Proc. IEEE Conference on Global Communication, GLOBECOM'10, Miami, FL, Dec. 6-10, 2010

PCFG Based Synthetic Mobility Trace Generation

Sahin Cem Geyik, Eyuphan Bulut and Boleslaw K. Szymanski
Department of Computer Science and Center for Pervasive Computing and Networking
Rensselaer Polytechnic Institute
Troy, New York 12180
Email: {geyiks, bulute, szymansk}@cs.rpi.edu

***Abstract*—This paper introduces a novel method of generating mobility traces based on Probabilistic Context Free Grammars (PCFGs). A PCFG is a generalization of a context free grammar in which each production rule is augmented with a probability with which this production is applied during sentence generation. A concise PCFG can be inferred from the given real world trace collected from the actual mobile node behaviors. The resulting grammar can be used to generate sequences of arbitrary length mimicking the mobile node behavior. This is important when new protocol designs for mobile networks are tested by simulation.**

In the paper, we describe the methods developed to construct such grammars from training data (mobility history). We also discuss how to generate the synthetic data with an already constructed grammar. We present the experimental results on two real data sets, measuring similarity of the actual traces with the synthetic ones. We compare our grammar based method to a 2-level Markov Model based trace generation method. The results demonstrate that the grammar based approach works as an excellent compression method for the actual data. On many metrics, the synthetic data generated from the PCFG match the training data much better than the one generated by the Markov Model.

I. Introduction

Mobility of nodes is one of the key attributes of today's networks. It most often implies that nodes use wireless communications. Mobile ad hoc networks, delay tolerant networks, robotic networks and mobile sensor networks are all examples of such networks.

New protocols and algorithms for wireless mobile networks benefit from their verification via simulation in their early design stages. However, such simulations require large amount of realistic mobility behavior data, which are difficult to collect. Therefore, development of methods which can generate long synthetic mobility data from sample traces is crucial for proper evaluation of protocols and applications via simulation.

In this paper, we propose a novel trace generation method based on Probabilistic Context Free Grammars (PCFGs). Our method takes a real world trace as input, and automatically constructs a PCFG which concisely represents movement sequences of mobile nodes. Once a PCFG is constructed from a real world trace, a large set of sentences can be produced from it creating a synthetic mobility trace.

The rest of the paper is organized as follows. In the next section, we discuss the previous work on mobility trace generation. Later in Section III, we give the definition for PCFGs and the features that we added to them to capture the spatial and temporal aspects of mobile node movements. We also give the trace generation method, which is basically the production of sentences from the mobility PCFG. The following evaluation section compares our method with a 2-level Markov model based synthetic trace generation method on two separate datasets. The last section contains conclusions and an outline of future work on this topic.

II. Previous Work

There were many attempts at creating synthetic mobility patterns, ranging from methods based on connectivity graph [3], action profiles [2] to combining terrain and vehicle properties separately [4], to capturing group behavior [5], event-driven [9], [10], [11], and finally to extraction of information from real world traces [6], [7].

Another approach based on a time-variant community mobility model is proposed in [8]. Communities are defined based on popular locations, most often visited by nodes. The model collects two characteristics, skewed location visiting preferences and periodical re-appearance at the same location from real world WLAN traces, in order to produce mobility traces. Urban pedestrian flows (UPF) mobility scenarios are discussed in [12]. The system uses a set of pedestrian densities on streets as well as a set of likely paths that the pedestrians may follow and creates mobility information based on them. The trace generator aims at also keeping the observed pedestrian densities and the ones in the synthetic data as close as possible.

The works closest to ours utilize Markov Models. In [17], transitions between areas are modeled by their probabilities. Markov Model based mobility predictors are compared to LZ-based mobility predictors in [18] and the results show that Markov Models perform better. Interestingly, the paper also demonstrates that in practice, a 2-level Markov Model predictor performs better than a 3-level or 4-level predictor, hence increasing the depth does not necessarily increase prediction accuracy. Markov Models were extended by adding time information through cumulative time distribution of transitions in [19]. A 2-level Markov Model is used to predict connectivity and quality of connection to access points in a mobile network in [20].

III. Methodology

A. Mobility PCFGs

A Probabilistic Context Free Grammar [1] consists of a five-tuple <S_{nt},S_t,R_g,Prob,Start> where:

- Start is the initial nonterminal symbol of the grammar,

- $\mathrm{S_{nt}}$ is a list of nonterminal symbols defined by production rules,
- $\mathrm{S_t}$ is a list of terminal symbols which are the symbols actually seen in the sentences,
- $\mathrm{R_g}$ is a list of production rules that map a string of terminal and nonterminal symbols onto a nonterminal symbol,
- Prob is a list of probabilities, each assigned to a rule to define the probability that this rule (as opposed to the other rules forming the same nonterminal) is chosen in parsing or string generation.

To put it simply, a PCFG is an extension of the ordinary context free grammar in which the rules of each nonterminal are assigned probabilities of use (these probabilities sum up to 1.0 for each nonterminal). Probability of generating a string given a grammar G is the product of the probabilities at each branch of its parsing tree (if there is more than a single parsing tree, a summation over all parsing tree probabilities must be performed). A simple grammar that generates strings of the form a^n is given below.

$$Start \rightarrow a\ (0.6)\ |\ a\ Start\ (0.4)$$

For the above grammar, the string $a\ a$ has the probability $0.4\ x\ 0.6\ =\ 0.24$ which can also be seen as P($Start\ \rightarrow a\ Start\ |\ Start, G$) x P($Start \rightarrow a\ |\ Start, G$).

To capture spatial patterns of node movements, a PCFG can be built when mobility trace consists of terminal symbols representing the locations at which a mobile node can reside. The probabilities provided in the PCFG give us the likelihood for movement patterns. Another mobility information that can be represented by a PCFG is the meeting sequences for mobile nodes. In this case however, the terminals represent mobile nodes in the network.

To represent temporal information of node movements within a PCFG, we utilize a special time terminal symbol, t. It represents a preset time interval specific to the application domain. Hence, a mobility sequence of a node contains both location terminals as well as time terminals to represent the time interval between two consecutive location terminals. For example, the following trace of movements of a node:

$$l_A\ \ 40\ \ l_B\ \ 25\ \ l_C$$

states that once the node arrived at location l_A, it has taken 40 time units to move to the next location, l_B, and another 25 units to reach l_C. If the time token was chosen with time interval of 25 units, the above trace will be represented (approximately) by a sentence:

$$l_A\ \ t\ \ t\ \ l_B\ \ t\ \ l_C\ \ .$$

It should be noted that there is a trade-off between the time interval of the time token (resolution) and the complexity of the grammar, which is related to the length of the sentences in the training data. By introducing the notion of time terminals to PCFGs, we can store the temporal and spatial aspects of mobility patterns in a single sentence.

B. Automatic PCFG Construction

In our previous work [1], we have described in detail how a PCFG can be constructed given a set of sentences (hence strictly from the positive data). This algorithm was an extension of the works done in [14] and [15] with improvements on the time complexity. Although we will not go into details of the grammar inference algorithm in this paper, we will summarize its methodology.

Inference algorithm consists of two stages: (i) data incorporation, and (ii) application of operators. In the first stage, all sentences are introduced to the initial grammar as rules of the $START$ nonterminal and probabilities are assigned according to the sentence frequencies. Each terminal symbol (token) is introduced to the PCFG by a nonterminal symbol which has a single rule (that terminal symbol) with probability 1.

In the second state, the grammar is generalized and made more compact using two operators:

- **Chunking** that creates a new nonterminal which is assigned a string of nonterminals and which replaces all the occurrences of this string in other productions with that new nonterminal. Frequency of this nonterminal (hence its single rule) is set to the number of replacements made.
- **Merging** that creates a new nonterminal defined as a combination of two nonterminals. The right hand sides of productions of both nonterminals form the productions of this new nonterminal and probabilities are assigned according to their respective frequencies. The merged two nonterminals are removed and occurrences of any of these nonterminals are replaced by this new nonterminal.

Since grammar inference is a search for operands for two possible operations, an evaluation method is needed to measure the goodness of a grammar which results from an application of each possible operation. A Bayesian posterior probability of the grammar G given the data D is used for this purpose and it is defined as:

$$P(G|D) = \frac{P(G)P(D|G)}{P(D)}. \tag{1}$$

For maximization purposes, $P(D)$ can be omitted from the formula above. $P(G)$ is calculated by using $l(G)$ which is the length of grammar description. The simple description method proposed in [1] allows for restricting the search space for operand of a possible chunk operation to strings of length at most 5. $P(D|G)$ is calculated as the product of separate sentence probabilities (d_i) in the training data:

$$P(G) = 2^{-l(G)} \quad \text{and} \quad P(D|G) = \prod_{i=1}^{|D|} P(d_i|G).$$

Formulation of $P(D|G)$ as above helps the algorithm to avoid re-parsing after merging operation and also reduces the search space for the operands of merging operation. [1] establishes the time complexity of $O(D^2 log(D))$ for the algorithm where D is the size of the training data.

C. Synthetic Mobility Trace Generation with PCFGs

As aforementioned, synthetic trace generation is basically creating a sentence from the constructed grammar. This sentence gives both the temporal and spatial information for the single mobile node. Furthermore, once the generated sequence is completed (all the nonterminals in the sentence are replaced with terminals), a new sentence can be generated for the corresponding mobile node. Hence, we present a single algorithm here, which gets as input the mobility grammar and initial location of the mobile node (can be $null$ for a node that has just begun its journey), and creates a new sequence beginning in the initial location. Of course, the probabilities of the production rules are taken into account when deciding which rule to apply next in sentence generation process. The silent assumption here is that the input data contain traces starting at each location that is the ending location of some trace.

Algorithm 1 Method for creating a random route for a mobile node from the mobility PCFG given an initial location of this mobile node

```
$init\_loc =$initial location
$g =$mobility grammar
for each rule $r$ in $g.START$ do
    $string = r$
    for each expansion $string_i$ of $string$ with terminal at position 0 do
        if $string_i[0] == init\_loc$ then
            $list.add(string_i)$
        else
            $delete(string_i)$
        end if
    end for
end for
normalize probabilities in $list$
$random = rand()$
$progressive = 0$
for all expansions $e_i$ of every string in $list$ do
    $progressive += prob(e_i)$
    if $progressive \geq random$ then
        return $e_i$
    end if
end for
```

In Algorithm 1, the initial stage checks for all possible movement sequences (hence all possible sentences produced by the PCFG), and keeps only the ones in which the first terminal is the same as the initial location of the mobile node. In the case of modeling meetings of mobile nodes, the symbols are the mobile nodes met, hence although the algorithm stays the same, the meanings of the symbols produced or matched are different. After the initial elimination, the remaining productions are chosen according to a probability distribution. Please note that the sum of all productions before elimination (but not after it) is 1.0, so a normalization is done by multiplying accordingly all the branches of parsing tree of selected sentences.

IV. Evaluation of the Trace Generation Method

In this section, we are measuring similarity between real world traces and the synthetic mobility traces generated by the proposed method. We compared our PCFG based system to a 2-level Markov Model based generator presented in [19]. This is not a memoryless approach and it has been shown to work better than other methods for mobility prediction (see *Previous Work* section). Hence, intuitively, it is also a good model for capturing properties of the actual traces. Both PCFG and Markov Models hold more information than classical statistics based approaches. A PCFG holds a set of routes with probabilities assigned to them according to how frequently they are used. The main difference between a PCFG and a Markov Model is the fact that while a Markov Model keeps transitions at a preset length, a PCFG has the ability to extend the pattern lengths according to the training data. Furthermore, the automatic construction method given [1] provides generalization, hence unseen, but probable patterns are also added into mobility grammars of nodes which can not be achieved by Markov Models.

For the evaluations, we have used two datasets, the first one [16] contains bus-to-bus meeting data collected in Amherst, MA (DieselNet - Spring 2006). To train the PCFG for this dataset, we have taken sentences to be the set of buses met during one round of a bus on the route. Each bus type in this dataset has a set route, therefore we can artificially set a start and end point (we chose those as the busiest grids in terms of the number of meetings). Hence we created the synthetic data as a set of rounds. The second dataset we have experimented on is the cab mobility data collected in San Francisco, CA [13]. This data basically contains the taxi routes defined by latitude and longitude of taxi positions. Furthermore, we have also information if a customer is in the taxi or not. To account for this information, we split traces into two subsets: one with traces with a customer in the taxi and another without the customer. Furthermore, we divided the area into a grid of 25x25, for discretization purposes.

We used the following metrics in the comparison. For DieselNet Dataset, we have collected what buses are met by a bus on a given route right after a certain sequence of meetings. For example, the error rate *Cons 2* gives the difference of a given model from the actual trace in terms of the distributions of which buses are met after a certain single bus is met. Hence it can be taken as the distribution difference of meeting sequences of length 2. To calculate the difference, we used the euclidean distance between the sequence distributions. In other words, given that generated data have g_i percentage of meeting with bus b_i and the real world data have r_i percentage of meeting with bus b_i after a certain single bus, we calculate $\Delta_{Cons} = \sqrt{\sum_{i=1}^{k}(g_i - r_i)^2}$ (where k is the number of buses).

Another metric is based on the inter-meeting times, in which we calculate the time it takes for a bus to meet another bus given it has met a certain sequence of buses. *Intern 2* means the

TABLE I: Difference of Meeting Distributions between Actual Traces and PCFG vs Markov Model in DieselNet Dataset.

	Cons 2	Cons 3	Cons 4	Cons 5	Cons 6
PCFG	0.149	0.213	0.410	0.238	0.344
Markov Model	1.145	0.440	0.974	1.947	2.172

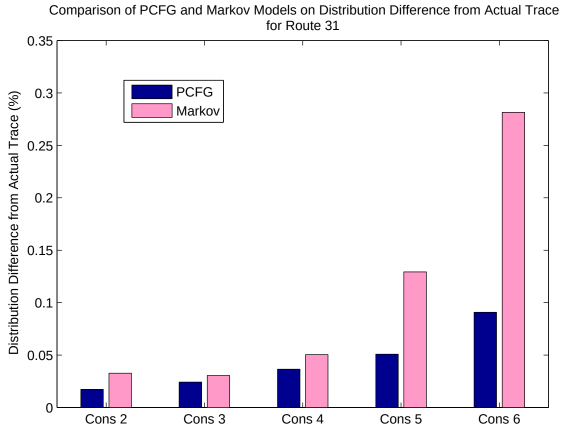

Fig. 1: Difference of Meeting Distributions between Actual Traces and PCFG vs Markov Model for Route 31 in DieselNet Dataset.

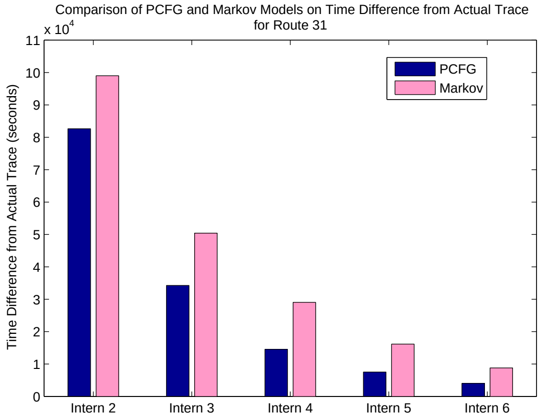

Fig. 2: Difference of Inter-meeting Time Distributions between Actual Traces and PCFG vs Markov Model for Route 31 in DieselNet Dataset.

time it takes to meet a second bus after a sequence of length one is met (the length is two for *Intern 3*, three for *Intern 4* etc.). Here, we used the **weighted** euclidean distance between the average intermeeting times for calculating errors. In other words, given that generated data have an average intermeeting time tg_i for bus b_i and the real world data have an average intermeeting time tr_i for bus b_i after meeting a certain bus, we calculate $\Delta_{Intern} = \sqrt{\sum_{i=1}^{k}(w_i \times (tg_i - tr_i))^2}$ (where k is the number of buses and w_i is the weight of bus b_i, calculated according to the frequency of meeting). For the taxi mobility dataset, we use the same metrics, however the buses are replaced with the location grids, hence *Cons 3* for the location distributions means the error on the distribution of three sequences of locations that a mobile node goes through.

Tables I and II give the overall results on DieselNet Dataset. They are averaged over eight routes (30,31,34,35,37,38,39,45) listed, and it can be seen that in all error categories, PCFG generates better traces than a 2-level Markov Model. We have already described how the generation with PCFG works, whereas a Markov Model creates third meeting given the previous two meetings of a given bus while generating the trace. We also provide the detailed results of an example route 31 in Figures 1 and 2 for illustration.

In Figures 3 and 4, we present the results on taxi mobility dataset. It can be seen that on all error categories, the synthetic data generated by the PCFG is closer to the actual trace than the synthetic data generated by the Markov Model.

The results demonstrate that real world traces are well mimicked by the sentences generated by the corresponding PCFG.

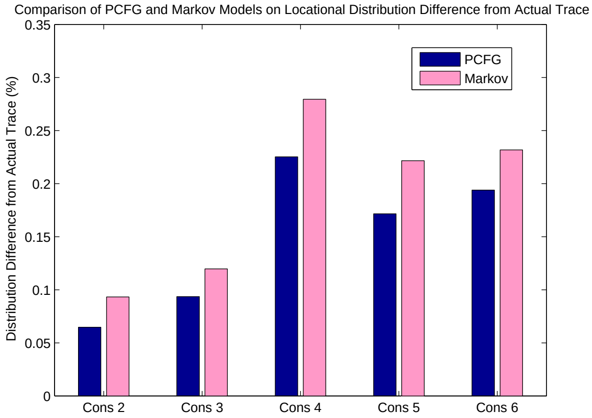

Fig. 3: Difference of Location Distributions between Actual Traces and PCFG vs Markov Model in Taxi Mobility Dataset.

V. Conclusion

In this paper, we address the problem of synthetic mobility trace generation. We propose the use of probabilistic context free grammars (PCFGs) which can represent a set of routes/patterns in a compact manner, and then generate similar routes/patterns of arbitrary length. We have shown how temporal information can be integrated into grammars through time tokens. After the description of mobility generation method, we have evaluated our model over two datasets (DieselNet and San Francisco Taxi Mobility Datasets), using metrics based on similarity to actual traces. We have shown that PCFGs generate synthetic traces that are much closer to the original ones than

TABLE II: Difference of Inter-meeting Time Distributions between Actual Traces and PCFG vs Markov Model in DieselNet Dataset.

	Cons 2	Cons 3	Cons 4	Cons 5	Cons 6
PCFG	693415.025	346170.555	178383.428	91175.888	57988.681
Markov Model	842365.695	496940.369	303851.526	174315.950	110041.562

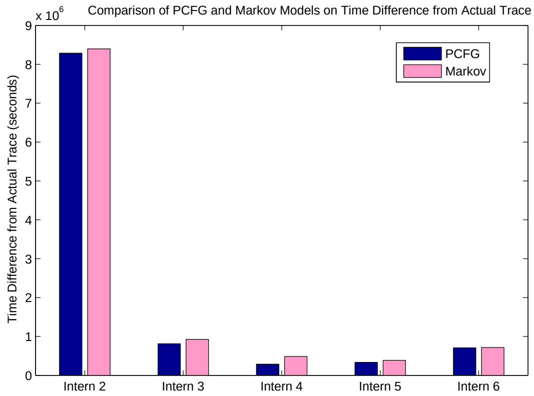

Fig. 4: Difference of Location Transition Time Distributions between Actual Traces and PCFG vs Markov Model in Taxi Mobility Dataset.

those produced by Markov Models, which are used as the state of the art mobility prediction method.

Our future work includes applying PCFG method to mobility prediction. In this paper, we have shown the capability of PCFGs to represent the mobility patterns of nodes in a mobile network. Such information can be utilized to predict the movements of mobile nodes, assuming efficient methods of processing PCFGs can be devised. We will work on these methods as well as other application domains where PCFGs can be utilized in a beneficial manner.

Acknowledgment

This research was sponsored by US Army Research laboratory and the UK Ministry of Defence and was accomplished under Agreement Number W911NF-06-3-0001 and under Cooperative Agreement Number W911NF-09-2-0053. The views and conclusions contained in this document are those of the authors, and should not be interpreted as representing the official policies, either expressed or implied, of the US Army Research Laboratory, the U.S. Government, the UK Ministry of Defense, or the UK Government. The US and UK Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

References

[1] Geyik, S. C., Szymanski, B., *Event Recognition in Sensor Networks by Means of Grammatical Inference*, IEEE INFOCOM 2009, Rio de Janeiro, Brazil, March 2009, pp. 900-908.

[2] Frangiadakis, N., Kyriakakos, M., Hadjiefthymiades, S. Merakos, L., *Realistic mobility pattern generator: design and application in path prediction algorithm evaluation*, Personal, Indoor and Mobile Radio Communications, 2002. The 13th IEEE International Symposium on, vol. 2, Page(s): 765-769.

[3] Calegari, R., Musolesi, M., Raimondi, F., Mascolo, C., *CTG: A Connectivity Trace Generator for Testing the Performance of Opportunistic Mobile Systems*, in Proceedings of the European Software Engineering Conference and the International ACM SIGSOFT Symposium on the Foundations of Software Engineering (ESEC/FSE07). Dubrovnik, Croatia.

[4] Karnadi, F. K., Mo, Z. H., Lan K., *Rapid Generation of Realistic Mobility Models for VANET*, IEEE Wireless Communications and Networking Conference. March 2007 Page(s):2506 - 2511.

[5] Tan, D.S., Zhou, S., Ho, J., Mehta, J., Tanabe, H., *Design and Evaluation of an Individually Simulated Mobility Model in Wireless Ad Hoc Networks*, Communication Networks and Distributed Systems Modeling and Simulation Conference, 2002.

[6] Kim, M., Kotz, D., Kim, S., *Extracting a Mobility Model from Real User Traces*, Proc. 25th IEEE International Conference on Computer Communications, INFOCOM April 2006, Page(s): 1-13.

[7] Tuduce, C., Gross, T., *A mobility model based on WLAN traces and its validation*, Proc. 24th Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM, March 2005, Page(s): 664-674.

[8] Hsu, W., Spyropoulos, T., Psounis, K., Helmy, A., *Modeling Time-Variant User Mobility in Wireless Mobile Networks*, Proc. 26th IEEE International Conference on Computer Communications, INFOCOM, May 2007, Page(s): 758-766.

[9] Chang, Y., Liao, H., *EMM: an event-driven mobility model for generating movements of large numbers of mobile nodes*, Simulation Modelling Practice and Theory, Volume 13, Issue 4, June 2005, Page(s): 335-355.

[10] Bhattacharjee, D., Rao, A., Shah C., Shah, M., Helmy, A., *Empirical modeling of campus-wide pedestrian mobility observations on the USC campus*, Vehicular Technology Conference, Sept. 2004, Page(s): 2887-2891.

[11] Stepanov, I., Hahner, J., Becker, C., Tian, J., Rothermel, K., *A meta-model and framework for user mobility in mobile networks*, 11th IEEE International Conference on Networks, ICON 2003, Page(s): 231-238.

[12] Maeda, K., Uchiyama, A., Umedu, T., Yamaguchi, H., Yasumoto, K., Higashino, T., *Urban pedestrian mobility for mobile wireless network simulation*, Ad Hoc Networks, vol. 7, iss. 1, January 2009, Page(s): 153-170.

[13] Piorkowski, M., Sarafijanovoc-Djukic, N., Grossglauser, M., *A Parsimonious Model of Mobile Partitioned Networks with Clustering*, The First International Conference on COMmunication Systems and NETworkS (COMSNETS), 2009, Bangalore, India.

[14] A. Stolcke (1994), *Bayesian Learning of Probabilistic Language Models*, Doctoral dissertation, Dept. of Electrical Engineering and Computer Science, University of California at Berkeley.

[15] S. F. Chen (1996), *Building Probabilistic Models for Natural Language*, Doctoral dissertation, Dept. of Computer Science, Harvard University.

[16] Zhang, X., Kurose, J., Levine, B., N., Towsley, D., Zhang, H., *Study of a bus-based disruption tolerant network: Mobility modeling and impact on routing*, In Proc. ACM Annual Intl. Conf. on Mobile Computing and Networking (Mobicom), Page(s): 195-206, 2007.

[17] Ashbrook, D., Starner, T., *Using GPS to Learn Significant Locations and Predict Movement Across Multiple Users*, Journal of Personal and Ubiquitous Computing, 7(5):275-286, October 2003.

[18] Song, L., Kotz, D., *Evaluating Location Predictors with Extensive Wi-Fi Mobility Data*, In Proc. INFOCOM 04, Page(s): 1414-1424, 2004.

[19] Song, L., Deshpande, U., Kozat, U. C., Kotz, D., Jain, R., *Predictability of WLAN Mobility and its Effects on Bandwidth Provisioning*, In Proc. INFOCOM 06, April 2006.

[20] Nicholson, A. J., Noble, B. D., *BreadCrumbs: Forecasting Mobile Connectivity*, In Proc. MobiCom 08, Page(s): 46-57, New York, NY, USA, 2008.