Predicting Length of Stay at WiFi Hotspots

Justin Manweiler*, Naveen Santhapuri†, Romit Roy Choudhury†, Srihari Nelakuditi‡
*IBM T.J. Watson Research †Duke University ‡University of South Carolina

***Abstract*—Today's smartphones provide a variety of sensors, enabling high-resolution measurements of user behavior. We envision that many services can benefit from short-term predictions of complex human behavioral patterns. While enablement of behavior awareness through sensing is a broad research theme, one possibility is in predicting how quickly a person will move through a space. Such a prediction service could have numerous applications. For one example, we imagine shop owners predicting how long a particular customer is likely to browse merchandise, and issue targeted mobile coupons accordingly – customers in a hurry can be encouraged to stay and consider discounts. Within a space of moderate size, WiFi access points are uniquely positioned to track a statistical framework for user length of stay, passively recording metrics such as WiFI signal strength (RSSI) and potentially receiving client-uploaded sensor data. In this work, we attempt to quantity this opportunity, and show that human *dwell time* can be predicted with reasonable accuracy, even when restricted to passively observed WiFi RSSI.**

I. Introduction

The convergence of sensing, computation, and communication on modern smartphones is offering valuable insights into human behavior [3], [8], [9], [16]. We attempt to bring these insights into a relatively unexplored problem space. We ask: by leveraging the sensor readings from users located at a WiFi hotspot, can we predict how long a given user will stay nearby? We call this the user's *dwell time*. Predictions for length of stay at a WiFi hotspot can be beneficial for enhancing the wireless network, as we will consider in the later part of this paper, and also for domains outside of networking. For example, we imagine shop owners launching a targeted mobile marketing campaign with coupons to slow down hasty shoppers.

We design and implement *ToGo*, a general framework for behavior-aware dwell prediction. With ToGo, mobile devices periodically report their sensor readings to the AP (e.g., accelerometer and compass). The AP runs a machine learning algorithm that accepts the sensor readings as features of user-behavior, combines these with other passively-observed measurement features from WiFi RSSI, and periodically predicts the user's dwell time. In the airport, for example, compass directions combined with WiFi signal strengths may reveal a signature, adequate to identify that a user will soon exit the terminal. *The system learns the appropriate signatures using the initial set of users as the training set; there is no need for any hotspot-specific configuration.*

We evaluate ToGo through live experiments with real users at a university cafe. Our testbed is composed of 9 Google Nexus One phones (for mobile users) and a laptop posing as the AP. For larger-scale experiments, we record real user behavior, and mimic them at the university's library, cafe, and McDonald's. Although our machine learning-based approach is reflective of a first attempt in this nascent space, evaluation results show reasonable success in predicting client dwell duration for real human activities. We believe that further refinements can make ToGo a robust, dependable service.

While we fully concede that ToGo is a prototype, and can benefit from further large scale testing and tuning, particularly across a wide range of hotspots, traffic patterns, and human users. Nevertheless, we believe that results in this paper are promising, and justify a longer-term research engagement. The promise is particularly pronounced because, *with limited training, ToGo performed seamlessly for a completely uncontrolled experiment with live users*. With improved machine learning and activity recognition algorithms, ToGo may become a useful building block for different kinds of predictive applications.

Our main contributions are summarized as follows.

(1) **We design ToGo, a framework for predicting length of stay at WiFi hotspots.** ToGo leverages automatic self-training for live dwell predictions without any hotspot-specific configuration.

(2) **We implement ToGo on Google Nexus One phones and on a laptop-based WiFi AP.** Results with real patrons at public locations encourage our approach.

(3) **We present *BytesToGo*, a case study application of ToGo.** BytesToGo considers the opportunity of offloading 3G or 4G traffic to WiFi through predictive prioritization. We show that mobile device sensors may reveal user intentions, facilitating informed network decisions.

II. Motivation

We begin with a discussion of the important research questions motivating our work on ToGo.

Is dwell time worth predicting? Do distinct classes of human behavior exist in practice? ToGo relies on the hypothesis that mobile users dwell for different durations at a hotspot, and their dwell times correlate to their activity. To verify the diversity in dwell times, we visited a university cafe and set up a WiFi-enabled laptop as a traffic sniffer. We selected the three strongest APs in the cafe, and used `tcpdump` to monitor the distinct devices connected to these APs. Dwell time for each device was estimated as the time-difference between the first and the last time the device was visible to the sniffer. In five hours of a weekday afternoon (11am to 3pm), we detected 340 distinct devices. Figure 1 shows the CDF of their dwell times. More than one third of the devices dwelled for less than 10 minutes (e.g., the user had a quick lunch/snack); and even among them, half of the devices stayed for 2 minutes or less (coffee/food to-go). More than one fifth stayed at least two hours. There is clear diversity in dwell time.

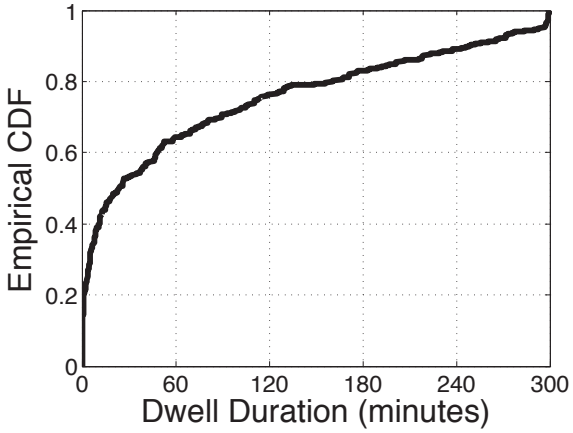

Fig. 1. Clients at a university cafe exhibit varied dwell times, reflecting multiple patterns of behavior. Some long-dwell clients study for hours while more mobile users take a meal to-go.

Predicting length of stay seems difficult and highly dependent on the location. How can ToGo operate effectively across a wide variety of locations and contexts? We design ToGo as a self-tuning system. Conveniently, the ground truth about the user's exact length of stay is available upon the user's departure—the AP knows the duration from client's association to disassociation. By exploiting the knowledge of ground truth, ToGo retrains itself and naturally adapts to the current trends in user behavior. Our measurements suggest that (1) observable trends exist in a variety of typical deployment settings; and (2) trends are readily recognizable by an application of established machine learning techniques.

Will ToGo require modifications on the mobile device to obtain their sensor readings? In testing ToGo, we consider the value of sensor feedback from clients, anticipating that mobile devices would run a background app to report live sensor measurements. Practically, this might pose a substantial barrier to an immediate deployment. However, we show that, while reported sensor feedback is potentially useful, a ToGo AP can utilize only passive observations from WiFi for dwell time prediction: uplink RSSI measurements, 802.11 data rates, and packet loss ratios. Moreover, controller-based WiFI solutions, such as those from Meru, Meraki, Aruba, Cisco, and other vendors, already aggregate such metrics across an entire enterprise. ToGo's accuracy can be potentially improved by combining these metrics across multiple APs.

III. Design and Implementation

Figure 2 illustrates the basic ToGo architecture. A module running at the AP gathers WiFi and sensor measurements, and predicts on them by leveraging past user behaviors at the same hotspot. The dwell time primitive can then be used for variety of applications, such as traffic shaping (considered in-depth later in this paper). We now describe the design of ToGo, followed by the details of functional components and our implementation choices, and finally evaluate its performance.

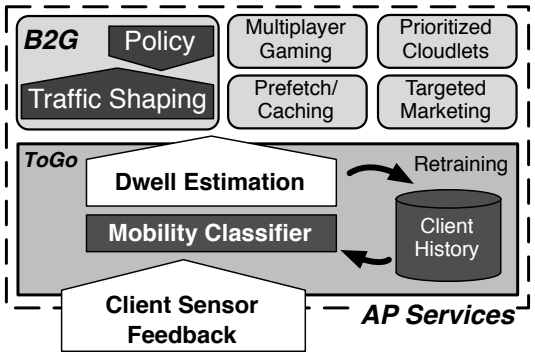

Fig. 2. ToGo synthesizes client sensor feedback to estimate dwell time for associated clients. Similar to B2G, other applications can leverage these predictions as necessary, for example, to ensure that multiplayer games will complete before one party leaves or to prioritize access to cloudlets [20].

A. Design Overview

ToGo clients on mobile devices export sensor readings to the AP. This could be an implicit operation, such as with RSSI, or explicit, as with acceleration, compass direction, etc. These sensor readings effectively reveal the features that characterize the user's behavior. The AP uses a support vector machine (SVM) to process these (multi-sensory) features and categorize the user's dwell time. Over time, the features from the same class of behavior begin to exhibit similarity. The SVM recognizes such similarities and employs them for prediction. Correct predictions reinforce the similarity; incorrect predictions imply that the feature set may not be sufficiently discriminating. The SVM learns from the failures and refines the prediction over time.

The (optional) client-side **Sensor Measurement Module** systematically probes phone sensors, extracts basic statistics, and periodically sends a summary to the ToGo server running on the AP. These may be viewed as a *timeslice of features*, capturing an instant of a user's micro-behavior. The summary amounts to a few bytes and sent over WiFi, incurring a small control overhead, which can be piggybacked on other upload traffic.

The **Dwell Time Prediction Module** operates on WiFi data and client sensor summary reports to predict the dwell time for clients as they arrive. Machine learning techniques are employed to classify each user into one of a few groups, corresponding to a coarse notion of expected dwell time. Our implementation uses five dwell time-classes on a *discretized logarithmic scale* (1-5), with lower values indicating a shorter expected dwell. In examining user behavior at a campus McDonald's, we found that broad types of user behavior cluster along this scale. For example, the dwell classes and corresponding behavior were often as follows: (1-2) walking past the restaurant, (2-3) taking food to-go, (4) buying food and eating in the restaurant, (4-5) studying in the dining area.

When a user walks in, her short-term mobility pattern may resemble the *walk-past-the-cafe* category, but as she stands in the queue near the counter, she may be moved to the *take-out* category. If she goes to pick up condiments, her pattern will resemble the 4th (*sit-down*) category. When the

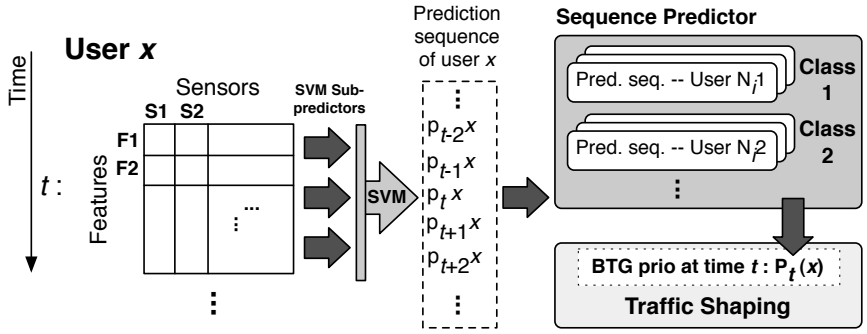

Fig. 3. Periodic Sensor-Feature matrices feed the SVM sub-predictors to generate short-term predictions. Time-indexed predictions form a growing sequence that are then used to predict the user's long-term dwell time class. B2G is an application that uses this prediction as an input to the traffic shaping module.

user finally leaves, ToGo can learn the truth about her dwell time, and use this data point to refine future predictions. With many users visiting a hotspot, we anticipate reasonably quick convergence to the true behavioral categories corresponding to that location. Thereafter, the predictions can become more accurate. The entire operation can be automatic, requiring no manual configuration or tuning.

B. Prediction Engine

Figure 3 presents the prediction engine underlying ToGo. We describe the key functional components next.

Feature Extraction. We used the Google Nexus One Phone which has a variety of sensing capabilities, including a three-axis accelerometer, light sensor, electromagnetic compass, and WiFi/GSM radios. A primary challenge in behavior classification is extracting effective sensor features that discriminate classes of user behavior. We experimented with various sensor features during the design stage of ToGo, before finalizing on those shown in Table I.

Sensor (S)	Feature (F)
Accelerometer	Directional intensity: *abs(x/y/z-axis acceleration)*
	Directional cumulative sum: *windowed sum of abs(x/y/z)*
	Direction-agnostic intensity: *averaged summed accelerator vector*
	Direction-agnostic cumulative sum: *windowed summed accelerator vector*
802.11 Radio	Upload/download signal strength: *RSSI mean, std. dev., histogram*
	Transmission rate: *bitrate mean, std. dev., histogram*
GSM Radio	Signal strength: *mean, std. dev., histogram*
Light Sensor	Light intensity: *mean, std. dev., histogram*
Compass	Compass angle: *mean, std. dev., histogram*

TABLE I
FEATURES EXTRACTED FROM EACH SENSOR

Short-term Predictions. The mobile device periodically generates the sensor-feature matrix and sends it to the AP. Each feature is computed over a moving time window to capture the short-term user behavior. A SVM classifier accepts the matrix, and based on training data from the past, predicts the user's likely dwell class. Fig. 3 shows that for a user x at time t, the SVM sub-predictor yields a class of p_t^x. Of course, this prediction does not capture long-term behavior – a person going to the restroom in a cafe may be mispredicted as leaving the cafe. However, this short term predictor is useful for obtaining quick predictions. Note that observing a user over the long term can yield high prediction accuracy; however, that may be far too late to be of help for an application. Instead, ToGo starts making quick predictions as soon as the user enters the hotspot, and continues to refine its guess over time.

Sequence Prediction. The series of time indexed short-term predictions form an increasing sequence over time, i.e., $\phi(t)$ = $<p_1^x, p_2^x, p_3^x ... p_t^x>$, where t is the current time. This may be viewed as a growing signature, that incrementally reveals the nature of the user's behavior. Of course, once a user leaves, the complete signature can be recorded, and her true dwell time learnt. During bootstrap, B2G records these sequences and dwell times, and trains itself with them – we call this the Sequence Predictor (Fig. 3). Clusters of sequences represent distinct classes of long-term behavior, characteristic of that hotspot. Now, as a user begins to dwell inside the hotspot, her partial sequence, $\phi(t)$, is matched against the recorded sequences. The resulting prediction begins to better reflect the user's long-term behavior.

Coping with Time-varying Behavior. Human mobility patterns may be dependent on time of day (customer behavior may differ between breakfast, lunch, and dinner times) and day of week. However, this does not pose a problem for ToGo as long as most customers exhibit similar behavior during a given time span. This is because "time" is also a feature in our system, and the SVM identifies that distinct clusters can be created using time as the dominant discriminator. In summary, ToGo can automatically adapt to hotspot-specific behavior (Starbucks versus McDonald's), as well as to variations in time (afternoons versus evenings).

C. Prototype Implementation

At the ***Mobile Client*** (Google Nexus One phone), a lightweight Java background process periodically probes its

sensors to create a summary report, representing the last few seconds of user behavior. The client forwards this report to the AP as a single datagram packet. The ToGo ***Hotspot AP*** runs on an Ubuntu 9.10 laptop (Linux kernel 2.6.31) with an Intel Core 2 Duo CPU, 3 GB RAM, and an Atheros chipset D-Link DWA-643 ExpressCard WLAN interface using the `ath9k` driver. The `hostapd` daemon software provides a fully-compliant 802.11b/g/n AP. We built our dwell time prediction module on top of *Click Modular Router* [14]. Click provides a convenient, high-performance mechanism to intercept client traffic to record RSSI and bitrate from upload packets. We use the `libsvm` C++ SVM library for implementing the dwell time prediction module.

D. Performance Evaluation

Our evaluation of ToGo focuses on the ability of the dwell time prediction module to accurately classify a client's dwell time. The device dwell time (used interchangeably with *user dwell time*) is defined as the total time the mobile device remains associated to the hotspot. ToGo starts predicting dwell time as soon as the user connects to the hotspot, and refines its guess across time. Accordingly, our results present ToGo's accuracy as a function of time. Naturally, we would expect the prediction accuracy of any reasonable scheme to improve across the dwell duration.

Comparative Schemes. Four variants of ToGo are evaluated: *NoFeedback*; *Basic*; *Basic+Compass*; and *Basic+Compass+Light*. In *NoFeedback*, client feedback summary reports are disabled. The WiFi AP infers user behavior strictly from time and (upload) RSSI/bitrate. NoFeedback requires no client-side changes, hence is compatible with all legacy devices. *Basic* generates client reports composed of accelerometer readings, GSM signal strengths, and (download) WiFi RSSI/bitrate. *Basic+Compass* adds electromagnetic compass, found in newer smartphones. It is also representative of the most feature-rich devices when placed in a pocket or purse. *Basic+Compass+Light* adds a light sensor to account for the case where the phone is exposed to the ambience. Finally, we also include a trivial-but-reasonable scheme, called *Naive*. This scheme predicts dwell time classes only based on the duration that the device has already stayed in the hotspot. Specifically, let $[t_1, t_2)$ correspond to class i and $[t_2, t_3)$ correspond to class $i+1$. Naive predicts class i until $(t_1+t_2)/2$ and $i+1$ until $(t_2+t_3)/2$, and so on.

To evaluate dwell time prediction accuracy, we use a metric, ***Mean Dwell Misprediction***, defined as follows. Assume that user u's true dwell time, δ^u_{true}, maps to a dwell class P^u_{true}. ToGo's goal is to converge to this dwell class as soon as possible, and maintain it until the user leaves. At a given time t, ToGo predicts dwell time as $\delta^u_{predict}(t)$, which maps to $P^u_{predict}(t)$. The instantaneous prediction error at t can be expressed as $D^u(t) = P^u_{true} - P^u_{predict}(t)$. We evaluate the prediction error across all N users at time t, as:

$$MeanDwellMisprediction(t) = \frac{\sum_{u=1}^{N} |D^u(t)|}{N} \quad (1)$$

All our accuracy results are obtained using *cross-validation*. The ToGo AP records micro-mobility signatures for each client, as a function of time. Offline, the AP attempts to predict the dwell time for a particular signature after training the SVM on all other signatures. Accuracy is reported as the mean dwell misprediction at each time-step (the mean computed over all signatures).

Prediction with Real, Uncontrolled Users. We tested ToGo prediction accuracy at a campus coffee shop (hereafter referred to as the *Cafe*) with *15 real customers*. As they entered the Cafe, each customer is asked to carry a phone running ToGo client. We gave no instructions to the customers regarding how to carry or handle the phone. A ToGo AP is centrally deployed to collect data from the phones. The actual behaviors were as follows: two users of Class 1 (bought something and walked out in less than 1 min), seven users of Class 2 (waited in a queue and walked out in 1-3 mins), two users of Class 3 (waited for grilled food and walked out in 3-6 mins) and four users of Class 4 (bought food and ate at one of the tables). Figure 4 shows prediction error for all users across their stay. All ToGo schemes converge to the correct dwell class (0 mean error) in $\approx$ 2.5 minutes. Of course, this is the average across all classes; short dwell users (such as class 2) are assigned to their class within the first 30 seconds of arrival at the Cafe.

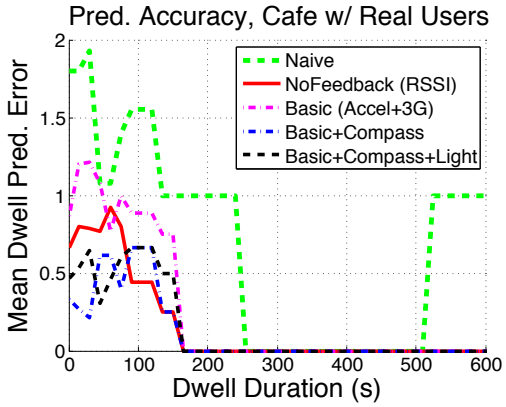

Fig. 4. Cross-validation on 15 real-user traces at the Cafe. With only 14 SVM training points, ToGo correctly classified users within 2.5 minutes.

Capturing User Behavior. Running uncontrolled experiments with real users is difficult at unfamiliar locations. To test ToGo at scale, we adopted an alternative methodology. We conducted a visual survey of people's movements at 3 different hotspots: a *Cafe*, a campus *Library* and a McDonald's frequented by students (*McD*). We tested our system at each of these locations, but for simplicity, we focus our discussion on the McD hotspot.

The floor plan of McD is shown in Figure 5. It is surveyed from 11am to 4pm, the busiest 5 hours on a weekday. We randomly picked users and drew their movement traces on a copy of the floor plan, along with timestamps for pauses. These recorded traces and the timestamps were used to mimic real user behavior. Figure 5 illustrates a user behavior along a representative path. Activities such as taking condiments

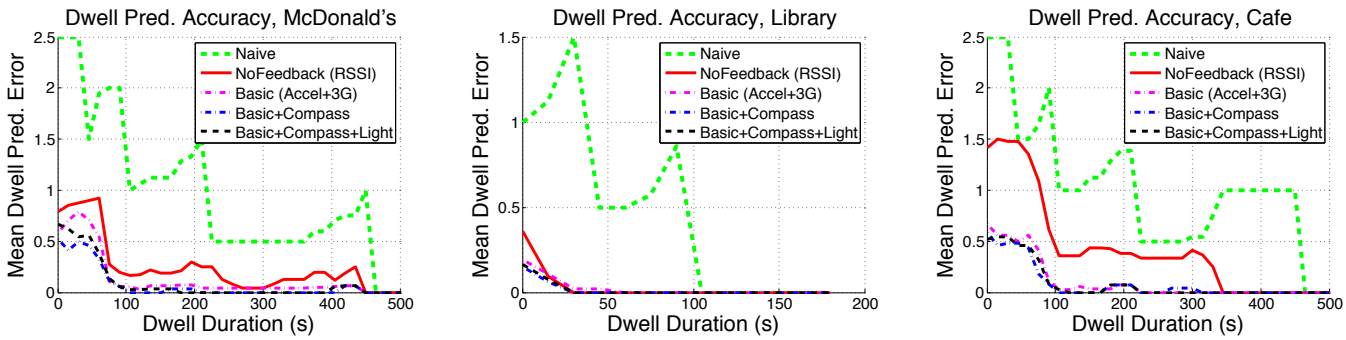

Fig. 6. Prediction accuracy at 3 hotspots: (a) McD; (b) Library; (c) Cafe. All ToGo variants perform better than Naive. NoFeedback performs reasonably well in library where there is enough RSSI diversity.

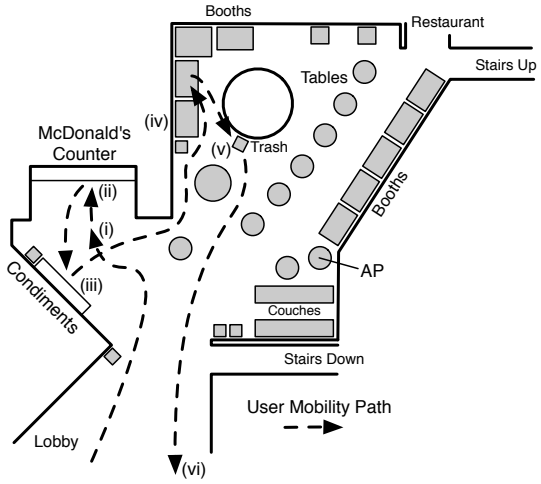

Fig. 5. Diagram shows user behavior along a representative path. User (i) walks up to McDonald's to examine wall-mounted menu and wait in queue line (10-60 seconds); (ii) places order, waits for food (1-2 minutes); (iii) takes condiments (2-15 seconds); (iv) sits and eats food (5-15 minutes); (v) discards trash (1-10 seconds); and (vi) exits to lobby.

indicate user's intentions to sit at a table and thus help discriminate different classes of users.

Emulating User Behavior. Having manually characterized the behavioral patterns at the McD, Library, and Cafe hotspots, we mimic random selections from the recorded behaviors holding a device running ToGo client. While reenacting, the dwell times of observed customers are proportionally shortened to reduce experimenter burden. We believe our reenactments are reasonably reflective of the original customers.

Prediction Accuracy and Sensor Contribution. We emulated 60, 72, and 48 user behaviors for the McD, Library, and Cafe hotspots, respectively. Figure 6 presents mean prediction error over the client dwell time for each hotspot. Again, all ToGo schemes substantially outperform the Naive (time-only) scheme. Variants with additional sensors and client feedback predict the correct dwell class sooner than the NoFeedback approach (RSSI/bitrate only). In the Cafe test, convergence time for the NoFeedback scheme is slower than the schemes utilizing client sensors.

Single User Behavior Prediction. Fig. 7 compares the prediction accuracy of different schemes for a specific user trace at McD shown in Fig. 5. In this trace, the user stayed within the hotspot for about 8 minutes. Hence, an ideal oracle-like predictor would place the user in dwell class 4 (5 to 10 mins) throughout the 8 minute dwell time. But any practical scheme needs to observe the user behavior for a while before converging to the correct dwell class. We chose this particular trace for comparison since 8 minute dwell time is expected to be sufficient for a good prediction scheme. In this instance, the users' behavior in the first 3 time slots (15 seconds each) matches closely that of class 3 users. At around 75 sec, this user picks up the food and moves towards the condiments. This behavior exhibited by class 4 users is different from the class 3 users. This results in a class 4 assignment to this user shortly thereafter. The convergence of NoFeedback to the correct dwell class is slower than the other schemes.

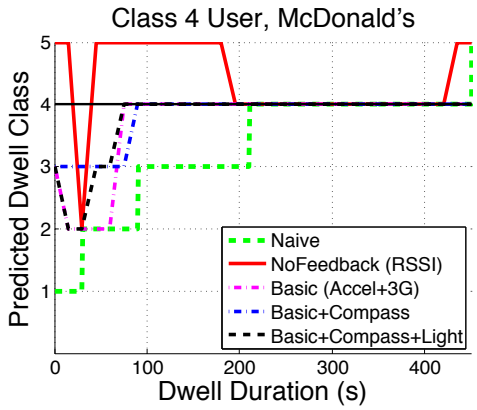

Fig. 7. Dwell class prediction for one user at McD. The user stayed in the WiFi range for ≈8 minutes. ToGo with client feedback converges to the correct dwell class 4 within 90 seconds.

In the next section, we demonstrate how an application can benefit from dwell time prediction. While we can imagine many applications such as multiplayer games mentioned before, we specifically focus on how traffic prioritization can be facilitated by ToGo framework.

IV. B2G: An Application of ToGo

We now present an application that leverages dwell time prediction to migrate would-be 3G traffic onto WiFi networks.

Briefly, by accelerating downloads of soon-to-depart users, less leftover demand is carried out onto 3G. This section motivates the application, implements it, and evaluates its benefits.

A. Dwell Prediction to Offload 3G Bandwidth

The rising density of mobile devices, combined with the availability of 3G data services, severely strains the cellular network [6]. Subscribers, especially in big cities, are experiencing deteriorating 3G quality, leading to widespread dissatisfaction. New classes of content-browsing devices, such as the Nexus 7 and Microsoft Surface, will further heighten this strain [4]. These devices will not only download more video and large sized pictures/eBooks, they will also do so *on-the-fly*. A student may begin downloading a Netflix movie while walking from her lab to the campus bus stop; a daughter may download an eBook while her mother drives out of their home garage, or picks up dinner from a drive-in restaurant. Such pervasive downloads explain the projected 39x increase in mobile data traffic by 2014. The 3x increase in cellular spectrum will be far less than adequate [11].

Responding to this concern, AT&T has announced an additional $2 billion investment to make sure it meets the growing demand for content consumption [19]. Part of this money will be invested in adding more WiFi access points (APs), *and effectively using them to assist 3G networks* [19]. Recent work has highlighted both the challenges and opportunity. Notably, Wiffler augments the effective capacity of mobile 3G networks by carefully exploiting WiFi [2]. It targets the particularly demanding case of vehicular access, where WiFi coverage is especially spotty. Motivated by this success, we identify a complementary scenario, applicable to pedestrian users connecting to a public access point.

Our intuition is simple. Among clients connected to WiFi hotspots, those likely to crossover sooner to 3G may be treated with proportionally higher priority. Prioritized traffic allows for greater data download to the user with short dwell time, reducing the burden that gets carried over to 3G. By more-aggressively satisfying mobile data demand during brief periods of connectivity, it is possible to enhance the return-on-investment for strategically-placed APs. Thus, airport travelers arriving at the baggage claim area could be allocated a larger bandwidth share over those walking towards check-in counters. Similarly, an iPad user beginning to walk away from the Starbucks AP can be prioritized over a seated laptop user. The AP could recognize these patterns from the phones' sensor readings, classify users into discrete dwell time categories, and prioritize them accordingly. Since per-user WiFi throughput is substantially higher than 3G (e.g., 3Mbps vs. 450kbps), a minute of WiFi prioritization can be valuable. We call our system *BytesToGo* (B2G), based on the observation that highly-mobile users download more bytes over WiFi before departing into the 3G network.

B. Potential Gains from B2G

We consider the potential of a B2G service, given an effective ToGo prediction API on which it could be based.

Just deploying a WiFi AP will substantially offload 3G networks – is B2G still necessary? We argue that the improvement from B2G should not be compared against the gains from WiFi deployments. B2G may be viewed as a software upgrade to make better use of WiFi APs, since they may anyway be installed in large numbers.

Earlier works have studied predictive offloading and handoff in cellular contexts [1], [13] and more recently in 3G and WiFi domains [2], [17]. Is B2G different? To the best of our knowledge, prior research has broadly focused on macro level behavior/mobility patterns, profiling how users transition between cells, encounter WiFi APs, or habitually dwell in them. B2G may be viewed as an attempt to exploit micro-level behavior/mobility, particularly via emerging opportunities in personal sensing and data mining. We believe that *micro-behavior guided networking* is relatively unexplored.

How much bandwidth can be offloaded from WiFi to 3G? The benefits from WiFi prioritization are proportional to the throughput difference between WiFi and 3G. When 3G throughput is considerably less than WiFi, a small increase in WiFi *utilization* can save considerable channel time on 3G. To characterize this difference, we measured the per-user WiFi bandwidth inside 8 different stores and 3 homes, and the corresponding 3G throughput just outside their coverage areas. Tests were conducted on different phones using a speedtest application from *dslr.net.* WiFi measurements were performed by walking through the hotspot, ensuring that transmission bitrates were not over-estimated. For 3G measurements, the user ran the speedtest app when located at the edge of the WiFi range, and walked away from the hotspot for 30s. Figure 8 reports an average of 6.64x higher TCP throughput over WiFi than with 3G, encouraging the prospects of predictive prioritization.

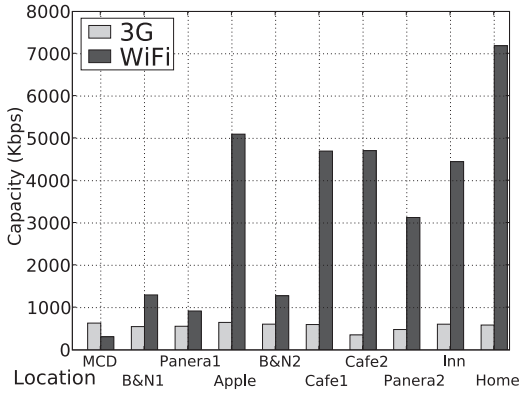

Fig. 8. Across hotspots, WiFi offers almost 6.5× throughput of 3G.

C. Design and Implementation of B2G

B2G supplies the predicted dwell time from ToGo as input to a **Traffic Shaping Module** that regulates the inflow of download TCP traffic from the Internet (see Figure 2). First, interactive traffic (e.g., VoIP) is identified (say by port number) and allowed to flow unimpeded at the highest priority. Non-interactive traffic is isolated by destination into per-client

queues. Each queue is allotted a maximum drain rate as a function of (1) the hotspot bottleneck bandwidth, (2) the total number of clients per priority class, and (3) the amount of spare capacity. Naturally, low-priority TCP sessions get rate limited, allowing high priority traffic to take a larger share of the backhaul bandwidth.

Traffic Shaping. The Linux kernel provides support for sophisticated traffic classification and rate-limiting. Specifically, we use the Hierarchical Token Bucket (HTB) queuing discipline. HTB distributes bandwidth according to the specified ratios up to a maximum cumulative rate, just below the bottleneck bandwidth (after accounting for unshaped interactive flows). For example, a priority level 2 user is expected to stay 3 times longer than a priority 1 user. Thus, she receives bandwidth in a 3:1 ratio to priority one clients. If there were two priority 2 users, and one priority 1 user, each priority 2 user gets $\frac{1}{5}$th of the capacity, while the priority 1 user gets $\frac{3}{5}$th. To prevent undue service degradation for low-priority clients, a minimum bandwidth must be assured before HTB ratios may be applied. Thus, during periods of especially high contention, B2G can degenerate itself to regular 802.11.

We used a Linux laptop as an AP and built prediction and prioritization module on top of *Click Modular Router* [14]. Additionally, our Click module observes bidirectional traffic patterns to feed into our traffic prioritization and shaping engine. Traffic shaping is conducted using the standard Linux Traffic Control subsystem. The Linux `TC` utility provides userspace hooks for live reconfiguration of the high-performance, in-kernel packet processing.

D. Performance Evaluation of B2G

Our B2G evaluation focuses on: (1) the effectiveness of traffic shaping to increase hotspot utilization; and (2) the amount of 3G bandwidth that can be saved by a B2G AP. The main findings from our evaluation are:

- Live traffic shaping, with accurate dwell prediction, successfully improves hotspot efficiency.
- Trace-based analysis suggests that B2G can save one half of a 3G channel per AP.

Next, we present our experimental assumptions followed by detailed performance results.

Traffic & AP. We assume that non-interactive traffic exerts the majority of the strain on 3G networks. This is already the case and is likely to get pronounced in future [4], [11]. Clients will download/upload full length movies, videos, picture-albums, eBooks, etc. Such types of traffic needs to be (and can be) offloaded to WiFi. We evaluate B2G with TCP download traffic in a single-AP system. We believe our approach is applicable to upload traffic as well as to multi-AP settings (more in Discussion). We also assume that the B2G AP owner is willing to provide selective treatment to the mobile clients.

User Demand. We assume that a client's "appetite" for data download is limited by the data consumption time, i.e., if a video takes 1 minute to download and 5 minutes to watch, the client does not initiate the next download until the end of 5 minutes. We believe this assumption models common data usage. Of course, the user might browse the web or send instant messages while buffering the video. To avoid delaying these applications, B2G does not prioritize interactive traffic.

We now evaluate the extent to which B2G can reduce 3G load. We present a live experiment of our complete implementation, highlighting the effectiveness of our design and implementation. Then, a trace-based evaluation characterizes 3G savings at scale. We begin by defining the performance metric next.

3G Time Saved. The 3G savings due to B2G arise from accelerating the download on WiFi before going to 3G. We sum the savings of individual users with prioritization, to get total savings. Let M^u_{prio} and M^u be the total downloaded WiFi data (with and without prioritization respectively) for user u. Then the total *3GSavings* are:

$$3GSavings = \sum_{u=1}^{N} (M^u_{prio} - M^u) \tag{2}$$

Live Experiment. Using multiple experimenters, we simultaneously emulate previously-recorded user behaviors. Based on live dwell predictions, B2G (Basic+Compass+Light variant) performs dynamic traffic shaping. This experiment is performed in the *Cafe* after training the AP with the Cafe trace data. The arrival and departure time of mobiles (drawn from recorded patterns) are as follows (in seconds): (0, 660); (15, 80); (60, 120); (200, 560); (240, 310); and (360, 580). Arrival and departure times are illustrated in Figure 9. Each arriving device begins a 100MB HD 720P video download (typical of YouTube HD videos). The video viewing length is about 6 minutes and 40 seconds. With just one device operating, it took about 75s to download the video (15 Mbps backhaul capacity). An experimenter emulating user behavior waits for the viewing duration of the video and, upon completion, starts another download of the same size (as if the user watches a series of videos, selecting the next, once the first completes). The complete experiment was repeated 3 times with and without prioritization. Figure 10 presents the results. B2G consistently achieves higher data transfer for shorter dwell clients, saving an average of about 55MB of 3G data per run.

Fig. 9. Arrival times and overlap of emulated live user behaviors.

Trace Based Evaluation. To test B2G 3G savings at scale, we conduct a trace-based evaluation. We consider both HD and non-HD video downloads. HD video parameters are the same as in the live experiment. For non-HD, we assume a 5 minute video at a 320kbps video encoding, with a size about 12MB, typical of non-HD YouTube videos [7], [10]. User arrival times are modeled based on real device arrival times obtained from the `tcpdump` (Fig. 1). The trace recorded a total of 340 devices, with 40 devices arriving per hour on average. Each user is assigned a mobility pattern (and corresponding dwell time) by randomly picking a trace from the McD hotspot. The

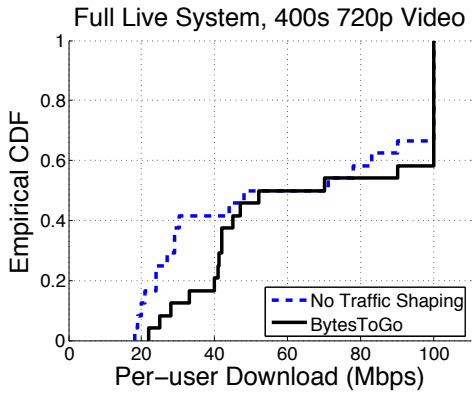

Fig. 10. Performance of B2G with live traffic prioritization in the Cafe hotspot. Traffic prioritization benefits clients with shorter dwell times.

total data downloaded by each class of devices for one hour is calculated based on the priorities (by extension bandwidth) assigned by the AP and the total WiFi capacity.

We conduct this experiment for each of our hotspots. However, in the interest of space (and similar results), we present 3G savings for only McD in Figure 11. Each data point reflects the mean of 100 trials. A hypothetical *Hindsight* scheme is shown for comparison. Hindsight "predicts" at the end of the trace with full knowledge of all client dwell times. For non-HD video, the B2G schemes save almost as much 3G data as Hindsight, approximately 100 MB/hour. At 3G rates, this equates to about 30 minutes of 3G channel time saved per hour, exclusively by prioritization. With HD video, the benefits of accurate prediction are better, boosting the savings to 45 minutes for the best B2G variant.

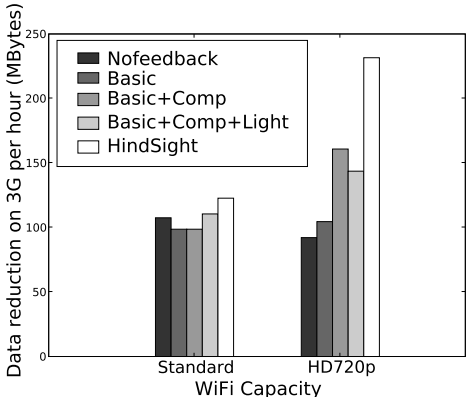

Fig. 11. 3G data saved per hour by one AP. B2G improves WiFi utilization, providing significant 3G network savings. Gains increase with larger HD files. Note that in some cases, RSSI based NoFeedback variant suffices to differentiate short dwellers.

V. Issues and Discussion

This section discusses a number of issues and open questions pertinent to ToGo and B2G.

Selecting the Right Policy. As in any form of prioritization, appropriate policy selection can be complex. In B2G, the AP owner must weigh the value of 3G bandwidth savings against quality-of-service for hotspot users. Without real-world preferences, B2G cannot optimize this tradeoff. However, traffic shaping parameters provide simple hooks to do so. Users may be assigned a minimum reservation bandwidth (as a function of the number of active users), and only the excess capacity may be prioritized by dwell time.

Energy Overheads. Accelerometer, compass, and light sensor usage incurs some energy cost. However, these sensors are often used for multiple applications concurrently, with very little marginal energy cost for additional applications. Further, for B2G prioritized clients, these costs would be at least partially mitigated by energy savings on WiFi compared to 3G. The NoFeedback scheme, passively relying on RSSI for upload WiFi traffic, does not incur any sensing cost.

What if a greedy user fakes sensor readings to get higher priority from a B2G AP? Currently do not have a mechanism to thwart fraudulent sensor measurements. However, without any self-reported data, the NoFeedback scheme would not be subject to selfish misbehavior. Further, all schemes could use a correlation with RSSI data in an attempt to identify cheaters and punish them with a low priority. We anticipate such an approach would be sufficient to disincentivize selfish users.

Multi-AP Hotspots. Thus far, we have assumed that ToGo would be deployed in a small hotspot location (e.g., a cafe) with only a single AP. In practice, a hotspot may have multiple APs extending over a larger coverage area. In these circumstances, a change of AP association should not be considered as a departure from the hotspot. ToGo naturally extends to these environments, especially when a collection of APs are administered by the same provider, say AT&T. A dedicated server or cloud application may serve as a network controller, in the style of the existing enterprise WLAN architecture. APs forward client feedback reports to the controller, paired with time, RSSI, and bitrate annotations. The controller performs training and prediction tasks for all APs in the hotspot. Aggregation of client data from multiple APs can also improve training quality. RSSI values for a client from non-associated APs can serve as additional features for prediction. With this RSSI feedback, along with knowledge of intra-hotspot AP-to-AP handoff patterns, the ToGo controller may provide a higher prediction accuracy than in the one-AP case.

Device Usage. For simplicity, our experiments have assumed that a user walking into a hotspot will have her device on running the ToGo client. Real behavior will exhibit greater diversity. Users may, for example, walk into a cafe, order food, and sit down all before turning on a ToGo-enabled device. In this case, the AP will have reduced information available for dwell prediction, possibly leading to increased error. However, we expect that these behavioral tendencies may also be learnable over time. Where a user activates her device may itself be a strong predictor of dwell time.

Alternative Learning Techniques. We chose to implement ToGo's prediction engine through SVM. A variety of other learning techniques such as Hidden Markov Models could be applied instead. We do not claim that SVMs are ideal, or our selections of SVM features, and believe it is possible to improve prediction accuracy through a more exhaustive

evaluation of alternatives. Further, the best learning technique for prediction might vary across locations. One possibility for performance improvement is to train multiple machine learning engines and dynamically select the current most-successful technique for future predictions.

Would B2G be beneficial even with 4G? The basic premise underlying B2G is that there exists a significant throughput difference between WiFi and cellular access technologies. While cellular network deployments are moving from 3G to 4G speeds, there are corresponding advances in wireless networks from 802.11n to 802.11ac. Furthermore, we believe *the main contribution of the paper is dwell time prediction, which gives valuable information to many applications, of which B2G is just one illustrative example.*

VI. Related Work

WiFi and Cellular. In CellShare [21] a rural WiFi network benefits from cellular network augmentation. The system allows the use of mobile phones to provide temporary Internet connectivity when parts of the network are disconnected. A similar architecture is used in CoolTether [22] to access the 3G network from the laptop using phone WiFi tethering. The scheme in [23] helps in recovering lost 3G multicast data by relaying on WiFi among neighboring devices. In MobTorrent [5], the cellular network is used as a control channel to predict mobility information and prefetch content. Our work makes use of WiFi to optimize the traffic offload from 3G. Wiffler exploits WiFi to reduce the load on 3G in vehicular networks [2]. Complementary to Wiffler, we are interested in offloading the 3G traffic onto WiFi during pedestrian hotspot connectivity.

Mobility Prediction. Macro mobility prediction for cellular networks is a well-researched area [1], [13], [15], [18]. BreadCrumbs [17] is a recent macro-mobility solution that harnesses habitual nature of human mobility. Based on a per-user history, Breadcrumbs makes use of a second-order Markov model to provide connectivity forecasts on which AP associations will be most useful. Our work complements BreadCrumbs, with a focus on micro-mobility and user behavior within the range of a single AP, and attempts to predict dwell time (with an approach similar to nth-order Markov model). Further, we aim to capture general behavior trends across clients, allowing the system to make informed decisions the first time a user associates to an AP.

Activity Recognition. Activity recognition has become an active research area in recent years [3], [8], [9], [16] due to the pervasiveness of sensor assisted phones. In our work, we build on this approach to capture the user behavior from phone sensor readings. There have been online-behavior aware schemes like Profile-Cast [12] but our work deals with physical-behavior awareness.

VII. Conclusion

This paper proposes ToGo, a system for predicting length of stay at WiFi hotspots, and BytesToGo, an example use case of the ToGo system. We address the challenge of predicting dwell time with and without the aid of client sensor data using machine learning algorithm at hotspot APs. Evaluation over traces collected from 3 different hotspots and live experimentation has confirmed ToGo's prediction efficacy, enabling BytesToGo to offload substantial 3G/4G data onto WiFi. ToGo is able to achieve reasonable accuracy without any hotspot-specific configuration or manual training. Instead, it learns and adjusts over time, automatically adapting to natural human behaviors. As part of future work, we plan to explore the other uses of dwell time prediction, which we believe to be a generic and valuable primitive across several domains.

References

[1] I. F. Akyildiz and W. Wang. The predictive user mobility profile framework for wireless multimedia networks. *IEEE Transaction on Networking*, 12(6), 2004.

[2] A. Balasubramanian, R. Mahajan, and A. Venkatramani. Augmenting Mobile 3G Using WiFi: Measurement, System Design, and Implementation. In *ACM Mobisys*, 2010.

[3] L. Bao and S. Intille. Activity Recognition from User-Annotated Acceleration Data. In *Percom*, 2002.

[4] P. Bellaria. Message from the iPad: Heavy Traffic Ahead. http://blog.broadband.gov/.

[5] B. B. Chen and M. C. Chan. MobTorrent: A Framework for Mobile Internet Access from Vehicles. In *Infocom*, 2009.

[6] B. X. Chen. What is Wrong with 3G in iPhone 3G? http://www.wired.com/gadgetlab/2008/08/whats-wrong-wit/.

[7] X. Cheng, C. Dale, and J. Liu. Statistics and Social Network of YouTube Videos. In *IEEE IWQoS*, 2008.

[8] B. Clarkson, K. Mase, and A. Pentland. Recognizing user context via wearable sensors. In *ISWC*, 2000.

[9] D. M. et. al. Recognizing and Discovering Human Actions from On-Body Sensor Data. In *International Conference on Multimedia and Expo*, 2005.

[10] P. Gill, M. Arlittz, Z. Li, and A. Mahanti. YouTube Traffic Characterization: A View From the Edge. In *ACM IMC*, 2007.

[11] D. Hanchard. FCC Chairman forecasts wireless spectrum crunch. http://government.zdnet.com/?p=7401.

[12] W. j. Hsu, D. Dutta, and A. Helmy. Profile-Cast: Behavior-Aware Mobile Networking. In *IEEE WCNC*, 2008.

[13] M. Kim, D. Kotz, and S. Kim. Extracting a mobility model from real user traces. In *IEEE Infocom*, 2006.

[14] E. Kohler, R. Morris, B. Chen, J. Jannotti, and M. F. Kaashoek. The click modular router. ACM Trans. on Computer Systems, 2000.

[15] B. Liang and Z. J. Hass. Predictive distance-based mobility management for multidimensional PCS networks. *IEEE Transaction on Networking*, 11(5), 2003.

[16] E. Miluzzo, N. D. Lane, K. Fodor, R. Peterson, H. Lu, M. Musolesi, S. B. Eisenman, X. Zheng, and A. T. Campbell. Sensing meets mobile social networks: the design, implementation and evaluation of the cenceme application. In *ACM Sensys*, 2008.

[17] A. Nicholson and B. Noble. Breadcrumbs: forecasting mobile connectivity. In *MobiCom*, 2008.

[18] P. Pathirana, A. Savkin, and S. Jha. Mobility modeling and trajectory prediction for cellular networks with mobile base stations. In *ACM MobiHoc*, 2003.

[19] M. Reardon. ATT to Invest $2B in Mobile Network. http://www.cnn.com/2010/TECH/01/29/att.network.boost/index.html.

[20] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies. The Case for VM-based Cloudlets in Mobile Computing. *IEEE Pervasive Computing*, 8(4):14–23, 2009.

[21] A. Sharma, E. M. Belding, and C. E. Perkins. Cell-Share: Opportunistic Use of Cellular Uplink to Augment Rural WiFi Mesh Networks. In *IEEE VTC*, 2009.

[22] A. Sharma, V. Navda, R. Ramjee, V. N. Padmanabhan, and E. M. Belding. Cool-Tether: Energy Efficient On-the-fly WiFi Hot-spots using Mobile Phones. In *ACM CoNext*, 2009.

[23] K. Sinkar, A. Jagirdar, T. Korakis, H. Liu, S. Mathur, and S. Panwar. Cooperative Recovery in Heterogeneous Mobile Networks. In *IEEE SECON*, 2008.