human walking patterns, will thus maximize the number of opportunistic contacts at the same time extending the battery life of the device.

To facilitate this prediction scheme we propose K-Fold Predictor(KFP). Each device can operate in two possible modes, the data transfer/scan mode or the idle mode. In the data transfer mode a device scans its neighborhood looking for other devices thereby spending considerable energy. In the idle mode the device does not look for any such device thus spending no extra energy. Our scheme, KFP has two goals:

I. To switch to the data transfer mode, when the chance of encountering a device in the neighborhood is high.

II. To maximize the number of opportunistic contacts, while spending minimal energy of the device.

To assess the viability of KFP, simulations were conducted. These simulations made use of experimentally collected data traces to represent a realistic opportunistic networking environment. Through various simulation results we show that collecting historical data of the number of contacts at a given time interval is able to predict the contact probability with considerable accuracy.

In this thesis, Chapter 2 gives an introduction to opportunistic networking and a discussion of the existing schemes for future contact prediction. Chapter 3 gives an overview of the theory and algorithms used in KFP. Chapter 4 discusses simulation and analysis and Chapter 5 includes a discussion and concluding remarks.

CHAPTER 2

BACKGROUND

2.1 Opportunistic Networks

An opportunistic network is created when mobile wireless devices come within transmission range of each other and establish a fully connection between them and exchange information. No other networking infrastructure is assumed for this data exchange to take place. Opportunistic networks show great potential in increasing the utility of wireless devices by allowing them to work with other wireless platforms in their immediate vicinity. The capability of any single node increases when that node can collaborate with other wireless devices to accomplish more collectively than any single device could accomplish on its own.

The duration of an opportunistic network is extremely varied. These temporary networks can exist for hours at a time with minimal interruption or it can last for a few seconds. The common factor in both cases is that the wireless devices have no prior knowledge regarding the duration of the Opportunistic contact. If A wireless device wishes to sense other devices around it, then it has to switch to a more energy hungry mode. Without any guideline as to when to look for other devices results in a much higher energy consumption and does not give any guarantee to the number of actual contacts made during the operating lifetime of the device. In many situations the device would be energy depleted looking for opportunistic contacts when there are no other devices around where as if it had preserved that energy then it could have established opportunistic contacts in future.

Human movement is highly regular[1][2].None of the previous strategies[3][4] take into account the regularity in human movement. Instead they rely on the local history (of the order of minutes to decide whether to switch to the more energy hungry mode of scanning mode or

not. Their algorithm works on the intuition that if a person is close to an Opportunistic contact then it

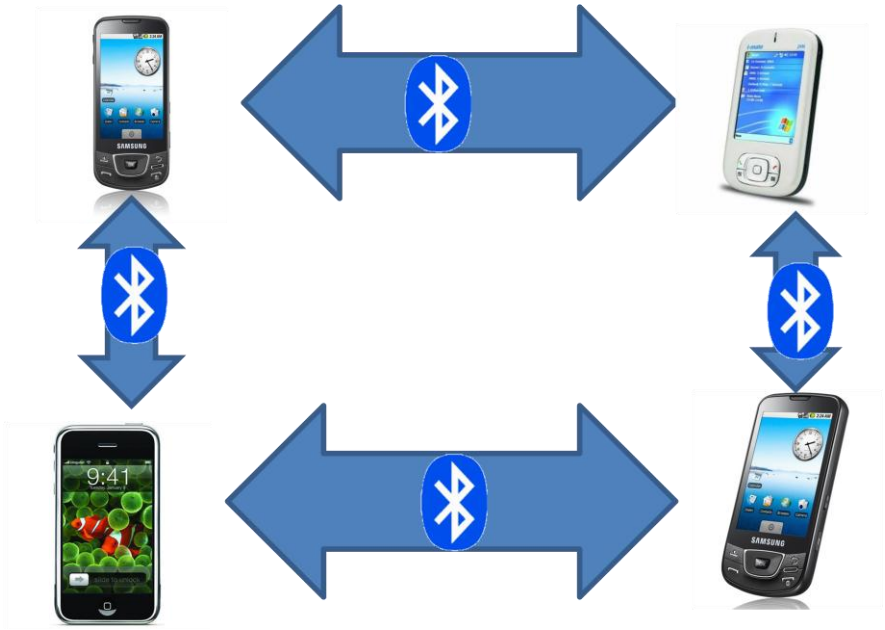

Figure 2.1 A sample opportunistic network of smartphones using Buetooth

is more probable that he will be encountering more such contacts in the near future. Now while this scheme works for some cases it does not take into account the long term contact history[7].As a result wastes a lot of energy in scanning for devices in those intervals where there are no devices in the neighborhood.

In [5] by using user mobility as a network transport mechanism, devices are intelligently routed latency-insensitive packets using power-efficient shortrange radio. Such a network shows the communication capability where no network infrastructure exists, or extendthe reach of established infrastructure.

In[6][8] the data transfer opportunities between wireless devices carried by humans is studied with the help of synthetic human mobility models. We observe that the distribution of the

intercontact time (the time gap separating two contacts between the same pair of devices) may be well approximated by a power law over the range. This observation is then utilized by to theoreticaly estimate the number of missed contacts and contact duration[3].

Our approach adopts a data mining methodology and uses the notion of true positives, false positives , true negatives and false negatives similar to that in[13].In [13] a system that predicts the availability of the Wi-Fi connectivity by using a combination of Bluetooth contact-patterns and cell-tower information. This allows the device to intelligently switch the Wi-Fi interface on only when there is Wi-Fi connectivity available, thus avoiding the long periods in idle state and significantly reducing the the number of scans for discovery.

In [4] a novel adaptive scheme for neighbor discovery in Bluetooth-enabled ad-hoc networks is introduced. In an ad-hoc peer-to-peer setting, neighbor search is a continuous, hence battery draining process. In order to save energy when the device is unlikely to encounter a neighbor, it adaptively chooses parameter settings depending on a mobility context to decrease the expected power consumption of Bluetooth-enabled devices. For this purpose, it first determines the mean discovery time and power consumption values for different Bluetooth parameter settings through a comprehensive exploration of the parameter space by means of simulation validated by experiments on real device. We then introduce two adaptive algorithms for dynamically adjusting the Bluetooth parameters based on past perceived activity in the ad-hoc network. Both adaptive schemes for selecting the discovery mode are based only on locally-available information. Simulations are carried out using a synthetic environment in which the human mobility is a variation of the Random way point mobility model.

In [9][10] the predictability in mobile interaction is quantified using the pre-existing social relations between users in a closed environment.. Mobile devices integrating wireless short-range communication technologies make possible new applications for spontaneous communication, interaction and collaboration. This work proposes to use collaboration to facilitate communication when mobile devices are not able to establish direct communication

paths. Opportunistic networks, formed when mobile devices communicate with each other while users are in close proximity, can help applications still exchange data in such cases. In opportunistic networks routes are built dynamically, as each mobile device acts according to the store-carry-and-forward paradigm. Thus, contacts between mobile devices are seen as opportunities to move data towards destination. In such networks data dissemination is done using forwarding and is usually based on a publish/subscribe. This work also presents the categories of a proposed taxonomy that capture the capabilities of data dissemination techniques used in such networks.

In [11] real-world measurements and simulations are used for deriving optimal parameters for symmetric ad hoc neighbor discovery using standard Bluetooth procedures. The inquiry procedure defined for Bluetooth is an asymmetric process: A device that wants to be discoverable enters the inquiry scan mode whereas the other device enters the inquiry mode in order to discover its neighbors Classical Bluetooth applications assume predefined roles for individual devices which at the same time leads to predefined neighbor discovery roles.

People often seek information by asking other people even when they have access to vast reservoirs of information such as the Internet and libraries. This is because people are great sources of unique information, especially that which is location-specific, community-specific and time-specific. Social networking is effective because this type of information is often not easily available anywhere else. In [12] a wireless virtual social network is conceived which mimics the way people seek information via social networking. PeopleNet is a simple, scalable and low-cost architecture for efficient information search in a distributed manner. It uses the infrastructure to propagate queries of a given type to users in specific geographic locations, called bazaars. Within each bazaar, the query is further propagated between neighboring nodes via peer-to-peer connectivity until it finds a matching query. The PeopleNet architecture can overlay easily on existing cellular infrastructure and entails minimal software installation. Two simple models are described, called the swap and spread models, for query propagation within

a bazaar. They propose a simple greedy algorithm which uses this limited information to decide which queries to swap. Their results demonstrate that PeopleNet, with its bazaar concept and peer-to-peer query propagation, can provide a simple and efficient mechanism for seeking information in opportunistic networks.

CHAPTER 3

K-Fold Predictor

This chapter discusses the goals, architecture and algorithms used with K-Fold Predictor for predicting an opportunistic contact in a purely opportunistic environment where no other networking infrastructure is utilized and the devices are smart phones carried by various human users.

3.1 Goals

The opportunistic contact prediction schemes in the previous chapter are a selected collection of methods that address the problem of contact prediction between two mobile devices. Their shared commonality is that none of the devices use their long term prior history of contacts but rely entirely on the local history i.e., their contact history in the recent past(of the order of a few seconds to few minutes) to decide contact prediction in the near future of the same order. All of these methods also maintain different modes of scanning the neighborhood, ranging from a most frequent mode to a least frequent mode. The most frequent mode scans the neighborhood for other devices the highest number of times in a unit time and vice versa for the least frequent mode. Also, none of these methods have any way of predicting the exact number of modes to be used.

Due to the above mentioned problems we propose K-Fold Predictor (KFP) for predicting opportunistic contacts. KFP has two primary goals:

- To switch to the data transfer mode, when the chance of encountering a device in the neighborhood is high.
- To maximize the number of opportunistic contacts, while spending minimal energy of the device.

The objective of KFP is to enable devices in an opportunistic networking environment to utilize their long term contact history to switch to the scanning mode (for other devices in the neighborhood) in anticipation of an opportunistic contact. From social theory[] , human users movements are not random ,each has a unique pattern and people tend to visit certain places only at certain times and thereby meet a certain group of other humans. KFP explores this regularity of movement to predict an opportunistic contact unlike other methods which do not take this into account.

Finally we also use KFP for information dissemination in the opportunistic network and perform a comparative study with other existing schemes.

3.2 Architecture

When a single device is operating under KFP it switches to the scanning mode for those time intervals which is in agreement with the prediction made by KFP. KFP involves a data collection stage in which an automated program runs in each users' device and maintains a log of all opportunistic contacts via Bluetooth. At the end of this data collection process, each user submits their contact logs to a central server, where KFP runs and returns its predicted slots to all the users.

This is an overhead of KFP in comparison with other methods, as it cannot start operating without the contact log being collected for a significant period of time.

In our work, we have used a dataset collected by others whose details are mentioned in Chapter 4. We assume that the user logs are already in the server and KFP accesses those logs to make the predictions. After collecting the data KFP makes the predictions and these predictions are communicated back to the devices. A schematic diagram of the process is shown in Fig 1 on page 10.

Finally we also use KFP for information dissemination in the opportunistic network and perform a comparative study with other existing schemes. The results show that KFP results in considerable energy savings as compared to the previous schemes.

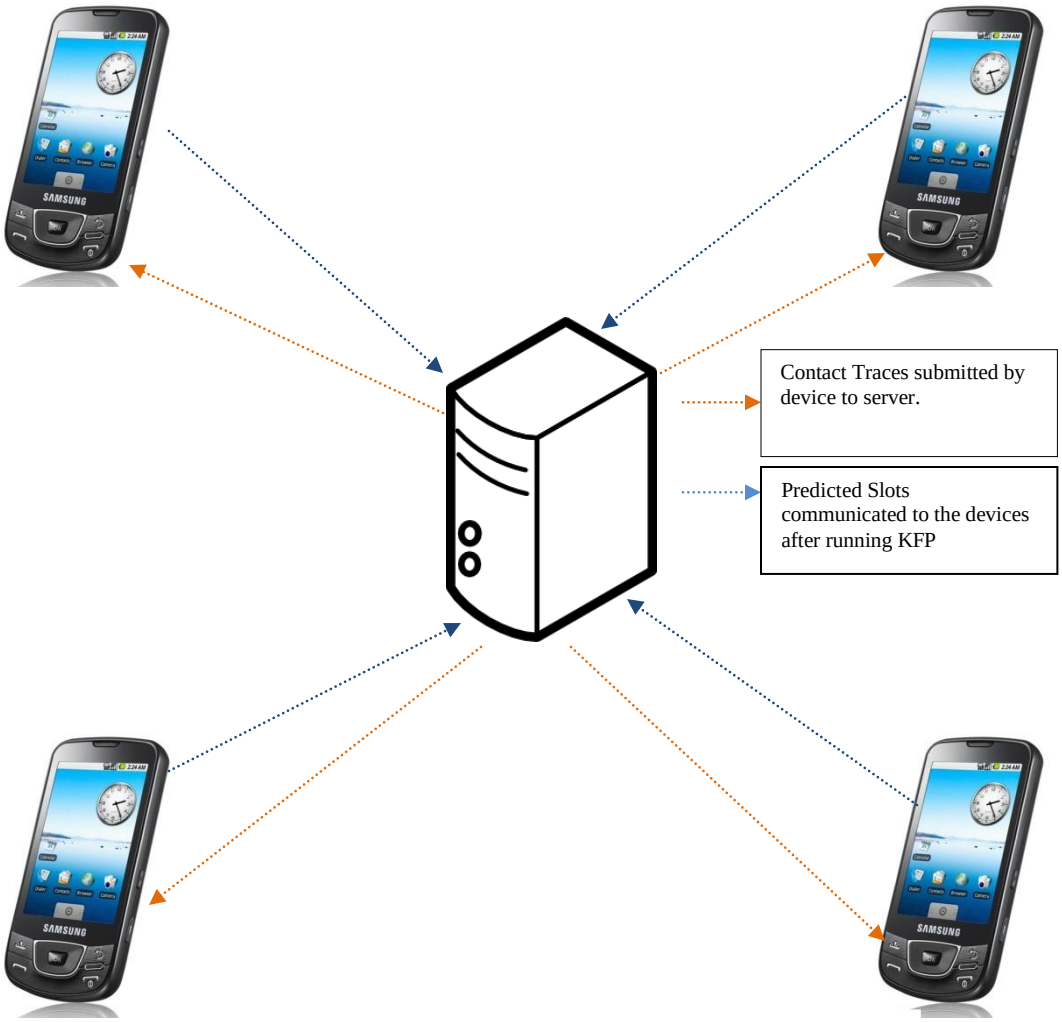

Fig 3.1 Architecture Of KFP

Also we do not deal with the problem of sending the predicted time slots back to the devices. In the next section we give a detailed overview of the server side working of KFP.

3.3 Temporal Dependence Ratio

In our work we use the time of an opportunistic contact to be repetitive in accordance with the regularity of a human mobility pattern. In order to validate this claim we define a metric called the Temporal Dependence Ratio which indicates as to which extent the Opportunistic

contacts of a single user is dependent on the time of contacts or they are more random. Our results indicate that the opportunistic contacts of any user is highly skewed. That is for most users the opportunistic contacts are concentrated in few one hour slots in the users for the entire day. This reaffirms our conviction that users have regular patterns in their daily lives and that the patterns tend to be repetitive.

Temporal Dependence Ratio is computed as follows:

1. The total number of distinct one hour slots monitored by a user during the data collection period (3 weeks) is acquired.
2. The number of opportunistic contacts for each such one hour slot is identified and sorted in descending order.
3. The percentage of opportunistic contacts for each slot for the user is computed.
4. Now for a given percentage(P) of the total number of opportunistic contacts the following is calculated:

Temporal Correlation Ratio(TCR) = (Minimum Number of one hour slots whose percentage of total contacts >= P)/(Total number of one hour slots)

TCR is calculated for each user for two values of P(70 and 80).

The TDR plot shows that there exists a high Temporal Dependence in the Opportunistic contacts for each user. The TCR plot shows that for a threshold of 80**% o**f OC ,only 46% of the one hour slots is sufficient on the average. (energy).For a threshold of 70 % ,only 35% of the one hour slots is sufficient on the average. For the purpose of clarity the plots are shown only for two values of threshold. For a threshold of 70 % ,only 35% of the one hour slots is sufficient

on the average. For the purpose of clarity the plots are shown only for two values of threshold.

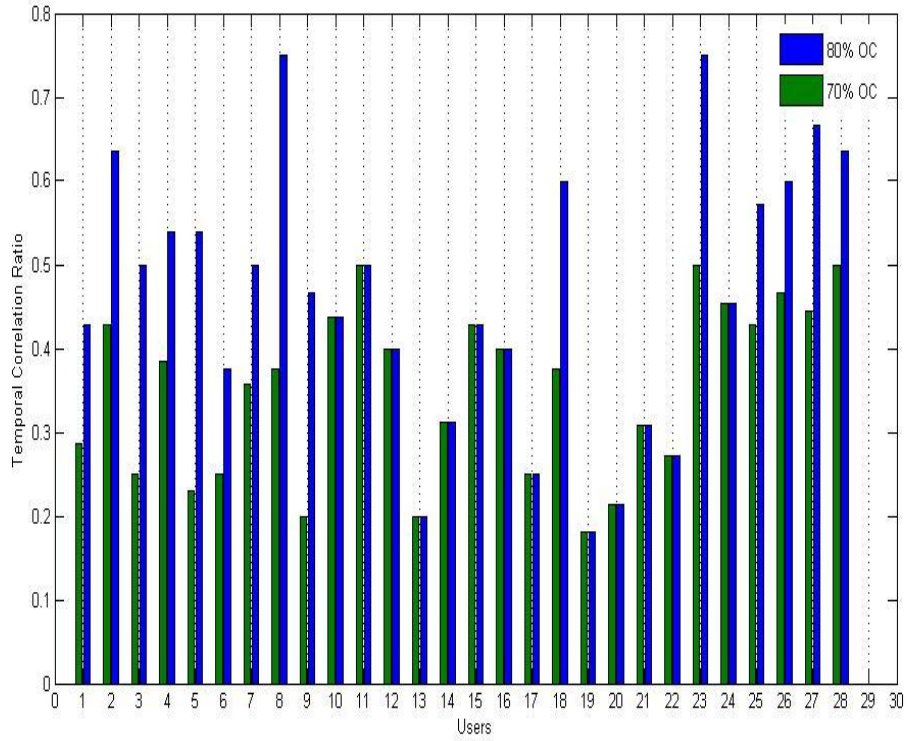

Fig 3.2 Temporal Dependence Ratio

3.4 Algorithms

The K-Fold Predictor can be divided into three distinct stages

- Data Processing
- Prediction
- Evaluation

The Output of each of these stages serves as the input of the next one.These stages are described in detail below:

3.4.1 Data Processing

In the data processing step the user logs are scanned to prepare a data structure which is essentially a list of Hash maps. Each Hash map corresponds to a unique user device that contains the details about their contact history. The key values of the Hash Map are the different one hour time slots and the value are a list of one dimensional arrays containing the count of opportunistic contacts for a distinct one hour time slot. The Hash Map is used as it is easy to check whether the contact history for one particular one hour time slot is already processed for one of the previous days. If not, then a new key value is added otherwise a new one dimensional array is added to the list of one dimensional array(s) corresponding to this key(one hour slot). A schematic diagram of this structure is shown in Figure 2. Each one dimensional array corresponds to a bucket whose size is determined by the scanning interval of the devices during the data collection period. The elements of the one dimensional array correspond to the contact history of the same one hour slot of the day over different days during the data collection period.

The number of such one dimensional arrays for a single one hour slot corresponds to the number of partitions that will be used in the prediction phase for cross-validation. The data processing step is the most time consuming process as for each user the list of Hash Maps is created after scanning through their respective contact history.

The scanning interval(α) is decided from the nature of the contact history. We assume that the scanning interval for each user during data collection period is constant for each user..

The psedo code for the data processing phase is presented below:

Input: BT Contact Patterns of users [U_1 , U_2,……., U_n],
scanning interval(α) in sec

Output: A List of hashmaps where each hashMap , SlotContacts<key,value> has the time slot(T) as the key and a List of one dimensional arrays Contacts[] containing the count of opportunistic contacts in T. Each HashMap corresponds to a unique user.

0. Declare a List of HashMaps (UserContacts).
1. **for** i = 1 to n
2. retrieve the list of BT traces for U_i.

```
3.      m = number of trace files for U_i.[f_1, f_2, f_3,….., f_m]
4.      for i = 1 to m
5.          extract Time Slot(T)  from the file f_i
6.          B = Duration(T)/ α          // This is the bucket size
8.          Declare a bucket(Contacts) of size B,( Contacts[B])  initialized to zero
9.          if  SlotContacts does not contain T then
10.             Declare SlotContactList.
11.         else
12.             SlotContactList = SlotContacts.get(T)
13.         for each line P in fi
14.             extract timestamp(ti(hh:mm:ss))  from P
15.             count =  count of opportunistic contacts(i.e device MAC Addreses)
16.             BNo = (mm * 60)/α           //Computing the correct bucket number
17.             if(ss > α) then
18.                Increment BNo by 1
19.             Contacts[BNo] = Contacts[BNo] + count
20.          end for
21.          Add contacts to SlotContactList
22.          Add <T, SlotContactList> to SlotContacts.
23.     end for
24.     Add SlotContacts to UserContacts.
25. end for
```

The heuristic used to decide the prediction of a slot is very simple. If the average number of contacts for all time intervals within a single one hour slot for K-1 number of days is less than the number of contacts for a single time interval within the same one hour slot for K-1 number of days then that single interval for the K-th day is predicted to be in the scan mode, otherwise it is predicted to be in the idle mode.

As the heusitic is simple its computation time is low. This heuristic is applied for each round of K-Fold Predictor. To avoid over fitting, only those one hour slots for an user is considered where K is greater than a pre specified constant. This constant is called the threshold partition or β. If the number of partitions are less than β, then no predictions are made for that particular one hour slot.

In this step all computations are performed on a user to user basis. During scanning the directory for a single user is scanned and the date and the one hour slot is computed from the filenames as deficed in the dataset. Also the scanning interval is extracted from the filenames which is thereby used to compute the number of buckets for a single one hour slot.

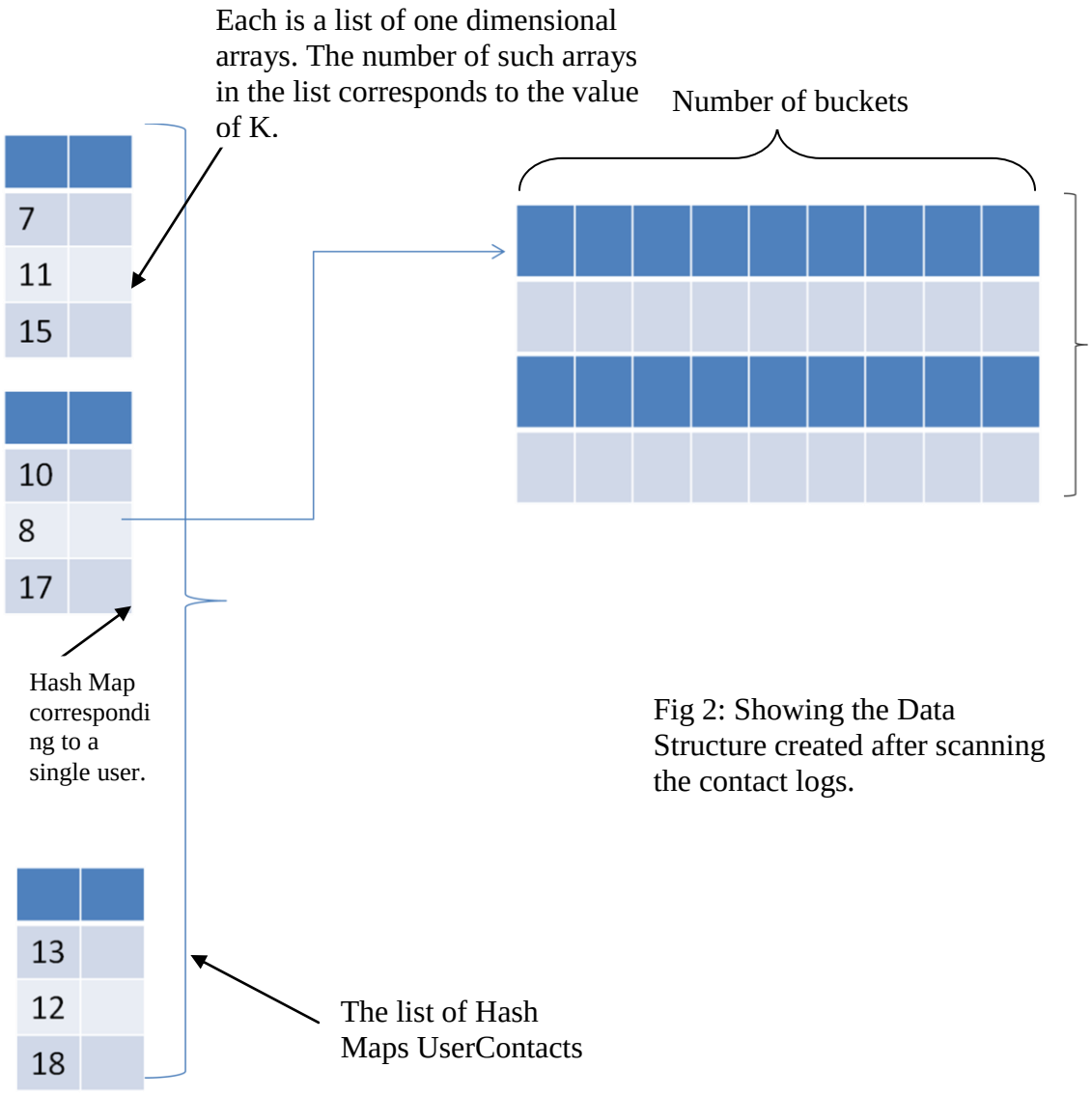

Fig 2: Showing the Data Structure created after scanning the contact logs.

Fig 3.3 Data structure after processing user logs

3.4.2 Prediction

In the prediction phase, we use a rotation based validation technique. One round of prediction involves partitioning a sample of data into complementary subsets, performing the analysis on one subset (called the training set), and validating the analysis on the other subset

(called the validation set or testing set). To reduce variability, multiple rounds of cross-validation are performed using different partitions, and the validation results are averaged over the rounds. This method is illustrated in the figure 3 given below:

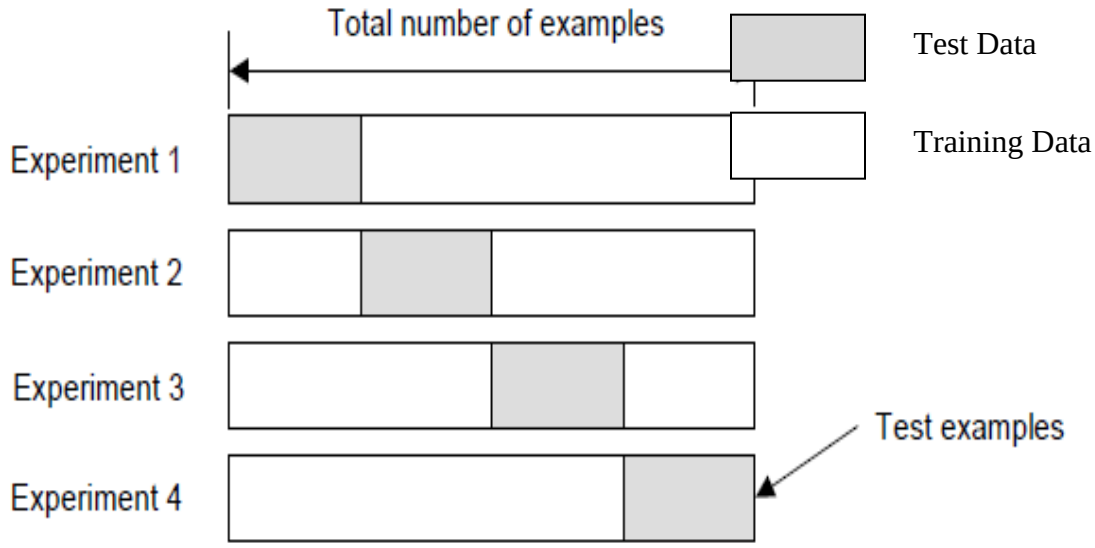

Fig 3.4 The Prediction Methodology

As shown in fig 3 above, each experiment corresponds to one round of cross validation. The total number of examples corresponds to the contact history which is divided into a number of equal size partitions. In the above figure the number of partitions, (K = 4).All but one of these partitions are used in the training data set to make the predictions and then the remaining partitions are used to validate the predictions. This process is repeated equal to the number of partitions created initially, and the accuracy of the method is averaged over the number of partitions created.

To avoid over fitting, only those one hour slots for each user is considered where K is greater than a pre specified constant. This constant is called the threshold partition or β. If the number of partitions are less than β, then no predictions are made for that particular one hour slot.

The pseudo code for the prediction phase which takes the output of the data processing phase as the input is given below:

Input: UserContacts: A list of HashMaps<T, SlotContactList>
Output: PredictedContacts: A list of HashMaps<T, PredictedSlotContactList> where PredictedSlotContactList is a single one dimensional array of boolean values denoting which interval needs to be switched on/off.

```
1. for i = 1 to n                    //n = length of UserContacts which is equal to no of users
2.        SlotContacts<T, SlotContactList> = UserContacts[i];
3.        Iterate over  all Keys of SlotContacts(Ti)
4.        SlotContactList = the list with key equal to (Ti)
5.        K = Length (SlotContactList) // Number  of OC traces for same time slot
6.        if (K < β)
7            No prediction is made for Ti
8         for p = 1 to K
9.           Choose P'th contact trace as the test data
10.          Choose all other slots(Q) as the training data.
11.          Declare a list for predictions. (PredictedContacts)
12.          Compute the mean of the number of contacts of all intervals of the one hour slots in
             the training data (Q), excluding all intervals for which there are no  contacts.
13.          If number of contacts in an interval in the training data > mean of the same interval
          in training data training data then
14.                Predict the slot as ON in PredictedContacts.
15.          else
16.                Predict the slot as OFF in PredictedContacts.
17           evaluateAcc (PredictedContacts, P)
18.       end for
19.       Compute the avg
19. end for
```

3.4.3 Evaluation

The prediction phase outputs a single one dimensional array of Booleans for each interval of a single user. The true value for one interval denotes that according to the prediction the device should switch to the more energy hungry mode of scanning devices In the neighborhood. This is computed on a per user basis for all one hour slots. For all those one hour slots which has been observed for enough number of days(K), the predictions are made, only those are considered for evaluation of performance.Now to determine the accuracy of KFP we adopt the approach shown in the diagram below:

Actual Opportunistic Contact (OC) for Intervals	Prediction of Opportunistic Contact (OC) for Intervals: +	Prediction of Opportunistic Contact (OC) for Intervals: -
+	1. Probing for OC when there is actually OC (tp).	2. Not Probing for OC when there are actual OC (fn).
-	3. Probing for OC when there is no actual OC (fp).	4. Not Probing for OC when there are no actual OC (tn).

Figure 3.5: Predict Contact Matrix

We classify the prediction of KFP into four different classes:

1. When KFP predicts for a device that a single interval should be in the scan mode and from the actual data of the contact history, it is found that there are indeed other devices in the neighborhood. This is defined as a true positive (tp).

2. When KFP predicts for a device that a single interval should not be in the scan mode and from the actual data of the contact history, but devices are found to be in the neighborhood. This is defined as a false negative (fn).

3. When KFP predicts for a device that a single interval should be in the scan mode but from the actual data of the contact history, it is found that there are no devices in the neighborhood. This is defined as a false positive (fp).

4. When KFP predicts for a device that a single interval should not be in the scan mode and from the actual data of the contact history, it is found that there are no other devices in the neighborhood. This is defined as a true negative (tn).

Classifying the predictions into these four categories enables us to analyze as to which of these types are favorable and which of them are not. Clearly, true positives (tp) and true

negatives (tn) are more favorable as they imply that the prediction is in agreement with the test data. The false positives (fp) and false negatives (fn) correspond to the error of the predictor.

As our final objective is to maximize the number of opportunistic contacts while saving as much energy as possible (as the scan mode is significantly more energy hungry compared to the idle mode) ideally both fp and fn should be low.

The evaluation phase takes the real contact history of a one hour slot(a one dimensional array containing the exact counts of OC per each interval) and a list of predicted intervals(i.e PredictedContacts) .For each interval it checks the number of OC from the real data ,for intervals which are predicted to be in the ON and OFF state by KFP. Depending on whether there are actual OC the algorithm correctly classifies them into one of the four categories described above. The pseudo code is shown below:

evaluateAcc (PredictedContacts, P)

Input: PredictedContacts: A list of HashMaps<T, PredictedSlotContactList> where PredictedSlotContactList is a single one dimensional array of boolean values denoting which interval needs to be switched on/off for a single one hour slot.

P: The real contact history of the same one hour slot for one particular day out of K days for the same slot.

Output: The tp, tn, fp, fn counts for the one hour slot and the precision and recall values.

```
1.    tp = fp = tn = fn = 0                    //Initialization
2.    contactCount = 0
3.    for all slots S in PredictedContacts
4.        if (no_of_contacts[S] > 0) then
5.            if (PredictedContacts[S] is ON) then
6.                tp = tp + 1
7.                contactCount = contactCount + no_of_contacts[S]
8.            else
9.                fn = fn + 1
10.       if (no_of_contacts[S] = 0)
11.           if (PredictedContacts[S] is ON) then
12.               fp = fp + 1
13.           else
14.               tn = tn + 1
15.   end for
16.   Precision = tp/(tp+fp)
17.   Recall = tp/(tp+fn)
```

For measuring the performance of KFP we employ two widely used metrics used in applications where successful detection of one of the class is considered more important than detection of the other class, namely *Precision* and *Recall*. A formal definition of these metrics is given below:

$$\text{Precision} = \frac{\text{tp}}{tp+fp}$$

$$\text{. Recall} = \frac{tp}{tp+fn}$$

Precision determines the fraction of records that actually turns out to be positive in the group the classifier has declared as a positive class .The higher the precision is, the lower the number of false positive errors committed by the classifier (i.e KFP).

Recall determines the fraction of positive examples correctly predicted by the classifier. Classifiers with higher recall values have very few positive examples misclassified as the negative class. A high value of recall also implies a low value of false negative errors committed by the classifier.

The Precision and Recall values are computed for a one hour slot once for each of the K-Folds and then it is averaged over K-Folds to get the overall Precision and recall values for a single one hour slot for a user. This process is repeated for all one hour slots for which prediction is made and ultimately the overall Precision and Recall value for a single user is computed by taking the average values of Precision ,Recall values for all such one hour slots.

The precision and recall values indicate the number of false positives and false negatives which in turn play a decisive role in formulating the energy model as described in the next section.

3.4.4 Energy Model

To model the energy consumption of KFP as compared to other schemes we adapt the methodology of Cost Sensitive Learning.

Let C(*i,j*) denote the cost of predicting a record from class *i* to class *j.*

So,C(+,-) denotes the cost of committing a false negative error, while C(-,+) denotes the cost of committing a false positive error.

Given a collection of N test records, the overall cost of a model M is,

$$C_t(M) = Tp * C(+,+) + FP * C(-,+) + FN * C(+,-) + TN * C(-,-).$$

Under the 0/1 cost matrix, i.e., C(+,+) = C(-,-) = 0 and C(+,-) = C(-,+) = 1, it can be shown that the overall cost is equivalent to the number of misclassification errors as shown below:

$C_t(M) = 0 * (TP+TN) + 1 * (FP+FN) = N * Err$, where Err is the misclassification error.

However, in modeling energy consumption, the binary 0/1 cost matrix does not hold good. First of all, as the energy consumed in idle mode is constant at all times, irrespective of the fact that the user is establishing opportunistic contact or not. So, in formulating the energy cost function, we consider only those intervals when the device is in the scan mode.

Also in our scenario, C(+,+) , C(-,+) i.e., the cost of a true positive(tp) and the cost of a false positive(fp) are the same as in both these cases energy is consumed as per the scan mode.

Again, C(+,-) = C(-,-) = 0 as no energy is consumed.

So our Overall energy metric(M) becomes:

$$M = (C(+,+) + C(-,+)) * P,$$

Where P is the energy cost associated with the mode of establishing an opportunistic contact.

CHAPTER 4

SIMULATION AND RESULTS

This chapter discusses the simulations conducted to demonstrate the viability of KFP. The data traces, scenarios, plots and results are provided below:

4.1 Data Trace

For simulation we have used the Bluetooth traces collected at the University of Illinois at Urbana Champaign, in their movement (UIM) framework. The dataset was generated using Google Android phones for a period of 3 weeks. 28 devices were given out to members of the faculty, staff and students of the Computer Science Department. Since most users use their phones as the daily phones, they collect the Bluetooth traces of the contacts these people encounter in places they visits (including home, school, super market, etc.).For the purpose of our simulation we do not assume any other networking infrastructure available other than these traces. For each user there is a directory 'btlog' which has all the contact history of the user. Some users are willing to charge their experiment phones a couple of times per day, then they accept higher scaning frequency. Others may not, then the frequency is lowered for them. The date and the time of data collection is encoded in the filename itself. Inside the log, a timestamp (scanning timestamp) and the list of scanned Wifi MACs, which are hashed for security reasons.

The number of modes for simulation of the Adaptive algorithm is decided after running the simulation for different values. With number of modes as 5 Adaptive Algorithm performs best, so it is considered as the benchmark for comparing against KFP.

At first we compute the number of contacts that a user encounters if he/she follows either one of the strategies. This is an important statistic because if a scheme predicts opportunistic contacts very accurately but as a side effect misses a significant portion of those contacts then the scheme will not be practically applicable.

The number of contacts each time KFP predicts that there will be a contact is computed in the prediction phase of KFP by actually checking against the test data. Similarly for the adaptive scheme, a count of successful contacts is kept. The results are plotted in the graph below:

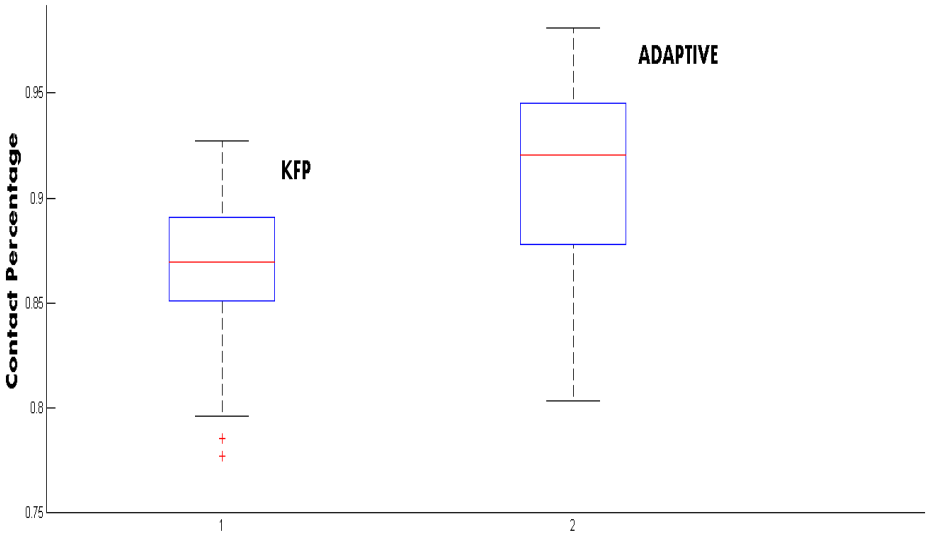

Fig 4.1 Contact Percentage for Adaptive and KFP Scheme

From the box plot we see that the number of encounters is on the average is 7% lower for KFP as compared to the Adaptive scheme. This is chiefly due to the fact that even if the likelihood of encountering an opportunistic contact is very low, the adaptive scheme still continues to scan the neighborhood for possible contacts at the least frequent scanning interval. As a result of which the adaptive scheme is able to encounter some contacts which fall out of the normal schedule of the user. As KFP exploits the regularity of human mobility, it is not able to account for these.

4.2 Precision for KFP vs Adaptive Scheme

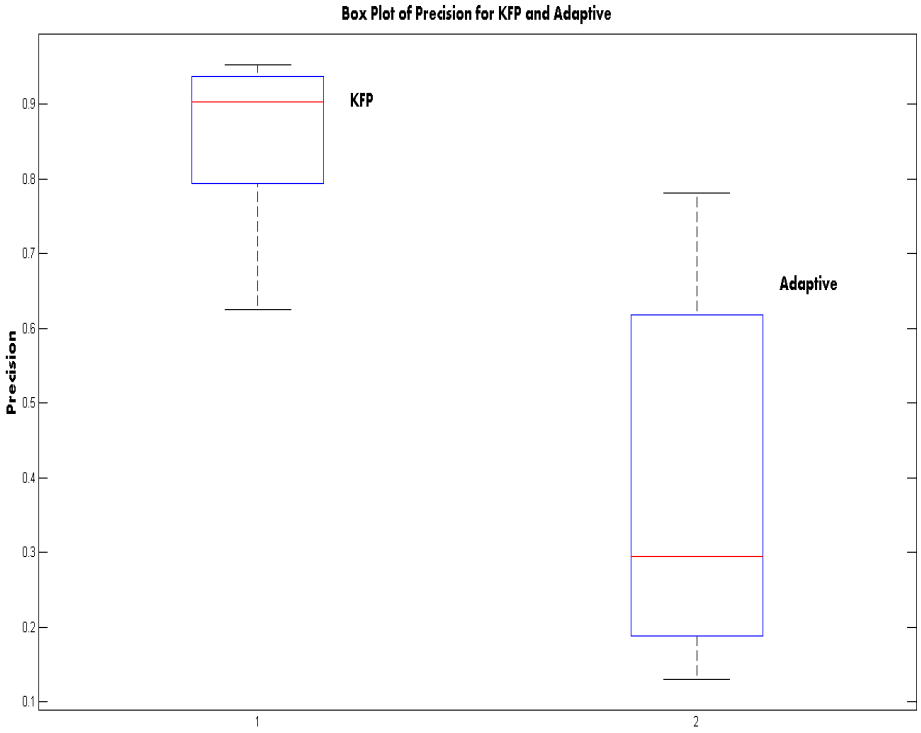

Fig 4.2 Precision for Adaptive and KFP Scheme

Precision determines the fraction of records that actually turns out to be positive in the group the classifier has declared as a positive class .The higher the precision is, the lower the number of false positive errors committed by the classifier (i.e KFP).

Precision is defined as the ratio of True Positives divided by the sum total of True Positives + False Positives.

The Precision values are plotted for both the KFP and the adaptive algorithm which dynamically chooses between 5 different scanning intervals. The number of modes for Adaptive is decided after running the simulation for different values and 5 gives the best performance for the Adaptive Algorithm, and so it is considered as the benchmark for comparing against KFP.

Table 4.1 Precision Statistics

	Min	Max	Mean	Standard Deviation	Median
KFold	0.625	0.964	0.853	0.100	0.902
Adaptive	0.130	0.781	0.398	0.226	0.294

For the Adaptive scheme , at the beginning of each one hour slot the most frequent scanning Interval is chosen. As with KFP the value for True Positive, False Positives , True Negatives are computed and the precision is computed. The precision values for KFP is 46% Better than the Adaptive Scheme thereby confirming that its False positives are much lower than that of the Adaptive Scheme. This indicates that when KFP predicts opportunistic contacts it has a much higher success rate as compared to its adaptive counterpart. This in turn indicates that KFP is more energy aware as its predictions for a device to be in the more energy consuming mode is much more accurate.

4.3 Recall for KFP vs Adaptive Scheme

Recall determines the fraction of positive examples correctly predicted by the classifier. Classifiers with higher recall values have very few positive examples misclassified as the negative class. A high value of recall also implies a low value of false negative errors committed by the classifier.

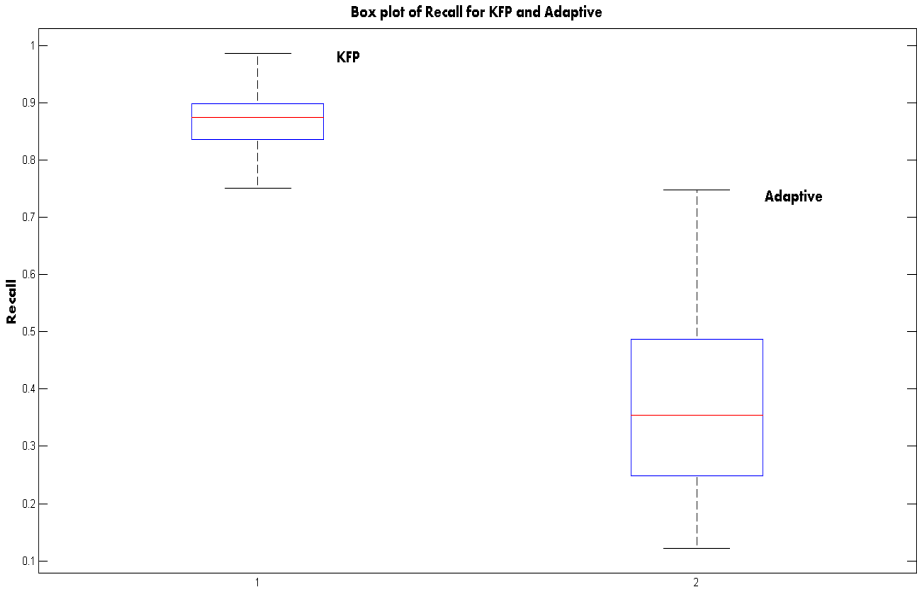

Fig 4.3 Recall for Adaptive and KFP Scheme

We compare the statistics for Recall for both KFP and Adaptive Scheme. The table below contains the summary of the results.

Table 4.2 Recall Statistics

	Min	Max	Mean	Standard Deviation	Median
KFold	0.750	0.987	0.873	0.057	0.883
Adaptive	0.122	0.747	0.378	0.162	0.376

We observe that the recall values are 50% better for KFP as compared to the Adaptive Scheme which in turn indicates that KFP is much more accurate than the adaptive scheme in predicting the non existence of opportunistic contacts in the neighborhood.

In order to model the energy cost of KFP in comparison to the Adaptive scheme at first the energy cost of Bluetooth is examined. Now the Bluetooth Energy Costs for a typical smart phone as per[13]are shown in the table 4.3 on page 27:

Table 4.3 Power consumption of Scan mode vs Idle mode

	Power Consumption in mW
idle	0.01
Scan	0.12

Power consumption in Scan mode is 12 times that of the idle mode.

Thus our energy cost function(M) becomes (C(+,+) + C(-,+)) * 12.

The Energy Metric is computed for each user for both the Adaptive Scheme and KFP.(M_{KFP} and $M_{adaptive}$ respectively).For each user it is found that $M_{adaptive} > M_{KFP..}$ In order to perform a comparative analysis of the adaptive scheme we define Energy gain(e) to be:

$$e = [(M_{adaptive} – M_{KFP}) / M_{KFP}] * 100$$

The energy gain(e) represents the percentage of energy saved with respect to the energy usage of the Adaptive scheme. We plot the value of e for all users in figure 4.4.

The statistics for Energy gain(e) is tabulated below:

Table 4.4 Statistics for Energy Gain

Min	Max	Mean	Standard Deviation	Median
21.37	58.74	34.57	10.88	32.43

The result clearly shows that KFP achieves significant energy gain at the cost of slight decrease in contact percentage. The minimum energy gain that KFP achieves is 21.37%. Hence by sacrificing a slight decrease in contact percentage KFP is able to achieve significant energy gain at the same time improving the accuracy of the predictions. Thus it is suitable for using in any opportunistic network application that wants to encounter future opportunistic contacts in an energy aware fashion. The suitable environment for deploying KFP is that of an office or a campus environment where the human agents have predictable movement pattern, especially when their movement has a high temporal correlation.

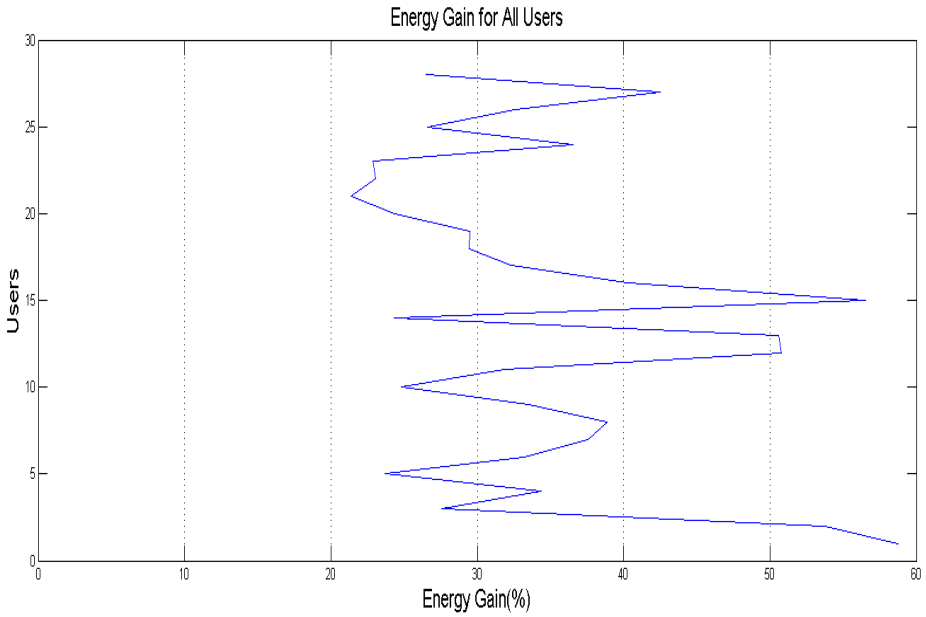

Fig 4.4 Energy gain(e) for KFP Scheme relative to the Adaptive Scheme

In comparison the adaptive scheme is able to account for slightly more number of opportunistic contacts at the cost of much higher consumption of energy.

CHAPTER 5

CONCLUSION AND DISCUSSIONS

This chapter provides an example application for use with KFP, summarizes the contents of this thesis and presents concluding remarks:

5.1 Example Application

This section discusses an application that could make use of the features within KFP. In this example, in a campus community notices are posted by the means of documents posted in notice boards. In a typical, campus community it is not always possible to go to these notice boards physically . Also many of the time, the information posted in such notice boards is not of interest to a particular person. As a result, on many occasions the members of the community loose the interest of following up with the information posted in such notice boards.KFP can be used both by the advertisers and the recipients to distribute and receive such information in the network, through their smart phones. Ideally the advertiser and the recipient would always like to keep their devices in the data transfer/scan mode, so that they can maximize their chances of receiving the information. However the energy cost for such an approach would be very high as the energy usage for a device in the transfer/scan mode is high as compared to the idle state. At the same time, persons in such an environment always have extremely regular schedules with repetitive patterns. As a result if they can switch to the transfer/scan mode based on their past history of meeting people in the same campus environment then effectively with a much less cost in energy they are maximizing their chances of receiving/distributing such information through opportunistic contacts.

5.2 KFP benefits and weaknesses

KFP is able to take advantage of the regularity of human movement in an environment(like campus/office) to maximize the number of opportunistic contacts for a person

at the same time conserving energy. Through a purely data driven methodology, KFP does not assume any other knowledge about the environment .As a result it can be applicable to many scenarios with no extra overhead or infrastructure.

On the other hand, the data collection process should be free of errors(like time synchronization between different devices should be adjusted).Some of the users may not want to participate in the data collection process for privacy concerns. Also the data needs to be refreshed from time to time, so as to reflect the current pattern in people movement.

5.3 Adaptive scheme benefits and weaknesses

The adaptive scheme requires no prior contact history of the users and operate purely in the adhoc mode depending on the recent contact history(of the order of few minutes).While the adaptive scheme is able to account for slightly more number of opportunistic contacts, it does so at the considerable expense of energy. In the absence of any prior contact history, the adaptive scheme continues to scan the neighborhood (with the least frequent scanning mode) even when for a significant time window there are no opportunistic contacts thereby increasing the energy cost.

5.4 Future Work

As part of our future work, we propose to validate with additional data sets in similar environments and also introduce the notion of contact duration in the predictions made by the KFP. In the contact duration aware KFP, the predictions accuracy will also depend on the specified contact duration(as per application context).Also we can study, the effect of information distribution using KFP and measure the contents availability in the opportunistic network with respect to other distribution schemes like flooding.

5.5 Conclusion

Overall KFP achieves a significant energy gain as compared to the adaptive scheme at the cost of a slight decrease in the number of contacts. This suggests that it can be a viable

alternative to the adaptive scheme when opportunistic contacts need to be established in an energy aware fashion.

REFERENCES

[1] Marta C. González, César A. Hidalgo, and Albert-László Barabási, "Understanding individual human mobility patterns," *Nature* **453**, 779-782 (5 June 2008).

[2] Schlich, R. & Axhausen, K. W. Habitual travel behavior, "Evidence from a six-week travel diary.Transportation" 30, 13-36 (2003)

[3] Wei Wang, Mehul Motani and Vikram Srinivasan Opportunistic Energy-Efficient Contact Probing in Delay-Tolerant Applications, IEEE/ACM TRANSACTIONS ON NETWORKING, VOL. 17, NO. 5, OCTOBER 2009

[4] Catalin Drula, Cristiana Amza, Franck Rousseau, and Andrzej Duda, Adaptive Energy Conserving Algorithms for Neighbor Discovery in Opportunistic Bluetooth Networks, IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, VOL. 25, NO. 1, JANUARY 2007

[5] Jing Su, Alvin Chin, Anna Popivanova, Ashvin Goel, and Eyal de Lara,"User Mobility for Opportunistic Ad-Hoc Networking," in 6th IEEE Workshop (WMCSA),2004.

[6] 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. UCAM-CL-TR-617, University of Cambridge, Computer Lab, 2005.

[7] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott, "Impact of human mobility on the design of opportunistic forwarding algorithms," in Proc. IEEE INFOCOM, 2006

[8] C. Boldrini, M. Conti, J. Jacopini, and A. Passarella,"HiBOp: a History Based Routing Protocol for Opportunistic Networks,WoWMoM 2007.

[9] R. C. Marin and C. Dobre. Exploring predictability in mobile interaction. May 2012.

[10] R. Ciobanu and C. Dobre. Data dissemination in opportunistic networks. In 18th Int. Conf. on Control Systems and Computer Science, CSCS

[11] D. Bohman, M. Frank, P. Martini, and Scholz C., "Performance of Symmetric Neighbor Discovery in Bluetooth Ad Hoc Networks," in GI Jahrestagung, 2004, vol. 1, pp. 138–142.

[12] M. Motani, V. Srinivasan, and P. Nuggenhalli, "Peoplenet: Engineering a wireless virtual social network," in Proc. ACM MobiCom, 2005, pp. 243–257.

[13] Ganesh Ananthanarayanan, Ion Stoica "Blue-Fi: Enhancing Wi-Fi Performance using Bluetooth Signals",MobiSys'09.

[14] Ganesh Ananthanarayanan, Ion Stoica "Blue-Fi: Enhancing Wi-Fi Performance using Bluetooth Signals",MobiSys'09.

BIOGRAPHICAL INFORMATION

Sujoy Kumar Bhattacharya was born in India in 1984. He took an early interest into computing when at the age of 10 he was introduced to LOGO and GWBASIC. He went on to pursue his Bachelors in Computer Science from Bengal Engineering and Science University in India and graduated with first class in 2007.

Sujoy has worked in Tata Consultancy Services(TCS) for two and a half years as a software developer working on Java/J2EE based web development frameworks. In 2010 he came to US to pursue graduate studies at the University of Texas at Arlington and started working under the supervision of Dr. Mohan Kumar. He plans to begin work in Amazon inc from January 2013 as a Software Developer.