

# Optimizing upload time of data from mobile devices

Ying Zhu

*Faculty of Engineering and Applied Science  
University of Ontario Institute of Technology  
Oshawa, Ontario, Canada  
Email: ying.zhu@uoit.ca*

**Abstract**—The technological advances in mobile devices have resulted in increasing numbers of applications on them, and users use these applications to create lots of data. Due to the truly ubiquitous nature of these devices, the data is created potentially at all times of the day and in different wireless environments, e.g., at home, on the subway, at a coffeeshop. These data must eventually be uploaded to the server cloud from the mobile device by connecting to a wireless network. In different environments, the available wireless networks change, and the cost of connecting to them in order to upload these data also changes. For different applications, the preference for the uploading time for their data may be very different. We study the problem of optimizing the time for uploading based on the preference measure as well as the cost of the wireless network available at the time of uploading. Furthermore, because the wireless environment is dynamic and nondeterministic in the future, we use machine learning techniques for modeling user mobility patterns and predicting the wireless network in the near future, to assist in finding the optimal uploading time.

## I. MOTIVATION

The proliferation of mobile handheld devices has been accompanied by the increased capabilities of these devices. They are fast becoming as indispensable and versatile in applications as the older laptops and PCs. Users use the applications on their mobile devices at home, during commute, at work, at coffee shops, etc; and while doing so, lots of data items of different types are created, such as photos, to-do lists, calendar events, e-mails, blog entries, etc. These applications running on mobile devices are distinct from non-mobile applications in their *ubiquitous* data production. At any given time, a mobile device may be in a different environment than it was before or it will be after that time. The environment consists of the wireless networks that are available, the bandwidths offered by them, the costs of using them, and other possible characteristics such as privacy and security. When a data item is created, it is generally desirable to upload it and synchronize it to the networked storage provided by some remote servers — for the remainder of the paper, we will refer to these as the server cloud or simply, the cloud. The naive way would be to upload the data item to the cloud at the earliest possible time: either immediately if there is a wireless network available for the device to connect to, or connect to the first available network. However, because the environment for the mobile device is dynamic, and there

are considerations of cost and speed of data transfer, the naive strategy is often not optimal. Consider the following scenario: Ada is camping in a national park, takes lots of photos and writes a blog entry about her experience, all using her mobile phone. She can upload these to the cloud using GSM on her mobile phone, but it would take a long time to upload and she would be charged for it. The better strategy in this case is to wait until she gets back home and use the free WiFi (free since she pays a flat monthly fee) there to upload these items. On the other hand, there are cases when the data should be sent out as soon as possible, e.g., an urgent e-mail, or an important update that needs to be synchronized to a shared calendar.

We study the problem of finding the optimal upload time for data items created and stored temporarily on mobile devices. We assume a very general model: (1) Multiple applications create different types of data items. Each data item has an associated utility function that maps upload times to preferences. (2) The environment at time  $t$  consists of a set of different wireless networks (e.g., 3G, WiFi) that are available at time  $t$ , their bandwidths, and the costs of using them. This set is temporally dynamic and nondeterministic to predict at time steps in the future.

Essentially, instead of maximizing only the utility function (preference) for upload time, we apply utility theory and use a multi-objective scoring function to maximize expected utility while minimizing cost. Because both the utility and the cost depend on the type of wireless network used for uploading, we must find a way to *predict* which wireless networks will be available in the near (say, within 24 hours) future. We use machine learning techniques for this inference of the wireless environment based on historical observations, in order to make informed decisions on the upload times.

## II. RELATED WORK

We are not aware any previous work that specifically addresses the problem of optimizing the upload time for data created and residing on mobile devices. The authors of [10] developed a remote storage client for mobile phones. A middleware interface is presented in [3] for applications to share content despite differences in the underlying protocols. There is a body of work in distributed file systems, e.g., [5],

[4], [1], [6]. However, the focus of these work is on problems such as sychronization of files in the distributed storage environment, concurrency control, cache consistency; they are not concerned with the specific problem we study in this paper: of optimizing the time of uploading data from the mobile device.

We use the machine learning tool of a hidden Markov model (HMM) to describe typical user mobility patterns based on historical data, and then use the HMM for the inference task of predicting the future state of the user. There have been many studies done in the area of mobility management in wireless networks, e.g., [2], [11], [7]. In all of these work, the mobility pattern is modeled and based on historic records and predictions are made with the objective of improving the performance of location management and resource reservation, and enhancing the quality of service in the wireless network. In [2], a mobility model based on Markov chains is proposed that use historic records to predict future locations of mobile users. A two-dimensional Markov walk model is proposed in [11] to characterize the dynamic intercell movement. In [7], the user mobility pattern is predicted using neural networks. In contrast, we are not interested in the low-level intercell movement of the user, instead, we are only interested in dynamics in the set of available wireless networks. Moreover, we use a hidden Markov model (HMM) for the profiling of typical patterns and for the prediction of future states. We believe that an HMM is much better suited for our purpose because its inherent temporal structure fits well with the mostly predictable structure of a user’s movement during a 24-hour day.

### III. FINDING THE OPTIMAL DATA UPLOAD TIME

In this section, we formulate the problem of finding the optimal upload time for data items created on mobile devices, and describe our algorithm for solving it.

#### A. Utility of upload time

Different applications may produce data that have very different requirements for upload time; within the same application, different types of data may be created that do not share the same requirement. The e-mail application may produce regular e-mails that can tolerate delays versus urgent e-mails that must be sent as soon as possible. An application for editing blog entries may only need to upload new entries once a day, rather than immediately after they are written. Let  $A$  be the set of different types of data, categorized by their upload requirements.

We propose to use *utility functions*  $\{U_a : a \in A\}$  that map upload times to measures of preference. Figure 1 shows the utility functions of three types of data, reflecting the following requirements on upload times:

- It is *preferred* that data items of type  $a_1$  are uploaded sooner rather than later, with the preference decaying

linearly, and it is *required* that it is done 12 hours after their creation.

- It does not make a difference when data of type  $a_2$  are uploaded but it is *required* to be within the next day.
- It is *urgent* that data of type  $a_3$  are uploaded as soon as possible, as the preference for upload time decays exponentially.

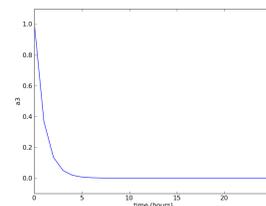
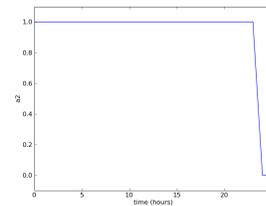
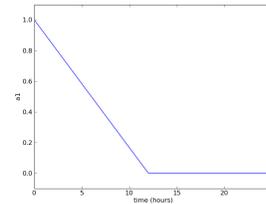


Figure 1. Three different utility functions for upload time.

It should be noted that the utility function for upload time is always monotonic (monotonically nonincreasing, in fact). This reflects the fact that if all other factors are equal, it is always preferable to upload a data item sooner rather than later.

An application may produce data items of only one type, or it could allow the user to choose to tag different data items with different types (e.g., non-urgent versus urgent). The upshot is that we assume that when a data item is created, it is assigned a type  $a \in A$ .

## B. Problem formulation

We have a sequence of data items  $d_1, d_2, \dots$  of different sizes being created by various applications on the mobile device at time  $t_0^1, t_0^2, \dots$ , respectively. At any given time, the mobile device can connect to a set of available wireless networks. Let  $W = \{w_i\}_{i=1}^n$  be the set of all  $n$  possible wireless networks and let  $W(t) \subseteq W$  be the ones that are available at time  $t$ , i.e., data can be sent by connecting to any wireless network  $w \in W(t)$ . Connecting to each  $w_i$  has an associated (nonnegative) cost,  $C : W \rightarrow \mathbb{R}$ . Each wireless network  $w_i$  also has a bandwidth  $rate : W \rightarrow \mathbb{R}$ . Suppose at time  $t_0$ , data item  $d$  of type  $a$  is created. The decision problem is to choose for  $d$  an upload time  $t_u$  in the future ( $t_u > t_0$ ) and a wireless network  $w$ ; i.e., at time  $t_u$ , upload  $d$  using wireless network  $w$  to the cloud. We have the following objectives:

- 1) Maximize the upload time utility,  $U_a$ .
- 2) Minimize the cost of using the wireless network,  $C(w)$ .

In order to be able to make such a decision at time  $t_0$ , we must have some information of the set of available wireless networks at time points in the future. Due to mobility, this set is dynamic over time and moreover nondeterministic. The information we need is the probability of  $w_i$  being available at time  $t, \forall t > t_0, i = 1, 2, \dots, n$ . Thus we must first infer these probability distributions and then use them in choosing  $t_u$  by maximizing utility and minimizing cost.

## C. Algorithm for choosing the upload time

We propose to use a scoring function to capture the trade-off between maximizing utility and minimizing cost,  $f : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$ , mapping upload time utility and wireless network cost to a single score. We use the scoring function  $f = U \cdot e^{\alpha C}$ ; the parameter  $\alpha$  can be tuned to control the preference between maximizing utility  $U$  and minimizing cost  $C$ .

Suppose that at any future time  $t$ , we can somehow determine the probability of each wireless network being available,  $P(w_i \in W(t)), \forall i$ . Then for data item  $d$  of type  $a$  created at  $t_0$ , the decision of its upload time  $t_u > t_0$  follows directly from the principle of *maximum expected utility* (MEU) from utility theory. The MEU principle says that we should choose the action that maximizes the expected utility. In our case,  $t_u$  should be chosen so that uploading  $d$  at time  $t_u$  maximizes the expected utility. Since we have a scoring function combining the two objectives of maximizing utility and minimizing cost, we slightly modify the MEU principle by choosing  $t_u$  to maximize the expected score ( $Ue^C$ ). At  $t_0$ , starting the upload of  $d$  at a given time  $t_u$  in the future is a nondeterministic action, due to the nondeterministic nature of the available wireless networks at  $t_u$ . For each  $t_u > t_0$ , the action of starting the upload at  $t_u$  has multiple possible outcomes of the *finish* time of uploading  $d$ , which depends on the size of  $d$ , the wireless

network used (nondeterministic) and its data transfer rate (also nondeterministic). A probability is therefore assigned to each possible outcome and the *expected score*,  $ES(t_u)$ , is calculated using the following formula:

$$ES(t_u) = \sum_i P(w_i \in W(t_u)) \cdot (U_a(t_f^i) \cdot e^{C(w_i)}), \quad (1)$$

where  $t_f^i = t_u + size(d)/rate(w_i)$

Note that  $t_f^i$  is the calculated finish time of starting the upload of  $d$  of  $size(d)$  at time  $t_u$  using the wireless network  $w_i$  with rate  $rate(w_i)$ . Also note that  $d$  is of type  $a$ , so the utility function for data type  $a$ ,  $U_a$ , is used.

To decide the best upload time, we simply choose the  $t_u$  that maximizes the expected score:

$$t_u^* = \arg \max_{t_u} ES(t_u) \quad (2)$$

Now it only remains to obtain the probability distributions of wireless network availability over time in the future,  $P(w_i \in W(t)), t > t_0, i = 1, 2, \dots, n$ . We describe our solution in the next section.

## D. Inferring probability distributions of available wireless networks in the future

For convenience, we discretize time into small time intervals of length  $\delta$ , so we have time points  $t_k, k = 1, 2, \dots$ , with  $t_{k+1} - t_k = \delta, \forall k$ .<sup>1</sup> We observe that on a mobile device, it is easy to obtain and record information of the available wireless networks at each time interval in the past. This information is obviously closely correlated to the mobility pattern of the user; for example, if the user is at home, then the free home WiFi is most likely available. That is, predicting available wireless networks can be done if user movement can be predicted. A recent influential study in *Science* [9] collected and analyzed extensive mobility data of cell phone users, and concluded that human movement is predictable 93% of the time. We observe that the predictable movement can be relied on to predict what we want, the probability distribution of available wireless networks in the future, or more precisely,  $P(w_i \in W(t)), i = 1, \dots, n$ .

We propose to apply machine learning techniques to make a *predictive inference* about the probability distributions of available wireless networks in the future, given the historical data (as recorded on the mobile device). Our approach is to build a hidden Markov model (HMM) to describe or model the temporal dynamics of the set of available wireless networks. Then we use a learning algorithm to learn the model parameters of this HMM from the recorded historical data, so that the HMM profiles the normal dynamics of the wireless availability. This HMM can then be used by

<sup>1</sup>The interval  $\delta$  can be made sufficiently small so that it does not bear an impact on the result.

another algorithm that does the inference task of *predicting* the posterior distributions in the future.

1) *Hidden Markov models*: A discrete (first-order) Markov chain or process is in one of a set of  $N$  states at any time step  $t$ . Let  $X_t$  denote the state at time  $t$ . The Markov process is governed by two assumptions: Markov and stationary. The Markov assumption is that the current state  $X_t$  depends only on the previous state  $X_{t-1}$ . The stationary assumption is that the laws of changes from state to state do not themselves change over time (even though the states change over time), i.e., the conditional probability  $P(X_t|X_{t-1})$  is independent of  $t$ . In summary, for all  $t$ ,  $P(X_t|X_{0..t-1}) = P(X_t|X_{t-1})$ , and the laws dictating how the state evolves over time are entirely contained within the conditional probability  $P(X_t|X_{t-1})$ .

In the hidden Markov model, the state of the Markov model is described by a single discrete random variable, whose possible values are the possible states of the world. Let  $O_t$  denote the observation or output that is seen at each time step  $t$ , which is dependent only on the state at  $t$ ,  $X_t$ . Each observation is affected (only) by the current state of the world — the state causes the observation or output to take on particular values — with the conditional probability distribution:  $P(O_t|X_t)$ . So the state of the world we are modeling is hidden from observation. We see only the  $O_t$ 's and we have knowledge of the conditional distribution  $P(O_t|X_t)$ .

An HMM is thus characterized by the following elements:

- The number of states,  $N$ ; i.e., there are  $N$  possible values for  $X_t$ .
- The number of distinct observation or output symbols,  $M$ ; i.e., there are  $M$  possible values for  $O_t$ .
- The state transition probability distribution given by the matrix  $T$ :  $T_{ij} = P(X_t = i|X_{t-1} = j)$ .
- The observation probability distribution for an observation  $k$  is the diagonal matrix,  $B^k$ :  $B_{ii}^k = P(O_t = k|X_t = i)$ ,  $1 \leq i \leq N$ ,  $1 \leq k \leq M$ .
- The initial probability distribution  $P(X_0)$  over the states at time 0, denoted  $P_0$ .

Let  $H = (T, B, P_0)$  denote an HMM with all its parameters.

2) *Prediction of future probability distributions of wireless availability*: In order to build an HMM to model the dynamics of the availability of different wireless networks over time, we must follow the steps:

- 1) Determine the set of states and the number of states.
- 2) Determine the set of observation or output symbols.
- 3) Learning of the HMM parameters — use training data to optimize the model parameters so the resulting HMM best describes the given observation sequence.

We now describe each step in detail. Motivated by the findings of the study in [9] that shows the high predictability of human movement,<sup>2</sup> as well as some data we collected

<sup>2</sup>Or at the very least, movement of mobile phone users.

from real mobile device users, we identified five distinct states of the user in terms of movement throughout the course of a 24-hour day: *commute*, *work*, *break*, *recreation*. We thus model the movement of a mobile device by a 5-state hidden Markov model (HMM) with these five states. The state transition model is shown in Figure 2. Where there are no arrows from state  $X$  to state  $Y$  in the diagram, it means that the transition probability  $P(X_t|Y_{t-1})$  is 0.

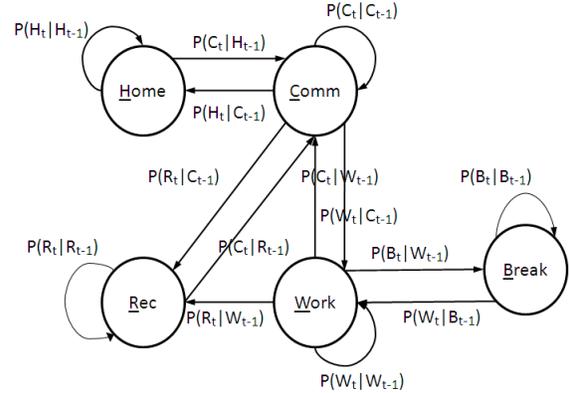


Figure 2. The state transition diagram of the HMM. Each state has a shorthand name that is its initial. Shorthand notation is used for the transition probabilities labeling the arrows between states:  $P(j_t|i_{t-1})$  means  $P(X_t = j|X_{t-1} = i)$ ; e.g.,  $P(W_t|C_{t-1}) = P(X_t = W|X_{t-1} = C)$ .

The *observation symbols or output* at time  $t$  is an  $n$ -bit binary string,  $b_1 \dots b_n$ , with  $b_i = 1$  if wireless network  $w_i$  is available and  $b_i = 0$  otherwise. Note that these observation symbols emitted by each state characterize that particular state, that is, they can be used to distinguish this state from other states. For instance, when the user is at home, the available wireless networks would most likely be the free home WiFi in addition to the GSM 2G or 3G. But when the user is in commute, it is more likely that only the 2G or 3G networks are accessible.

The next step is to learn the HMM from historic evidence (which is sometimes called the training data). For the initial distribution  $P_0$ , basically the belief of which state the world is in at the very beginning, we simply set it to the uniform distribution. The learning problem is essentially to determine the HMM model parameters  $(T, B)$  that maximizes the likelihood of the training data given the model. The training data is the observed available wireless networks over time. For the HMM learning, we use the well-known *Baum-Welch* algorithm that belongs to the family of Expectation Maximization algorithms. It is an iterative algorithm that starts with an initial HMM and in every iteration, computes a new HMM based on the old HMM from the previous iteration, using the given observation sequence. The algorithm always converges and finds local maxima.

Once the learning stage is done, the trained HMM now profiles normal dynamics of available wireless networks over time throughout a 24-hour day. Suppose the current

time is  $t$  (i.e., data  $d$  is created at  $t$ ), then the *prediction* task of finding the posterior distribution over the future state given all observed evidence up to date is to compute  $\mathbf{P}(X_{t+k}|O_{1:t}), k > 0$ . This can be done using the recursive computation:

$$\begin{aligned} \mathbf{P}(X_{t+k}|O_{1:t}) \\ = \sum_{X_{t+k}} \mathbf{P}(X_{t+k}|X_{t+k-1})P(X_{t+k-1}|O_{1:t}) \end{aligned}$$

The FORWARD algorithm [8] is used to compute this probability distribution.

With  $\mathbf{P}(X_{t+k}|O_{1:t})$ , we can easily find  $\mathbf{P}(O_{t+k}|X_{t+k})$  using the observation probability distribution  $B$  (this is one of the learned parameters of the HMM). Since each possible observation  $O_{t+k}$  is just an  $n$ -bit binary string whose  $i$ -th bit,  $O_{t+k}[i]$ , indicates the availability of  $w_i$ , we obtain the predictive inference we needed:

$$\begin{aligned} P(w_i \in W(t+k)) = \sum_{O_{t+k}} P(O_{t+k}) \cdot O_{t+k}[i], \\ i = 1, \dots, n \end{aligned}$$

#### IV. EVALUATION

We implemented our method of finding optimal mobile data upload time in Python and conducted experiments to evaluate it. We obtained some data of typical mobile device user’s movement patterns — in terms of the wireless networks that are available — throughout the day, over the course of 5 days. The set of all possible wireless networks  $W$  consists of: GSM-2G, 3G, free-WiFi (e.g., at home), and charging-WiFi (e.g., at Starbucks). The data transfer rates for these networks are set to the average bandwidth that one expects to receive from them. For the cost function, we simply map each  $w_i \in W$  to an integer that basically ranks them in terms of cost, ranging from 0 for free-WiFi to 3 for charging-WiFi. We then used this 5-day typical data as the blueprint to simulate the training data set for the learning of our hidden Markov model (HMM). The training data is generated by adding some noise to the data using a few Gaussian distributions of different variances with the blueprint data serving as the mean. We generated 100 days of training data. The time interval  $\delta$  used for discretizing time is 10 minutes.

We then introduced at random time steps, the creation of data items of sizes that are randomly selected from the range of 1 KB to 50 KB. Every data item is also randomly assigned one of three types that correspond to the three utility functions for upload time, shown in Fig. 1. For each data item, our algorithm for finding the optimal upload time  $t_u$  is executed. We also implemented the naive method for setting the upload time to be the earliest possible time for uploading, i.e., maximizing only the upload time utility. For our experiments, we did this for 1000 data items and compared the two methods using three measures of evaluation.

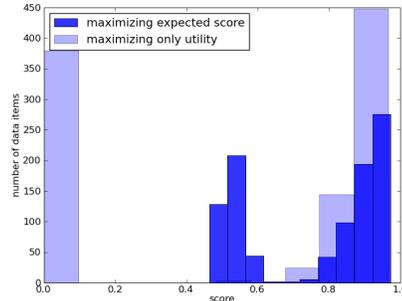


Figure 3. Histogram of score distribution for the upload times computed by score maximization and utility-only maximization for 1000 data items.

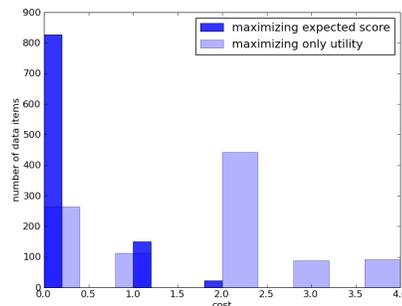


Figure 4. Histogram of cost distribution for the upload times computed by score maximization and utility-only maximization for 1000 data items.

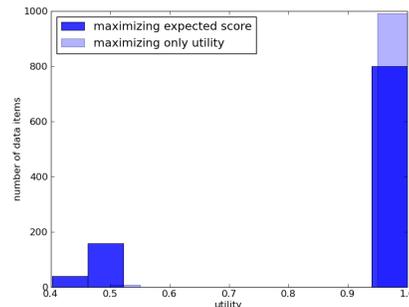


Figure 5. Histogram of utility distribution for the upload times computed by score maximization and utility-only maximization for 1000 data items.

In Figure 3, the scores of the upload times computed by our method of score maximization and of those by the method of maximizing only utility are compared. The histogram of the distribution of the scores over all the 1000 data items for both methods is shown. It is not surprising that since our method maximizes the score, it results in higher scores for the majority of data items. It is worth noting that for almost 40% of the data items, the upload times computed by the utility-only maximization result in 0 or near 0 scores. This is obviously a result of maximizing the utility only and not considering the cost of wireless networks used for uploading. The histogram of the distribution of costs for both methods are given in Figure 4, which clearly shows that score-maximizing is highly effective in minimizing cost, while utility-maximizing incurs costs that are 2 to 4 times as high. The utility measure is compared in Figure 5, which shows that score-maximizing matches the utility of the utility-maximizing method for the vast majority of the data items.

## V. CONCLUDE

We observed the common situation of mobile devices having access to multiple available wireless networks over time, and thereby identified the problem of selecting a "better" time to upload data created on the mobile devices. We quantify the "goodness" of an upload time by combining time *and* cost of uploading (which varies depending on the wireless network used) into a single utility function. An algorithm is proposed to choose an upload time by maximizing the utility, which chooses the earliest upload time while minimizing the cost. We also use a Hidden Markov model to infer the probabilities of the different wireless networks being available to the mobile device over time. Our simulation results confirm that our algorithm performs well in choosing better upload times than the naive method.

## REFERENCES

- [1] Web-dav linux file system (davfs2). <http://dav.sourceforge.net/>.
- [2] Ian F. Akyildiz and Wenye Wang. The predictive user mobility profile framework for wireless multimedia networks. *IEEE/ACM Trans. Netw.*, 12(6):1021–1035, 2004.
- [3] Petros Belimpasakis, Juha-Pekka Luoma, and Mihaly Börzsei. Content sharing middleware for mobile devices. In *MOBILEWARE '08: Proceedings of the 1st international conference on MOBILE Wireless MiddleWARE, Operating Systems, and Applications*, pages 1–8, ICST, Brussels, Belgium, Belgium, 2007. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).
- [4] P. J. Braam, M. Callahan, and P. Schwan. The intermezzo file system. In *in Proceedings of the O'Reilly Perl Conference*, 1999.
- [5] Peter J. Braam. The coda distributed file system. *Linux J.*, page 6.
- [6] B. et al. Pawlowski. The nfs version 4 protocol. In *in Proceedings of the 2nd International SANE Conference*, 2000.
- [7] A. Quintero. A user pattern learning strategy for managing users' mobility in umts networks. *IEEE Transactions on Mobile Computing*, 4(6), 2005.
- [8] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice-Hall, 2nd edition, 2003.
- [9] C. Song, Z. Qu, N. Blumm, and A. L. Barabasi. Limits of predictability in human mobility. *Science*, 327(5968):1018–1021, 2010.
- [10] J. Tolvanen, T. Suihko, J. Lipasti, and N. Asokan. Remote storage for mobile devices. In *in Proceedings of the First International Conference on Communication System Software and Middleware*, 2006.
- [11] C.H. Wu, H.P. Lin, and L.S. Lan. A new analytic framework for dynamic mobility management of pcs networks. *IEEE Transactions on Mobile Computing*, 1(3), 2002.