Blue-Fi: Enhancing Wi-Fi Performance using Bluetooth Signals

Ganesh Ananthanarayanan
UC Berkeley
Berkeley, California
ganesha@cs.berkeley.edu

Ion Stoica
UC Berkeley
Berkeley, California
istoica@cs.berkeley.edu

ABSTRACT

Mobile devices are increasingly equipped with multiple network interfaces with complementary characteristics. In particular, the Wi-Fi interface has high throughput and transfer power efficiency, but its idle power consumption is prohibitive. In this paper we present, *Blue-Fi*, a sytem 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.

Our prediction results on traces collected from real users show an average coverage of 94% and an average accuracy of 84%, a 47% accuracy improvement over pure cell-tower based prediction, and a 57% coverage improvement over the pure bluetooth based prediction. For our workload, *Blue-Fi* is up to 62% more energy efficient, which results in increasing our mobile device's lifetime by more than a day.

Categories and Subject Descriptors

C.2.1 [**COMPUTER-COMMUNICATION NETWORKS**]: [Network Architecture and Design]

General Terms

Design, Algorithms, Measurement, Performance

Keywords

Bluetooth, Wi-Fi, location, context-awareness, energy-efficiency, mobile device

1. INTRODUCTION

Today's mobile devices, such as smartphones, are increasingly equipped with multiple network interfaces, including Wi-Fi, bluetooth, and, of course, cellular interfaces. These interfaces have widely different, often complementary, characteristics in terms of throughput, range, and power [12, 21, 22] (see Table 1). Ideally, one would like to intelligently leverage the strengths of these interfaces to optimize the application performance and minimize the power consumption.

	T'put	Transfer (J/MB)	Idle (W)	Scan (W)	Range (m)
Cellular	few 100 kbps	100	**0***	**0***	**500**
Wi-Fi	**11-54 Mbps**	**5**	0.77	1.29	100
Bluetooth	700 kbps	0.1	**0.01**	**0.12**	**10**

(* - *The cellular interface is typically always on.*)

Table 1: Various interface characteristics

Among the typical network interfaces found in today's mobile devices, Wi-Fi provides arguably the best combination of throughout, range, and power efficiency for data transfers. On the downside, Wi-Fi is the least power efficient in idle state and incurs a high overhead when scanning for new networks (e.g., anecdotes with *iPhone* [2, 4]). Thus, ideally, one should use Wi-Fi whenever it is available, switch it off when it is not, and avoid scanning whenever possible. This is particularly useful for applications that use the network periodically like Microsoft Pocket Outlook [5]. It is easy to see that implementing such a strategy requires one to *efficiently detect the Wi-Fi availability without switching it on.*

In this paper, we address this challenge by presenting *Blue-Fi*, a simple scheme that predicts the availability of Wi-Fi by using bluetooth contact-patterns and cell-tower information. The main observation behind our scheme is that users tend to repeatedly encounter the same set of bluetooth devices and cell-towers. Examples of such scenarios include a bluetooth mouse or spouse's mobile device while at home, colleagues' devices or printer at work, shop-owner's mobile device at his regular coffee place and even fellow commuters.

Most previous work on context-aware applications has focused on using GPS and cell-tower information. While GPS is highly accurate, it is power hungry (e.g., [24] and [8]), and thus not a good fit for our problem where the focus is on reducing the power consumption. Furthermore, GPS cannot be used indoors, where Wi-Fi connectivity is arguably most likely to be available. Cell-tower information has been successfully used for inferring contexts [12, 10] but as we demonstrate in this paper, its inaccuracy reduces its applicability to Wi-Fi prediction.

Despite bluetooth's power efficiency, so far it has been largely ignored by the context-aware applications. There are several reasons behind this state of affairs. First, blue-

MobiSys'09, June 22–25, 2009, Kraków, Poland.

tooth has a much lower range compared to other ubiquitous network technologies (see Table 1). Second, the discovery process of bluetooth devices is time-consuming and could take as long as 10 seconds [14]. Finally, unlike cell-towers and access points that are stationary, bluetooth devices are primarily carried by users and hence mobile.

Blue-Fi leverages bluetooth's low range to its advantage to achieve high prediction accuracy of Wi-Fi network availability. However, the high accuracy comes at a price: coverage. The smaller the communication range is, the less likely for *Blue-Fi* to find a nearby bluetooth device. To increase the prediction coverage, *Blue-Fi* complements bluetooth with cell-tower information. To alleviate bluetooth's high discovery time, *Blue-Fi* implements periodic discovery and uses the latest discovered list of bluetooth devices. Finally, despite the fact that many bluetooth devices are mobile, our results indicate that there is enough repeatability in contact-patterns of bluetooth devices that can be leveraged to provide accurate context information, especially indoors.

In its basic form, *Blue-Fi* requires no distributed infrastructure, or running complex distributed protocols. Each mobile device periodically logs *locally* the bluetooth devices, cell-towers, and Wi-Fi access points in its proximity, and later uses this information to predict Wi-Fi connectivity. To speed up the learning process, we also propose variants of *Blue-Fi* which employ peer-to-peer protocols or centralized web services to share the logs.

We make the following contributions. First, we advocate the use of bluetooth contact patterns for context inference, effectively providing a low-powered location system. In doing so, we address the limitations of the bluetooth devices with respect to low range, high discovery time and mobility. Second, we leverage the complementary properties of bluetooth and cell-tower to improve the prediction of Wi-Fi availability. By combining bluetooth and cell tower based predictions we obtain an average coverage of 94% and an average accuracy of 84%, a 47% improvement in accuracy over the pure cell-tower based prediction and a 57% improvement in coverage over the pure bluetooth based prediction scheme. This translates to an energy efficiency of up to 62% and increase in our mobile device's lifetime by more than a day. Finally, we analyze the benefits of collaborative prediction including security and privacy concerns.

The rest of the paper is organized as follows. Section 2 describes the system's functioning including the prediction schemes. Section 3 deals with the problem of bluetooth discovery. Algorithms to infer special bluetooth devices - landmark devices and mobile accessories - are presented in Section 4. Collaborative prediction, along with the security and privacy implications, is addressed in Section 5. Evaluation of the different aspects of the system is done in Section 6. We present improvements to *Blue-Fi* - identifying browser-authenticated Wi-Fi networks, and multihop bluetooth discovery - in Section 7. We discuss an indoor monitoring system, and enhanced prediction models in Section 8. Section 9 presents related work and contrasts it with *Blue-Fi*, and we conclude in Section 10.

2. PREDICTING WI-FI AVAILABILITY

In a nutshell, *Blue-Fi* predicts the Wi-Fi network availability by leveraging existing cell-towers and bluetooth devices. Each mobile device periodically logs all the network signals in a log L, locally. The log entries are of the form (*Timestamp*, {*Bluetooth devices*}, {*Cell Towers*}, {*Wi-Fi networks*}). Bluetooth devices are identified by their MAC addresses, cell-towers are identified by the tower identifier and Wi-Fi access points by their SSID/BSSID. The mobile device then uses its log to predict Wi-Fi connectivity.

The key question we need to address is: *how accurately can a bluetooth device or a cell-tower predict the Wi-Fi availability?* We first discuss the reliability of bluetooth prediction. If all bluetooth devices were fixed, predicting Wi-Fi availability would be easy. Given a bluetooth device b, just check whether there is a log entry containing b and a Wi-Fi access point: if yes, then we predict the Wi-Fi availability. Unfortunately, in practice many bluetooth devices are mobile (*e.g.*, phones, notebooks), so they cannot straightaway be used to predict fixed Wi-Fi access points.

To account for the mobile bluetooth devices we consider the correlation between the observations of the bluetooth devices and Wi-Fi access points. Intuitively, we consider a bluetooth device b to be a reliable predictor for Wi-Fi connectivity, if most of the log entries in which b appears contain at least a Wi-Fi access point.

L	Log of all network signals over time
τ	Threshold for a bluetooth device or cell-tower to be a reliable predictor
$predict_{cell}$	Cell-towers that are reliable predictors
$predict_{BT}$	Bluetooth devices that are reliable predictors
S_{BT}	Set of all bluetooth devices in the log
W_c	Connectable Wi-Fi networks
ED_T	Threshold of euclidean distance below which a user is considered stationary

Table 2: List of notations.

More precisely, let S_{BT} be the set of bluetooth devices in log L and W_c be the set of Wi-Fi networks which provides connectivity to the device (we describe mechanisms to obtain W_c in Section 7.1). For each Bluetooth device, $BT_i \in S_{BT}$, we compute the *predictability* of BT_i as $n(L, BT_i, W_c)/n(L, BT_i)$, where $n(L, BT_i, W_c)$ is the number of entries in L when BT_i was present and at least one of the networks in W_c was present, and $n(L, BT_i)$ is the number of entries in L when BT_i was present. *Predictability* is a confidence measure of how much a device's presence indicates Wi-Fi connectivity. If *predictability* is greater than a threshold τ, we add BT_i to $predict_{BT}$. In Section 2.2, we present a simple algorithm to set this threshold.

While a single bluetooth discovery or Wi-Fi scan can potentially miss some bluetooth devices or access points respectively, we believe the errors due such misses are minimized or negligible when $predict_{BT}$ is calculated over a sufficiently large set of log entries over time.

The algorithm to obtain $predict_{cell}$ is exactly the same as the algorithm described above, with bluetooth devices replaced by cell-towers. $predict_{cell}$ contains the list of cell-towers at places where the device has Wi-Fi connectivity.

2.1 Prediction Schemes

In this section, we describe the Wi-Fi prediction schemes based on bluetooth and cell-tower signals. The Wi-Fi prediction schemes predict Wi-Fi availability using $predict_{BT}$ and $predict_{cell}$. We evaluate the prediction schemes on two metrics:

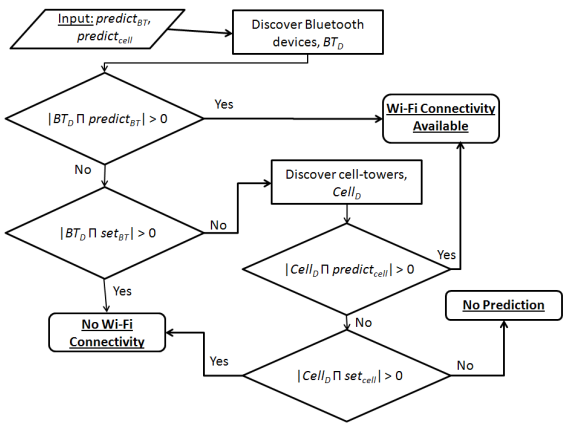

Figure 1: Hybrid Prediction Scheme

- **Coverage**: The fraction of existing Wi-Fi access points that are predicted.
- **Accuracy**: The fraction of the Wi-Fi predictions that are correct.

Bluetooth based Prediction: We predict that Wi-Fi is available if any of the bluetooth devices in $predict_{BT}$ is currently nearby. If none of the nearby devices are present in $predict_{BT}$, but there are some present in S_{BT}, then we predict that there is no Wi-Fi connectivity. If none of the nearby bluetooth devices have been seen before, the prediction scheme offers no prediction (we treat this as absence of Wi-Fi connectivity). Bluetooth based prediction is expected to exhibit high accuracy but low coverage as the range of bluetooth devices is much smaller than Wi-Fi (see Table 1). For example, a mobile device within Wi-Fi coverage may fail to predict Wi-Fi availability, if the bluetooth devices in $predict_{BT}$ are too far to be detected.

Cell-tower based Prediction: Cell-tower based prediction is based on the presence of currently visible cell-towers in $predict_{cell}$. As discussed in Section 1, the ranges of cellular signals is much higher than Wi-Fi networks resulting in coarse-grained predictions. As a result, cell-tower based prediction achieves high coverage but less accuracy.

Hybrid Prediction: This scheme uses a combination of bluetooth and cell-tower based predictions. More precisely, it uses bluetooth prediction first, and falls back to cell-tower based prediction when the former fails to predict. This way, hybrid prediction achieves both of best worlds by combining the accuracy of bluetooth prediction with the coverage of the cell-tower prediction. Figure 1 shows the flow diagram of the hybrid prediction scheme. The scheme starts with discovering the bluetooth devices that are currently nearby and checks whether any of them is present in either $predict_{BT}$ or S_{BT}. If yes, it uses bluetooth prediction. Otherwise, it resorts to cell-tower based prediction.

In the rest of the paper, we assume that if *Blue-Fi* does not make any prediction, then there is no Wi-Fi connectivity, *i.e.*, "No Prediction" is equivalent to "No Wi-Fi Connectivity" (see Figure 1).

2.2 Prediction Reliability Threshold (τ)

In this section, we describe an algorithm to select the appropriate prediction reliability threshold, τ, to populate

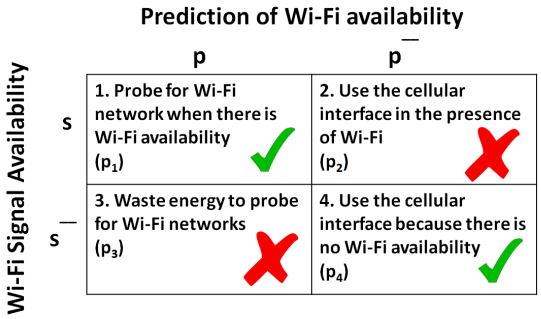

Figure 2: Predict-Signal Matrix

$predict_{cell}$ and $predict_{BT}$. High values of τ increase accuracy in prediction but lower coverage. On the other hand, low values of τ result in inaccurate prediction.

We determine the appropriate value of τ using a predict-signal matrix (see Figure 2). The matrix models the cases when our scheme predicts the availability or lack of Wi-Fi connectivity versus reality. The variables s and $\bar{s}$ indicate the presence and absence of Wi-Fi availability respectively, and p and $\bar{p}$ indicate cases when *Blue-Fi* predicts the availability of Wi-Fi. Cases 2 and 3 correspond to failures scenarious. In case 2, *Blue-Fi* predicts the absence of Wi-Fi connectivity even when it is present resulting in the device using the power-inefficient cellular interface. In case 3, the mobile device wastes power probing for Wi-Fi networks.

For a data transfer of size F, the expected wastage in energy is $E_{waste} = p_3 * E_p + p_2 * (F * e_c - (E_p + F * e_w))$ where p_2 and p_3 are the probabilities of cases 2 and 3, E_p is the energy consumed in probing for Wi-Fi networks, and e_c and e_w are the energy consumed per data unit for transferring data using the cellular and Wi-Fi interfaces respectively. E_p, e_c and e_w are constants. Automatically measuring and calibrating the values of E_p, e_c and e_w is orthogonal to *Blue-Fi's* objectives and has been addressed in prior work [17].

For a given transfer size F, the energy wastage E_{waste} is a function of p_2 and p_3 only. Next, we show that p_2 and p_3 can be expressed as functions of τ alone. Note that with the notations in the predict-signal matrix, *Accuracy* is $P(s \mid p)$ and *Coverage* is $P(p \mid s)$.

$$
\begin{aligned}
p_2 &= Pr(s \mid \bar{p}) \\
&= Pr(\bar{p} \mid s)\ (Pr(s)/Pr(\bar{p})) \\
&= (1 - Pr(p \mid s))\ (Pr(s)/(1 - Pr(p))) \\
&= (1 - Pr(s \mid p) * Pr(p)/Pr(s))\ (Pr(s)/(1 - Pr(p))) \\
&= (1 - Accuracy * Pr(p)/Pr(s))\ (Pr(s)/(1 - Pr(p))) \\
&= (Pr(s) - Accuracy * Pr(p))/(1 - Pr(p))
\end{aligned}
$$

$$
\begin{aligned}
p_3 &= Pr(\bar{s} \mid p) \\
&= 1 - Pr(s \mid p) \\
&= 1 - Accuracy
\end{aligned}
$$

$Pr(s)$ is the percentage of times when Wi-Fi networks from W_c was present in the log L and is a constant (not dependent on τ). Accuracy and $Pr(p)$ - the percentage of times when we found at least one bluetooth device from the set $predict_{BT}$ or one cell-tower from the set $predict_{cell}$ - are dependent only on τ. Therefore p_2 and p_3 are essentially functions of only τ. Note that p_2 and p_3 can also be expressed as functions of coverage instead of accuracy.

Hence, for a given transfer size, *Blue-Fi* finds the value of τ that minimizes E_{waste}.

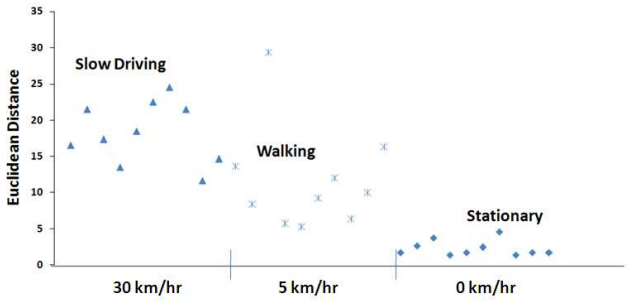

Figure 3: Variation in Euclidean Distance when a device is stationary, walking (5 km/hr) and driving (30 km/hr).

3. BLUETOOTH DISCOVERY

Bluetooth discovery takes considerable time (over 10 seconds [14]) causing high latency in prediction and subsequent application data transfers. We note that when the user is not moving at high speeds, Wi-Fi connectivities are likely to be relatively stable. Based on this observation, *Blue-Fi* scans for nearby bluetooth devices periodically, and stores and uses the latest discovered list for prediction. Periodic discovery reduces the latency of prediction using *Blue-Fi*. The period of discovery is appropriately set depending on the Wi-Fi scanning frequency and the ratio of power consumption of Wi-Fi scanning to Bluetooth discovery.

However, periodic discovery could be wasteful in times when device's Wi-Fi scanning frequency is low or the user is stationary. In such cases, Wi-Fi characteristics are not expected to change. For example, a user sitting in his office with his office's Wi-Fi connectivity is unlikely to see a change in his Wi-Fi prediction. In cases when the user is stationary, the last predicted Wi-Fi connectivity can be used without wasting power on updating the visible bluetooth devices.

Observations from prior work [25, 23] and our own experiments indicate that cell tower signal strength values have low variance when the device is stationary. A cell-tower fingerprint is a set of tuples containing the cell-tower identifier (CT) and signal strength (SS). Given two cell-tower fingerprints, $F_i = \{(CT_{i1}, SS_{i1}), ..., (CT_{in}, SS_{in})\}$ and $F_j = \{(CT_{j1}, SS_{j1}), ..., (CT_{jn}, SS_{jn})\}$, the euclidean distance is defined as $\sqrt{(SS_{i1} - SS_{j1})^2 + ... + (SS_{in} - SS_{jn})^2}$. Cell-towers that are present in only one of the fingerprints are not used in the calculation. High values of euclidean distance indicate low similarity between the fingerprints.

Figure 3 plots the average euclidean distance between consecutive readings in our experiments when a user is stationary, walking (5 km/hr) and driving slowly (30 km/hr). Consecutive measurements were taken a minute apart. As shown in Figure 3, the euclidean distance is a good indicator of whether a user is stationary or mobile.

Blue-Fi calculates the average euclidean distance since the last Wi-Fi prediction and if it is below a threshold, ED_T, periodic bluetooth discovery is not performed and the last made prediction of Wi-Fi availability is used. While we note that there are alternatives to the average euclidean distance like Spearman rank correlation coefficient [6] and common number of cell-towers, our metric is suited for our purpose and produces good results (see Section 6.4). Figure 4 illustrates the steps in *Blue-Fi's* periodic bluetooth discovery.

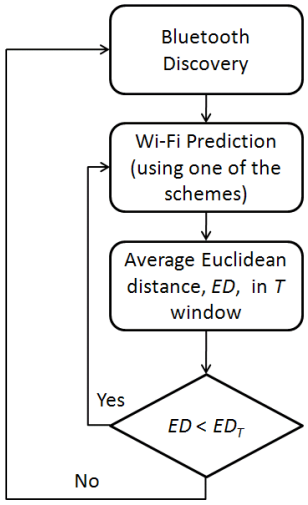

Figure 4: Periodic Bluetooth Discovery combined with Euclidean Distance Estimation

3.1 Euclidean Distance Threshold (ED_T)

The threshold of euclidean distance for inferring stationary periods, ED_T, depends on the fluctuation of cellular signal in a user's surroundings and hence fixing a static and uniform value across all users is problematic. We present a simple heuristic by which devices can calibrate this value themselves.

The heuristic is based on estimating the times when the device was stationary and calculating the average euclidean distance between the cell-tower fingerprints in those periods. We consider being in the proximity of a stationary bluetooth device (we discuss them in Section 4.2) or being connected to the same Wi-Fi access point (BSSID) as an indication of being stationary. Note that we are only interested in inferring times when a user's Wi-Fi connectivity is relatively stable as opposed to a user being completely stationary in the traditional sense. From the log, L, devices can calculate the average euclidean distance during stationary periods and fix the value of ED_T appropriately.

4. SPECIAL BLUETOOTH DEVICES

There are two classes of bluetooth devices that are especially interesting - *landmark devices* and *mobile accesories*. Landmark devices are devices that are stationary like a bluetooth mouse, keyboard or printer, and mobile accessories correspond to devices like bluetooth headsets. This section describes ways to identify these two categories of devices from the log, L. Landmark devices can be shared among users as they are not dependent on any single user's movements. Mobile accessories are not reliable indicators of Wi-Fi availability and should be removed from the logs.

Our identification technique is based on correlating the Wi-Fi access points and cell-towers, both of which are essentially stationary, with the bluetooth devices. In other words, we intend to capture the *variation* in the difference locations at which a bluetooth device was sighted. To this end, we first introduce the notion of *diversity* for a device and then use that to identify landmark devices and mobile accessories.

4.1 Diversity

Diversity captures the variance among the list of locations a device is sighted. For every bluetooth device in S_{BT}, we extract the list of Wi-Fi and cell-tower signatures that were co-sighted along with it. A Wi-Fi signature is a list of BSSIDs, and a cell signature is a list of cell-tower identifiers. Note that the list of signatures can contain repetitions. We use the list of signatures as an indicator of the locations the bluetooth device was present.

We define *similarity* between two signatures A and B as $|(A \cap B)| \ / \ |(A \cup B)|$. Clearly, the higher the number of intersecting wireless access points or cell-towers, the greater the value of *similarity*.

We use *similarity* to now calculate *diversity* for a list of signatures. Let $W = \{W_s\}$ be the list of Wi-Fi signatures for a bluetooth device. We identify the *median* of W using K-Medians clustering [18]. K-Medians is suited for our purpose as it can perform clustering even when the "points" to be clustered are not in Euclidean space, as in our case where we have to cluster signatures. K-Medians clustering exhaustively evaluates the suitability of each W_s in W to be the median and picks the best. Table 3 describes the algorithm to pick the median and calculate the diversity of a given list of signatures. The presence of repetitions in the signature list, W, ensures that the signatures are automatically weighted during the calculation of *diversity*. *Diversity* is calculated for every device and varies between 0 and 1. Diversity for a list of cell-tower signatures is exactly similar with Wi-Fi BSSIDs replaced by cell-tower identifiers.

```
Input: List of signatures, W = {W_s}
Output: Median, W_m and Diversity, D

Variables:
Aggregate Similarity, agg_sim
Highest Similarity, best_sim ← 0
Current Median, W_mc

Algorithm:
for each W_si in W
    W_mc ← W_si
    agg_sim ← 0
    for each W_sj in W
        agg_sim += Similarity(W_si, W_sj)
    end for
    if(agg_sim > best_sim)
        best_sim ← agg_sim
        W_m ← W_mc
end for

for each W_s in W
    D += Similarity(W_s, W_m)
end for

D ← D/|W|

return (D and W_m)

float Similarity (Signature W_1, Signature W_2)
    float similarity = |(W_1 ∩ W_2)| / |(W_1 ∪ W_2)|
    return similarity
```

Table 3: Diversity of a signature list

4.2 Landmark Devices

Intuitively, a landmark device is one that a user discovers only at one location (e.g., home) and always discovers that device when he is at that location. Precisely, a device BT is classified as landmark if it satisfies two properties: (1) Its $diversity$ is sufficiently low, and (2) Whenever a signature that is similar to its median, W_m, occurs in the log, BT should also be present. Property (1) ensures that BT is seen only at one location. Property (2) ensures that whenever the user is at that location, BT is seen.

4.3 Mobile Accessories

Mobile accessories are not reliable indicators of Wi-Fi availability and should be removed from the logs. We treat devices that occur in a high fraction of the log entries to be mobile personal accessories and unreliable indicators of a user's location.

5. COLLABORATIVE PREDICTION

This section explores the scenarios, benefits and issues if devices were to collaborate and share information about Wi-Fi availability. An example scenario when such sharing is beneficial is when a user goes to a new place with no prior context (improve coverage). Sharing essentially speeds-up the learning process. We present two sharing approaches − peer-to-peer and global − and also discuss their security and privacy implications.

5.1 Peer-to-peer Sharing

Devices query each other over bluetooth for the availability for Wi-Fi networks. Any device that is currently using the Wi-Fi network can respond with details including the bandwidth and authentication mechanisms. In fact, users can exchange more useful information like "*I performed a 1 MB download 30 seconds back and got a 10 Mbps throughput*". Such sharing can be used for collaborative selection of access points. Hence, presence of at least one bluetooth device can potentially result in a device knowing about Wi-Fi connectivity.

We can extend the hybrid prediction scheme to include peer-to-peer querying wherein the device queries its neighbors when bluetooth prediction is inconclusive and before resorting to cell-tower based prediction.

5.2 Global Sharing

The alternative to peer-to-peer sharing is global sharing facilitated through a central service. Devices, when using the Wi-Fi network, periodically upload entries − (*Timestamp*, {*Bluetooth devices*}, {*Cell Towers*}, {*Wi-Fi network,* {*Characteristics*}}) − to a centralized server. *Characteristics* specify if a network requires authentication along with performance parameters like throughput and latency. While information about security of a network (e.g., WEP encryption) is specified for all Wi-Fi beacons, performance parameters are provided only for the associated network. Servers index these entries by bluetooth devices and cell-tower identifiers for efficient retrieval.

Any device that wants to know about its current Wi-Fi availability can query the central server by supplying the list of currently visible bluetooth devices and cell-towers. The server responds by matching the bluetooth devices and cell-towers in its database and returns the Wi-Fi networks along with their characteristics. Communication with the server

happens over the cellular interface (data channel or simpler options like SMS). The details of matching application needs with the best suited Wi-Fi network are beyond the scope of this paper.

Matching cell-towers need not involve the timestamp field in the entries in the server because they are stationary. Bluetooth devices are mostly dynamic and should be matched only if the time of uploading of the log entry and time of querying is within a bounded interval. Note that the server can process the logs and obtain the set of landmark devices as described in Section 4. Matches with landmark devices need not consider the timestamp.

The ability to leverage landmark devices makes the global sharing option potentially more useful than peer-to-peer sharing; landmark devices are not expected to do peer-to-peer communication. But on the other hand, peer-to-peer sharing does not need any infrastructure support and hence is more readily deployable.

5.3 Privacy and Security

In this section, we discuss the security and privacy concerns related to global and peer-to-peer sharing. A trusted service model, like the emerging location-based services [1], alleviates these concerns for the case of global sharing. Next, we discuss the security and privacy issues in peer-to-peer sharing, and compare it with the global sharing model.

1. **Intrusion:** Peer-to-peer collaboration requires devices talking to other devices around them, and users might be wary of such communication as it might potentially lead to intrusions. Without any prior knowledge, there is no feasible filter that users can apply. Global sharing requires the device to just talk to a trusted server.

2. **Usage Pattern:** If devices were to exchange details about their network activity (size of download, time of download), a malicious person can continually query a device in the guise of knowing the Wi-Fi connectivity characteristics, and end up obtaining the exact access patterns of the device. This can be correlated to make decisions about a user's activity like streaming, browsing etc. potentially leading to annoying and targeted advertisements. With global sharing, the central server is expected to hide the identity of the users who uploaded the data.

3. **Industrial Espionage:** Assume a scenario where coffee shop A has Wi-Fi connectivity for its customers. A competitor, B, is looking to set up its store near A's and expects the same customers to visit B too. Now B can have a person with a device sitting in A's shop, querying all the users visiting A continuously and aggregating information about their Wi-Fi usages. B can use this information to appropriately provision its network, thereby obtaining information that A would not have provided otherwise. The central server in global sharing will provide only representative information in response to queries.

   While it is true that Wi-Fi frames themselves can be passively scanned to get the required information, peer-to-peer sharing provides yet another way for this information to be leaked.

6. EVALUATION

We collected logs from twelve volunteers for a period of two weeks each. In this section, we use the logs to evaluate our prediction schemes for accuracy and coverage, appropriate threshold selection, effects of periodic discovery and energy efficiency for our workload. We also evaluate our algorithms for identifying landmark devices and mobile accesories. Finally we quantify the benefits of collaborative prediction. We briefly summarize our results here.

1. Hybrid prediction scheme produces high coverage (93.5 %) and accuracy (84.2%), a 47% improvement in accuracy over the pure cell-tower based prediction and a 57% improvement in coverage over pure bluetooth based prediction scheme.

2. Periodic discovery results in negligible reduction in accuracy and coverage.

3. Energy consumption reduces by up to 62% for our workload using *Blue-Fi's* prediction techniques.

4. Collaborative prediction through sharing improves the coverage by up to 36.2%.

6.1 Log Collection

Twelve volunteers were given i-mate PDAs programmed to log the Wi-Fi, bluetooth and cell-tower signals as mentioned in Section 2, every minute. The i-mate runs Windows Mobile 5.0 and is equipped with a Class-2 bluetooth interface. We performed Wi-Fi scanning using a library that was built using the Windows Driver Development Kit. Bluetooth scanning happened using the open-source library, *InTheHand* [3]. GSM tower information was obtained by reading a well-known memory location that has been obtained by the community via reverse-engineering [19]. The volunteers were a mix of graduate students in Berkeley as well as working professionals in San Francisco Bay Area. Volunteers carried the PDA along for two weeks in their normal routine. No instance of the PDA's battery discharging or any such incident that stopped the logging was reported.

Table 4 lists the details about the log. There are two noteworthy features from the logs:

1. The fraction of times when there was Wi-Fi connectivity from the preferred networks varies from 32.1% to 68%, indicating it is not ubiquitous and hence the need for a mechanism for predicting Wi-Fi availability.

2. The number of bluetooth devices and the fraction of times they are visible (49.6% to 77.2%) are encouraging to base a prediction scheme on them.

One half of the data was used for training and the other half for testing.

6.2 Coverage and Accuracy

We measure the accuracy and coverage of using bluetooth and cell-tower data individually for prediction, highlight their complementary properties and demonstrate the benefits of using them in conjunction in the hybrid scheme.

Bluetooth based Prediction: Figures 5 and 6 show the accuracy and coverage (in %) for bluetooth-based prediction for the different users, for varying values of the prediction reliability threshold, τ. We observe that accuracy is directly

User	Duration (weeks)	Cell-Towers*	Wi-Fi SSIDs*		Bluetooth Devices*
			Preferred	All	
U1	2	121 (99%)	48.1%	490 (76.1%)	159 (50.1%)
U2	2	337 (99%)	56%	1156 (84.5%)	194 (71%)
U3	2	191 (98%)	32.1%	731 (68.1%)	181 (51.2%)
U4	2	287 (99%)	36%	846 (54.3%)	133 (66.4%)
U5	2	101 (99%)	68%	744 (81.2%)	99 (77.2%)
U6	2	202 (98%)	44.3%	553 (65.2%)	152 (49.6%)
U7	2	221 (99%)	55.8%	701 (77.4%)	201 (55.1%)
U8	2	188 (99%)	67.3%	553 (86.1%)	199 (51.8%)
U9	2	241 (98%)	58%	552 (71.1%)	288 (59.3%)
U10	2	302 (100%)	60.3%	774 (80.8%)	222 (63.7%)
U11	2	254 (99%)	53.2%	691 (63.7%)	241 (66.4%)
U12	2	198 (98%)	54.1%	801 (70.2%)	201 (62.2%)

(* – *Fraction of times when at least one signal was observed on that interface*)

Table 4: Details of logs collected by users

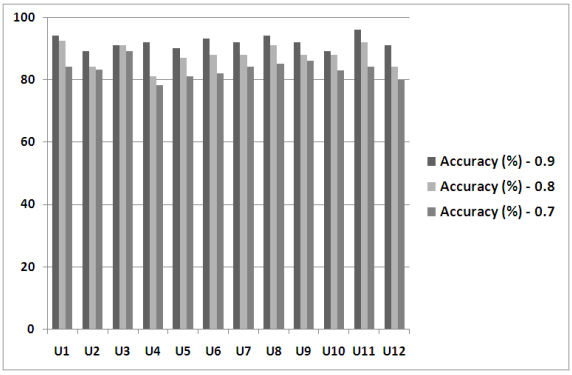

Figure 5: Accuracy of Bluetooth-based Prediction

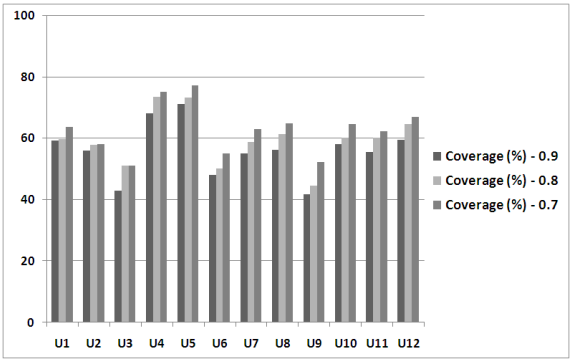

Figure 6: Coverage of Bluetooth-based Prediction

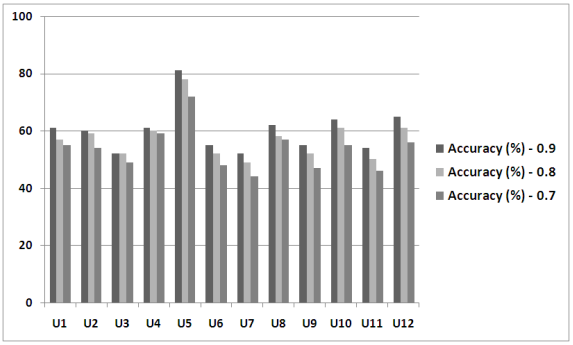

Figure 7: Accuracy of Cell-Tower based Prediction

proportional to τ while coverage is inversely proportional with τ. Due to the limited range of bluetooth, the average accuracy of the prediction is high (87.25%), but the average coverage is low (61%). High accuracy but low coverage leads to erroneous conclusions of lack of Wi-Fi availability even in their presence. U4 and U5 have coverage of near 80% indicating that if a user visits very few places routinely, bluetooth-based prediction can provide good coverage.

Cell-Tower based Prediction: Cell-tower based prediction has complementary properties to bluetooth based prediction. We see high average coverage of 93.5% but an average accuracy of only 59.66% (Figures 7 and 8). Note that U5 has a high accuracy of close to 80%. If a user's movements in a given area (when in range of a cell-tower) is limited, then cell-tower based prediction can be sufficient. Again, accuracy is directly proportional to τ while coverage is inversely proportional with τ. High coverage and low accuracy lead to wastage of energy in unnecessary scanning for Wi-Fi networks when there are none.

Hybrid Prediction: The Hybrid Scheme combines the advantage of both the earlier schemes and shows coverage of 93.5% and an accuracy of 84.2% for $\tau = 0.8$. Figure 9 plots the accuracy and coverage of the hybrid prediction scheme. This is a 47% improvement in accuracy over the pure cell-tower based prediction and a 57% improvement in coverage over pure bluetooth based prediction scheme. We believe this to be an encouraging validation for the usage of bluetooth contact-patterns and cell-tower information in tandem for inferring contexts.

6.3 Prediction Reliability Threshold (τ)

Picking the right threshold for a given file transfer influences the expected energy loss due to erroneous predictions. We measured how the energy loss, E_{waste}, varies with the prediction reliability threshold, τ, for different transfer sizes. We used the values of $e_c = 100$ J/MB, $e_w = 5$ J/MB and $E_p = 5$ J [12]. Figure 10 plots the results for U3. The optimal threshold for the 100 KB transfer is 0.6 while it is 0.7 for higher file transfers. Smaller transfer sizes are more sensi-

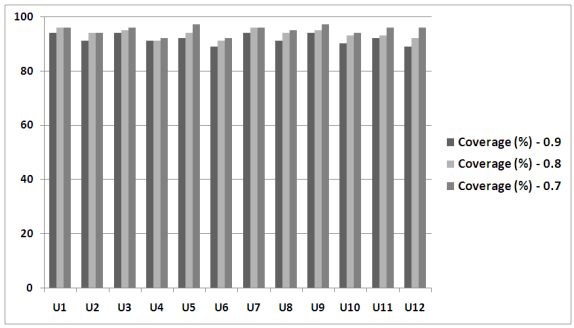

Figure 8: Coverage of Cell-Tower based Prediction

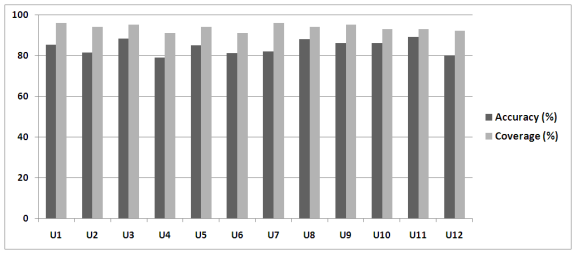

Figure 9: Hybrid Scheme combines the Accuracy and Coverage of Bluetooth and Cellular Contexts

tive to τ. Note the sharp variation in the value of E_{waste} on either side of $\tau = 0.6$ for the 100 KB download. We see a plateau after the value of τ exceeds 0.7 for transfer sizes of 1 MB and 10 MB.

Across other users, we note that the optimal value of τ varies between 0.5 and 0.7 for the 100 KB download, and between 0.7 and 0.8 for the 1 Mb and 10 MB transfer. Interestingly, for users U2 and U8, the optimal value of τ is 0.7 for all download sizes.

6.4 Periodic Discovery

Recall that *Blue-Fi* deals with bluetooth's high discovery time by periodic discovery and using the latest discovered list. It uses euclidean distance between cell tower fingerprints to avoid bluetooth scans if a device is stationary. Threshold for variation in euclidean distance, ED_T, was calculated from the readings in L when the user was continuously in the presence of the same Wi-Fi BSSID provided by one of his preferred networks, and landmark devices (we fixed a minimum of five readings). Table 5 lists the ED_T values calibrated by the different devices using W_p BSSIDs and landmark devices. True to intuition, we observe that ED_T values obtained using landmark bluetooth devices is smaller than those obtained using Wi-Fi BSSIDs as reference, as the range of bluetooth signals is significantly lower than Wi-Fi networks. Also, the diversity in values of ED_T across different users indicates that individual devices calibrating themselves is better than fixing a uniform value.

We evaluate change in accuracy and coverage of bluetooth based prediction when periodic discovery is performed for periods of 10 minutes, 15 minutes and 30 minutes. We see negligible reduction in accuracy and coverage because of using the last discovered list compared to on-demand discovery.(see Figures 11 and 12).

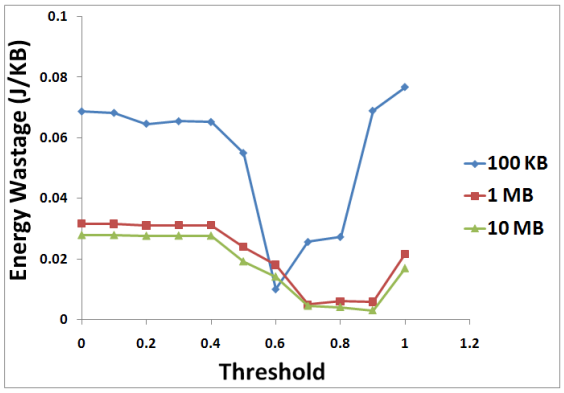

Figure 10: Expected Energy Wastage per KB for different file transfers

User	Preferred Networks	Landmark Devices
U1	4.1	1.2
U2	5.6	1.6
U3	2.5	2.2
U4	6.7	1.1
U5	6.6	1.9
U6	3.2	0.9
U7	2.1	2.2
U8	4.4	1.8
U9	3.1	2.7
U10	4.9	1.2
U11	4.1	2.2
U12	2.2	2.4

Table 5: Calibrated value of threshold for Euclidean Distance, ED_T, using BSSIDs of preferred Wi-Fi networks (W_p) and landmark bluetooth devices

6.5 Energy Consumption

We evaluated the energy consumption of our schemes and compared it to other schemes using a workload that models commonly used background applications like email synchronizers and RSS feed readers. Background applications use a "pull" model by periodically polling the server for new data and synchronizing the copy on the mobile device with the server. Our workload consists of periodic synchronization activities of 100 to 200 KB. Using the full battery capacity of the phone, we measure the number of such synchronizations performed before the phone runs out of power (total capacity of 16200 J).

We compare our hybrid prediction scheme with two commonly used modes of network usage in practice: (a) use the cellular interface always ($E_{cellular}$), and (b) scan and check for Wi-Fi availability always − use Wi-Fi if available, cellular connectivity otherwise (E_{Wi-Fi}).

For our twelve users, we report encouraging improvements of 19-62% compared to $E_{cellular}$ and 20-40% compared to E_{Wi-Fi}. In addition, we make the following observations:

- **Preferred Networks:** The overall fraction of times when a user has Wi-Fi connectivity also affects the gains compared to $E_{cellular}$ and E_{Wi-Fi}. The overall coverage of Wi-Fi networks vary from 32.1% to 68% of the time for our twelve users. Figure 14 plots the gains

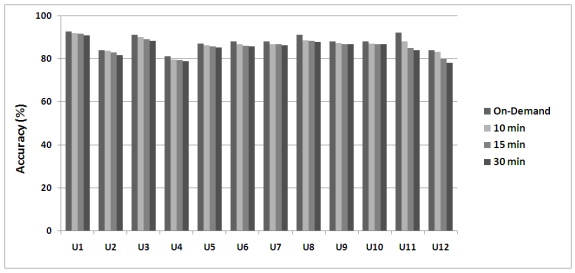

Figure 11: Reduction in Accuracy because of periodic Bluetooth discovery

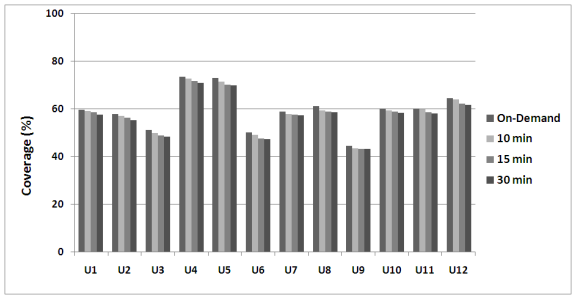

Figure 12: Reduction in Coverage because of periodic Bluetooth discovery

compared to $E_{cellular}$ and E_{Wi-Fi}. When the fraction of Wi-Fi coverage is low, the device ends up using the cellular interface most times and hence the gains compared to $E_{cellular}$ are limited. Low fraction of Wi-Fi coverage potentially leads to many wasteful scans and hence the utility of prediction is high (compared to E_{Wi-Fi}). For a high fraction of Wi-Fi coverage, the probability of a scan finding Wi-Fi connectivity is high and hence the value of prediction is limited.

- **Download Size:** The performance of Wi-Fi prediction schemes vary depending on the amount of data downloaded in every round. Figure 13 plots the gain for varying data sizes. The gains compared to $E_{cellular}$ reduces for small downloads. For small downloads, the energy difference between using the cellular interface and Wi-Fi (when available) is limited. The gains compared to E_{Wi-Fi} on the other hand, decreases for large downloads. Large downloads amortize the energy wastage due to incorrect Wi-Fi scanning (i.e., scanning when there is no Wi-Fi connectivity) and the advantage of using prediction schemes is limited.

- **Lifetime:** The percentage increase in the device's battery lifetime is the same as the percentage increase in the number of synchronizations; the actual increase in lifetime is a function of the synchronization frequency. For downloads of 100 KB, and synchronization frequencies of 1 minute and 5 minutes, we see that the lifetime of the device compared to $E_{cellular}$ increases by 1.05 to 5.23 hours, and 5.25 to 26.15 hours respectively for our users. The corresponding numbers compared to E_{Wi-Fi} are 5.51 to 7.78 hours, and 27.5 to 38.9 hours. We make the assumption that the mobile device is not expending power on any other activity.

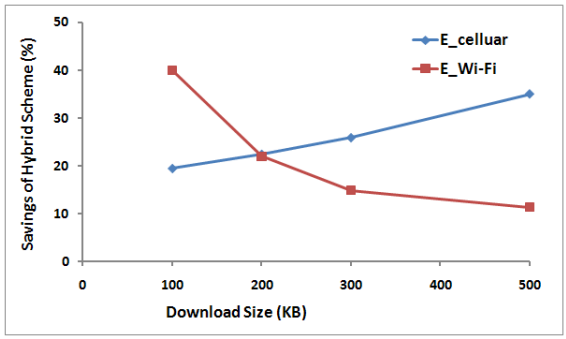

Figure 13: Gains compared to $E_{cellular}$ and E_{Wi-Fi} vary with download size for U3

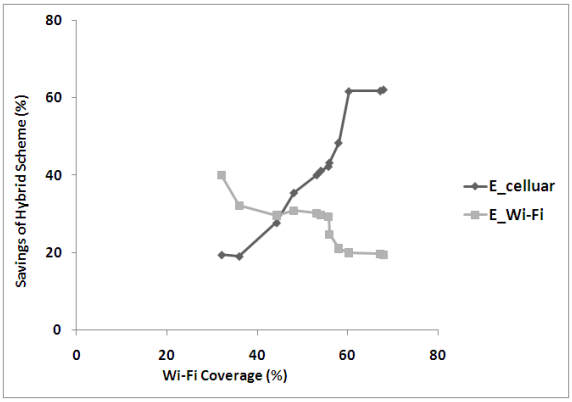

Figure 14: Savings compared to $E_{cellular}$ and E_{Wi-Fi} vary with overall Wi-Fi coverage

6.6 Landmark and Personal Devices

Now we evaluate the clustering techniques for identifying landmark and personal devices.

Landmark Devices: Users were given a bluetooth mouse to be kept at their house during the log collection process. This is the landmark device we aim to identify. Figure 15 plots the diversity of the devices in our log for user U2. Even if a device is sighted only at the same location, the requirement that it has to be sighted every time the user is at that location ensures that we correctly identify the landmark devices without false positives or false negatives.

Note that the majority of the devices have a very low value of *diversity*. This implies that users have a spatial corelation with the bluetooth devices they encounter − users see most bluetooth devices only at select locations. We consider this as a validation of our technique for using bluetooth contact-patterns for providing context.

Personal Devices: Along with our i-mate PDA, users carried a bluetooth headset with them most of the time and also their personal phones with bluetooth on always. Figure 16 plots the fraction of entries in which a device occurs in the log for user U2. The two devices at the top correspond to the user's phone and bluetooth headset.

6.7 Collaborative Prediction

Sharing of Wi-Fi information among devices helps improve the coverage of *Blue-Fi*. In our logs, we calculated the fraction of times when at least one bluetooth device

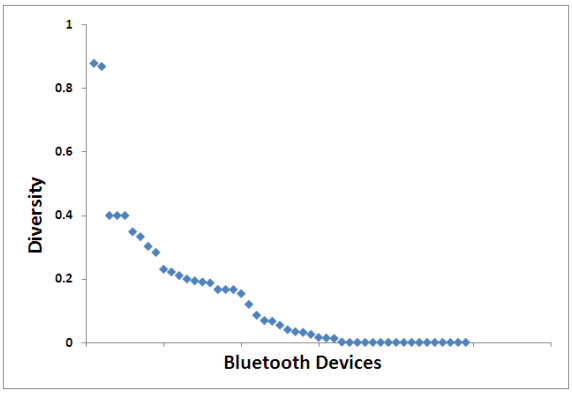

Figure 15: Distribution of diversity of bluetooth devices

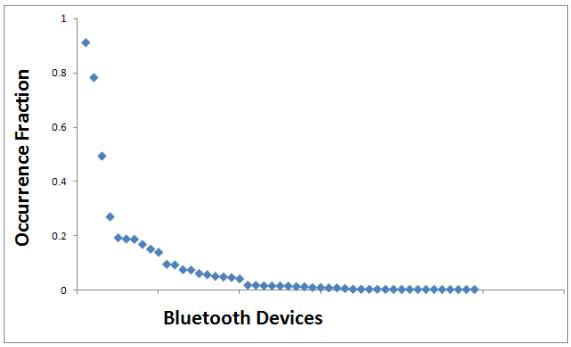

Figure 16: Distribution of the fraction of occurrence of bluetooth devices in the log

(not a landmark device) was present in the log and a Wi-Fi network from W_c was available. This represents the upper-bound on the coverage achievable by peer-to-peer sharing. The accuracy with peer-to-peer sharing can be 100% if correct information is provided by the nearby devices. Table 6 lists these numbers and contrasts it with the coverage of the bluetooth based prediction scheme. We compare it with bluetooth based prediction because it has the highest accuracy. Global sharing takes advantage of landmark devices too and hence achieves a higher coverage. Overall, we see an encouraging result that coverage can reach up to 90% presenting an overall improvement of up to 36.2%. Note that the accuracy of the prediction depends on the correctness of the information provided by the devices.

7. ENHANCEMENTS TO BLUE-FI

We now present useful additions to *Blue-Fi* – detecting open access points that are browser-authenticated, and multi-hop bluetooth discovery.

7.1 Connectable Access Points

Recall that *Blue-Fi* needs to know the list of Wi-Fi networks it is allowed to connect, W_c. Most devices store the set of Wi-Fi networks to which it has connected in the past [13]. Commonly used operating systems like Windows XP, Linux or Mac OS, this list is stored under "*preferred networks*". W_c is initialized with this list.

In addition to predicting Wi-Fi availability, *Blue-Fi* also

User	$\tau = 0.8$	p2p	Global
U1	59.6%	68.3%	79.4%
U2	57.8%	72.2%	84.5%
U3	51.1%	78.4%	86.3%
U4	73.4%	78.8%	89.1%
U5	74.2%	81.8%	90.1%
U6	50.2%	66.5%	88.1%
U7	58.8%	71.1%	90.1%
U8	61.2%	69.4%	83.3%
U9	44.6%	65.1%	80.8%
U10	60.2%	70.8%	76.4%
U11	60.1%	72.2%	77.3%
U12	64.5%	70.9%	73.2%

Table 6: Improvement in Coverage because of peer-to-peer and global sharing

aims to learn new access points that provide connectivity, i.e., open or unsecured wireless networks. However, the presence of an open network does not automatically imply connectivity. Often times, web requests are redirected to authentication pages and this section aims to detect such access points and discard them from prediction operations. We choose not to employ an exclusive and dedicated server for testing redirection as we believe it is more advantageous in terms of deployability and scalability to use popular webservers already available on the Internet. Since majority of the traffic on the Internet is HTTP based, we test redirection only for port 80. Note that the problem of picking the best available access point is orthogonal to our work.

Redirection happens after the device has connected and obtained a DHCP assigned address. Web requests are initially redirected to an authentication webpage. But note that redirections (HTTP response code 301) in themselves do not represent browser-authentication as they are routinely used in many web servers. We present two techniques that detect browser-authentication by essentially making a request for a URL with known characteristics and comparing if the response is as expected.

1. **Response Size:** Typically authentication webpages are a few 10's to 100's of kilobytes. *Blue-Fi* makes a request for a large file (say, a few megabytes) whose size is known, and check the size of the response.

2. **Secure HTTP:** Authentication webpages use the secure version of HTTP (*https://*). A secure response (*https://*) for a request for an unsecured webpage (*http://*) indicates browser-level authentication.

The techniques presented above are used to identify open networks that require browser-authentication and are discarded. The rest can be passed to a Wi-Fi profiling system like Virgil [9] for further connectivity tests (like throughput and latency), the results of which would be stored in W_c.

We evaluate the accuracy of the identification of browser-authenticated Wi-Fi networks. We collected 19 redirected webpages from open access points in coffee shops, airports and universities. We used webpages from popular webservers for our testing. Requests for large files were made to podcast media files on popular news webservers (e.g., on *cnn.com* and *nytimes.com*). The popular webservers have archiving facilities allowing access to old files. We also made

requests to popular webservers that did not require communication over secure HTTP (*google.com* and *live.com*).

Checking using the size of the response is most accurate identifying all the pages in our redirected pages dataset. The dataset had an average size of 62 KB and a minimum and maximum of 13 KB and 132 KB respectively. Secure HTTP identifies only 68.4% of the authentication pages. This is because splash screens and end-user-license-agreement (EULA) notices are also served over unsecure HTTP. Differentiating between authentication webpages, and EULA and splash screens is part of future work.

7.2 Multi-hop Bluetooth Discovery

The discrepancy between the Wi-Fi and the bluetooth ranges (i.e., 100m vs 10m) is the main reason behind the high accuracy of bluetooth prediction. On the downside, this discrepancy leads to low coverage, as a bluetooth device may not be able to discover other nearby devices, even though all devices are in the same Wi-Fi coverage.

One possible solution to reduce the discrepancy between the Wi-Fi and bluetooth ranges is to use multi-hop bluetooth discovery. A device establishes a connection with each device present in its immediate range, and queries for devices that are in its range, recursively, for a preset number of hops. This way, a mobile device will discover bluetooth devices beyond its bluetooth range. This technique allows one to trade accuracy for coverage: the higher the number of hops the higher the coverage and the lower the accuracy. The overhead of multi-hop bluetooth discovery can be reduced using the technique described in Section 3.

We performed an initial evaluation of the multi-hop discovery protocol by placing bluetooth monitoring devices in a large lab area where the bluetooth devices tend to be clustered in two groups. We were able to see an increase of up to 234% in the number of discovered bluetooth devices by using multi-hop discovery.

8. DISCUSSION

In this section, we discuss extensions to *Blue-Fi* along with fugure work: deploy "reference" devices to increase prediction coverage, and use correlated observation of bluetooth devices to improve accuracy.

8.1 Reference Bluetooth Devices

Blue-Fi depends fundamentally of the existence of landmark bluetooth devices. However, as the coverage results suggest, in a non-trivial number of cases, the mobile device is not able to identify such bluetooth devices. To address this issue, we propose *BlueDust*, a pervasive deployment of *reference* bluetooth devices spatially distributed in a given area. These devices are stationary and serve as means of providing contexts to the user. We envisage reference devices to be simple and inexpensive, for example bluetooth USB dongles or bluetooth mice. In contrast to Wi-Fi based indoor monitoring systems [20, 16], *BlueDust's* advantage is power efficiency. Also, a clear benefit of such a system would be increase in coverage for *Blue-Fi*.

A simple and practical deployment mechanism, especially for enterprise environments, is to plug in bluetooth dongles into desktops [20]. Reference devices can also be placed using systematic techniques. A database contains the locations of the reference devices.

BlueDust can be used for monitoring the spatial variation of resouces like cooling (e.g., *conference room A is uncomfortably cold*) and Wi-Fi performance (e.g., *Connectivity in office B is spotty*). Users log their sensed data (like Wi-Fi performance) along with the visible reference bluetooth devices. The database of the locations of the reference bluetooth devices can be used to find the location of the point at which the data was collected and appropriate analysis can be conducted. We plan an extensive deployment and evaluation of *BlueDust*.

8.2 Prediction Models

Blue-Fi assumes that the sensing of each bluetooth device is an independent event. But clearly, there is a rich opportunity to incorporate mobility patterns and correlation across multiple bluetooth devices in our prediction models. Mobility patterns can be used for predicting Wi-Fi availability — e.g., "Wi-Fi connectivity will be available *ten minutes* after you spot the bluetooth device b". Such predictions are useful for delay-tolerant applications to appropriately plan their network activities in future.

Also, $predict_{BT}$ can be augmented to include sets of bluetooth devices. This is useful in scenarios when, say, bluetooth devices b_1 and b_2 are not sufficiently reliable on their own, but whenever sensed together turn out to be reliable indicators of Wi-Fi availability. Understanding the benefits of such a correlated predictions on *Blue-Fi's* coverage and accuracy is part of future work.

9. RELATED WORK

Improving energy efficiency of Wi-Fi networks is a long-standing problem in wireless networks and has been approached from different directions. Techniques range from protocol optimizations in the various layers of the networking stack for a single Wi-Fi radio to techniques that leverage multiple radios on the same device that often involve specialized infrastructure elements.

The key idea of using a separate low-powered radio to wake up a high-powered radio was proposed in Wake-on-Wireless [15]. Wake-on-Wireless proposes the use of a second special-purpose radio that serves as a wake-up channel for a Wi-Fi radio. However the short range custom radio necessitates multiple intermediate proxies and presence servers. Also, it requires significant modifications to existing mobile devices. While On-Demand-Paging [27] builds on this idea to use the widely available Bluetooth radio as the low-powered channel, it still needs substantial infrastructure support in the form of specialized access points that have both Wi-Fi and bluetooth interfaces. Cell2Notify [28] uses the cellular interface to wake up the Wi-Fi interfaces on an incoming VOIP call using specialized servers. In contrast, *Blue-Fi* does not require any modification to the existing infrastructure and can be deployed readily on mobile devices.

War-driving has been performed in prior work that collects and maps the Wi-Fi access points [7, 10, 24]. War-driving is an expensive operation requiring investment of time and money. While a few major cities have good war-driving data, the majority of them have scarce or no mapping. In contrast, *Blue-Fi* adopts the approach of users automatically learning their own sorroundings and Wi-Fi availability. Also, war-driving data tend to become unreliable and out-dated over time.

Context-for-Wireless [12] aims to provide energy-efficient ubiquitous wireless connectivity, and their ideas of using

cell-tower information for predicting Wi-Fi availability are closely related to our work. But as we demonstrate cell-tower information is coarse-grained and can be beneficially combined with bluetooth contact-patterns for fine-grained context localization. Intel Place Lab [19, 10] collected extensive network data on GSM cellular networks for the sake of positioning Wi-Fi hotspots. To the best of our knowledge, ours is the first work that demonstrates the benefits of using bluetooth contact-patterns (including mobile bluetooth devices) for context localization. We are not aware of prior work that demonstrates a beneficial convergence of the bluetooth, cellular and Wi-Fi interfaces.

Our work has been supported by recent studies that show the existence of bluetooth contact-patterns in devices [11, 26]. CoolSpots [22] proposes the installation of "bluetooth access-points", which if deployed widely, strengthens our idea of using bluetooth contact-patterns for providing context information to mobile devices.

10. CONCLUSIONS

Blue-Fi proposed using bluetooth contact-patterns as context identifiers for predicting Wi-Fi availability. The low range of bluetooth devices make them accurate predictors of Wi-Fi availability; we compensated for their lack of coverage by using cell-tower identifiers. Our evaluation with data collected from real users' contact-patterns shows encouraging results in providing coverage as well as accuracy.

In addition, we also presented techniques to overcome bluetooth's high discovery time essentially using periodic scanning. To speed up the learning process, *Blue-Fi* uses collaborative prediction through sharing of logs and Wi-Fi connectivity details.

11. ACKNOWLEDGMENTS

We would like to thank Joe Hellerstein , Dilip Joseph, Anthony Joseph, Randy Katz, Venkat Padmanabhan, Lucian Popa and Matei Zaharia for their advice and guidance regarding various technical aspects as well as writing of this paper. We would also like to thank our shepherd, Albrecht Schmidt, and the anonymous reviewers for their comments and suggestions.

12. REFERENCES

[1] Companies Eye Location-Services Market. *http://online.wsj.com/article/SB122722971742046469.html*.

[2] IMAP IDLE monitor/notifier. *http://www.hackint0sh.org/forum/f126/6116.htm*.

[3] In The Hand. *http://32fet.net*.

[4] Is WiFi draining when not connected. *http://forums.macrumors.com/archive/index.php/t-330279.html*.

[5] Microsoft Pocket Outlook. *http://www.microsoft.com/windowsmobile/en-us/downloads/microsoft/office-outlook-mobile.mspx*.

[6] Spearman Rank Correlation Coefficient. *http://mathworld.wolfram.com/SpearmanRankCorrelationCoefficient.html*.

[7] WIGLE: Wireless Geographic Logging Engine. *http://wigle.net*.

[8] A. J. Nicholson and B. D. Noble. BreadCrumbs: Forecasting mobile connectivity. In *Proceedings of The Fourteenth Annual International Conference on Mobile Computing and Networking (Mobicom '08)*, Sep 2008.

[9] A. J. Nicholson, Y. Chawathe, M. Y. Chen, B. D. Noble, and D. Wetherall. Improved access point selection. In *Proceedings of The Fourth International Conference on Mobile Systems, Applications, and Services (MobiSys '06)*, June 2006.

[10] A. LaMarca, Y. Chawathe, S. Consolvo, J. Hightower, I. Smith, J. Scott . Place lab: Device positioning using radio beacons in the wild. In *Proceedings of The Third International Conference on Pervasive Computing (Pervasive '05)*, May 2005.

[11] A. Natarajan, M. Motani, and V. Srinivasan. Understanding Urban Interactions from Bluetooth Phone Contact Traces. In *Proceedings of The Eighth Passive and Active Measurement Conference (PAM '07)*, Apr 2007.

[12] A. Rahmati and L. Zhong. Context-for-Wireless: Context-Sensitive Energy-Efficient Wireless Data Transfer. In *Proceedings of The Fifth International Conference on Mobile Systems, Applications, and Services (MobiSys '07)*, Jun 2007.

[13] B. Greenstein, R. Gummadi, J. Pang, M. Y. Chen, T. Kohno, S. Seshan, and D. Wetherall. Can ferris bueller still have his day off? protecting privacy in the wireless era. In *Proceedings of The Eleventh Workshop on Hot Topics in Operating Systems (HotOS-XI)*, May 2007.

[14] E. S. Hall, D. K. Vawdrey, and C. D. Knutson. RF Rendez-Blue: reducing power and inquiry costs in Bluetooth-enabled mobile systems. In *Proceedings of The Eleventh International Conf. on Computer Communications and Networks (ICCCN '02)*, Oct 2002.

[15] E. Shih, P. Bahl, and M. J. Sinclair. Wake on wireless: An event driven energy saving strategy for battery operated devices. In *Proceedings of The Eighth Annual International Conference on Mobile Computing and Networking (MobiCom '02)*, 2002.

[16] F. Dabek, R. Cox, F. Kaashoek, and R. Morris. Vivaldi: A decentralized network coordinate system. In *Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM '04)*, Aug 2004.

[17] G. Ananthanarayanan, V. N. Padmanabhan, L. Ravindranath, and C. A. Thekkath. COMBINE: Leveraging the Power of Wireless Peers through Collaborative Downloading. In *Proceedings of The Fifth International Conference on Mobile Systems, Applications, and Services (MobiSys '07)*, Jun 2007.

[18] Ke Chen. On k-Median Clustering in High Dimensions. In *Proceedings of The Seventeenth ACM-SIAM Symposium on Discrete Algorithms (SODA '06)*, Jan 2006.

[19] M. Chen, T. Sohn, D. Chmelev, D. Haehnel, J. Hightower, J. Hughes, A. LaMarca, F. Potter, I. Smith, and A. Varshavsky. Practical Metropolitan-Scale Positioning for GSM Phones. In *Proceedings of The Eighth International Conference on Ubiquitous Computing (UbiComp '06)*, Sep 2006.

[20] P. Bahl, R. Chandra, J. Padhye, L. Ravindranath, M. Singh, A. Wolman and B. Zill. Enhancing the security

of corporate wi-fi networks using dair. In *Proceedings of The Fourth International Conference on Mobile Systems, Applications, and Services (MobiSys '06)*, Jun 2006.

[21] P. Mohan, V. N. Padmanabhan, and R. Ramjee. TrafficSense: Rich Monitoring of Road and Traffic Conditions using Mobile Smartphones. In *Microsoft Technical Report, MSR-TR-2008-59*, Apr 2008.

[22] T. Pering, Y. Agarwal, R. Gupta, and R. Want. Coolspots: Reducing the power consumption of wireless mobile devices with multiple radio interfaces. In *Proceedings of The Fourth International Conference on Mobile Systems, Applications, and Services (MobiSys '06)*, Jun 2006.

[23] T. Sohn, A. Varshavsky, A. Lamarca, M. Chen, T. Choudhury, I. Smith, S. Consolvo, J. Hightower, W. Griswold, and E. de Lara. Mobility detection using everyday gsm traces. In *Proceedings of The Eighth International Conference on Ubiquitous Computing (UbiComp '06)*, Sep 2006.

[24] V. Bychkovsky, B. Hull, A. Miu, H. Balakrishnan, and S. Madden. A measurement study of vehicular internet access using in situ wi-fi networks. In *Proceedings of The Twelfth Annual International Conference on Mobile Computing and Networking (MobiCom '06)*, 2006.

[25] V. Otsason, A. Varshavsky, A. LaMarca, and E. de Lara. Accurate GSM Indoor Localization. In *The Seventh International Conference on Ubiquitous Computing (Ubicomp '05)*, Sep 2005.

[26] V. S. W. Wang and M. Motani. Adaptive contact probing mechanisms for delay tolerant applications. In *Proceedings of The Thirteenth Annual International Conference on Mobile Computing and Networking (MobiCom '07)*, Sep 2007.

[27] Y. Agarwal, C. Schurgers, and R. Gupta. Dynamic Power Management Using On Demand Paging for Networked Embedded Systems. In *Proceedings of The 2005 Conference on Asia South Pacific Design Automation (ASP-DAC '05)*, Jan 2005.

[28] Y. Agarwal, R. Chandra, A. Wolman, P. Bahl, and R. Gupta. Wireless wakeups revisited: Energy management for voip over wi-fi smartphones. In *Proceedings of The Fifth International Conference on Mobile Systems, Applications, and Services (MobiSys '07)*, Jun 2007.