An Empirical Activity Model for WLAN Users

Caleb Phillips
Portland State University
Portland, Oregon 97201
Email: calebp@cs.pdx.edu

Suresh Singh
Portland State University
Portland, Oregon 97201
Email: singh@cs.pdx.edu

***Abstract*—Understanding *user behavior* in wireless environments is useful for a variety of reasons ranging from the design of better sleep algorithms for components of mobile devices to appropriately provisioning the wireless network itself to better serve the user. Our work goes in a different direction from prior work on WLAN modeling and attempts to undersand the *protocol independent* behavior of users by developing packet-level models for user activity using diverse training data. Additionally we validate the derived model using a stochastic similarity metric adapted from human control strategy modeling and present a novel way to compare traces using this metric.**

I. Introduction

Understanding *user behavior* in wireless environments is useful for a variety of reasons ranging from the design of better sleep algorithms for components of mobile devices such as laptops to appropriately provisioning the wireless network itself to better serve the user. Indeed, if we can *predict* how a user behaves, in terms of using the wireless network, then we can attempt ambitious system-level approaches for resource allocation within the wireless local-area-network (WLAN) as well as provide hints to the user device itself on when to power off which components.

Much work has been done to characterize wireless network usage at coarse time-resolution and in the aggregate [12]. More recently, there have been some attempts to study the fine-time scale characteristics of wireless network traffic and link-layer behaviour [19], [22] as well as solutions to the inherent difficulties of wireless data-collection at this resolution [6]. In some cases, these studies are exploratory [4], in others they treat specific extreme cases such as congestion [11] or interference [21], and some serve to define models for behavior which can then be re-used [10], [14]. It is this third category which most closely mirrors our approach.

A model for user activity is perhaps most applicable to the areas of QoS profiling and dynamic power-management. For this reason, literature in both areas have made attempts to integrate measurement-driven models. In [8] Irani et al. propose a model which assumes no a-priori information about user behavior and then performs "online learning". Our model could easily be adapted to such an application and then improved with learning. Similarly, [13] which presents an elegant trace-derived algorithm for power-management, might be improved further by a protocol-independent model for user-activity.

[0]This work was funded by NSF awards NeTS-WN:0722008 and ITR (Medium):0325014.

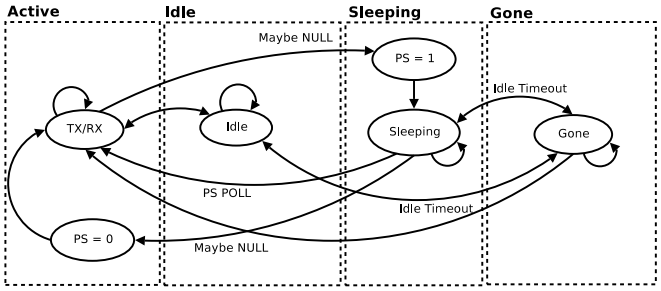

Fig. 1: Mapping of IEEE 802.11x actions to our 4-state activity model.

When deriving a model for any datum, a key task is validation. Many authors validate their model by showing improvement from a prior model. Others apply the model to a specific application and then demonstrate an improvement there. Because we have no prior model to compare to and would prefer to keep our model general so that it can be applied to many applications, we were forced to find a different approach. To this end, we have adapted a similarity measure used in human-control strategy modeling (a sub-field of robotics) to perform validation. To our knowledge, this is the first use of this method in networking, and we believe it holds promise for general-purpose model validation and also traffic comparison and characterization.

II. User Model

The behavior of a wireless user can be simply summarized with four states: (1) active, (2) idle, (3) sleeping, or (4) gone. The Active state means that the user is either receiving (Rx) or transmitting (Tx) packets. The idle state may best be thought of as a "thinking" state where the user is participating actively but is not transmitting or receiving. The sleeping state is one where the user device powers off to save energy. Finally, a user may enter the gone state if she leaves the coverage of an AP or powers off her device. Figure 1 illustrates this user model. Note that, with the exception of a transition from active to gone, which is impossible in this representation, the four-state digraph is fully connected. We chose this formulation due to its simplicity - the least complex formulation we could imagine which still captured the necessary attributes of a user's behaviour with respect to the wireless medium.

In order to compute the transition probabilities between states, we discretize the trace data by dividing time into intervals of length t seconds. Any packet transmission/reception by a user in an interval is treated as a transition to the active state whereas total lack of activity is treated as either a transition to idle (if the user was previously idle or active) or a transition to sleep (if the user was previously sleeping). Figure 1 illustrates additional transitions and states relating to power saving. These state transitions can be understood as follows. Idle users who have put themselves in power-saving mode by sending a (possibly NULL) frame with the PS bit set to 1 are labeled sleeping. Periodically sleeping clients will wake up to receive buffered packets from the access point (by issuing a PS-POLL frame), during this time, they are considered active, as they are receiving. When a client wishes to leave power-saving mode, they send a (possibly NULL) frame with the PS bit set to 0, and at this point they transition to either active or idle. Finally, users are considered "gone" if they are idle or sleeping with no activity for more than ten minutes.

For our binning interval, we've selected $t = 1$ seconds. When selecting a value for t, two things must be considered. Firstly, we would like to select a small enough value, that we are able to model user behavior at a level that is consistant with reaction times as reported by human computer interaction research [15] - on the order of hundreds of milliseconds to several seconds. Secondly, we would like to select a value big enough to avoid artifacts from 802.11 DCF contention. In other words, we would like to be sure that a user is idle because they have nothing to say, and not because they have a full buffer but are the loser in a contention window. Due to the possibility of channel capture, we can't be absolutely certain what a big enough interval is to allow everyone a chance to speak. With this in mind, we look to the literature for an empirical estimate at such a value. In [11], Jardosh et al. studied a congested network at the 62nd IETF Meeting in Minneapolis, MN. They found that the acceptance delay (the time between the first transmission and the eventual ACK including intermediate retries) has a worst-case upper-bound of 0.08s. In fact, the mean value is closer to 0.02s, with spikes up to 0.08 only for the largest frame sizes (greater than 1200 bytes) and during the most congested periods (when airtime utilization is nearly 100%). Given this worst case, we can fit 12.5 such delays within 1 second. Of our traces (see table I), the most contending active users we ever see is 8, in the "uw" trace, indicating that even in the worst case, we will never get to a point of total saturation. Given this, we believe that $t = 1$s is a good choice for the binning interval.

III. Deriving User Model Parameters

Table I summarizes the traces we used in this work. Four of the five traces were collected at locations around Portland, Oregon in the summer of 2006. Our initial attempt at characterization of the interesting features of these traces is in [19]. The fifth trace, "uw", was collected by University of Washington researchers at the SIGCOMM 2004 conference and is characterized in [16], [22]. All of the traces were collected passively, using vicinity sniffing techniques and are available at [2]. Our traces were collected using the VeriWave WT20 appliance [1], and the UW traces were captured using commodity hardware. In sum, these captures provide 131 usable[1] user-traces which can be used to derive user models.

Trace Name	Duration	Location Type	Distinct Users
uw	11.5 hours	Conference	42
cafe	4 hours	Cafeteria	23
library	4 hours	Library	31
powells	4 hours	Bookstore	25
ug	3 hours	Coffee shop	11

TABLE I: Summary of Training Data

A. Data Mapping

Real wireless traces are necessarily both noisy and incomplete [9], and as such a principle concern is weeding out erroneous information. Additionally, the majority of our traces were collected with hardware at a receive sensitivity deficit [19]. Because of this, it isn't acceptable to use the 802.11 FCS alone to identify erroneous frames, since frames with a bad FCS in our traces may have been received correctly by the AP (or client). In light of this we used a set of heuristics and filters to classify client MAC addresses and identify those that were not consistant. We start by making a "clean" copy of the trace by eliminating all frames with a bad 802.11 FCS, bad IP header checksum, and which are malformed (i.e. packet dissection fails). We use this "clean trace" to generate some statistics about the remaining users and then do some final scrubbing: if a given MAC address is responsible for less than 10 IP packets and is responsible for less than 40% of all traffic originating from its IP, we consider it *bogus* and discard all traffic from it. This fairly conservative method was quite successful for all of our traces.

We next extract traces for each of the 131 users that consist of packet transmit/receive events and PS events. We then split each user trace into two – one in which we only have Rx events and one in which we only have Tx events. The reason we did this is to determine whether (1) there is any correlation between Rx and Tx events and (2) if the user models thus derived differ substantially (e.g., if some users are predominantly receivers whereas others are active in transmitting as well as receiving). Finally, we split the users again by those who use Power-saving (PS) mode (*sleepy* users) and those who do not (*sleepless* users). The rationalle here is that those who periodically go to sleep are expected to display a significantly altered behaviour (one dictated by the power-saving algorithm and not by the users' behavior).

B. Derived User Models

In Table II, we provide the 4x4 transition probability matrices for Rx and Tx in both the sleepy and sleepless categories. Even by casual inspection, we can see a commonality between the various state-transition models. Indeed, if we examine

[1]One trace with zero data frames was omitted.

	active	idle	sleeping	gone
active	0.8487	0.1508	0	0
idle	0.0468	0.9526	0	0.0005
sleeping	0	0	0	0
gone	0.0009	0	0	0.9994

(a) All training data combined, sleepless category, Tx

	active	idle	sleeping	gone
active	0.6707	0.1636	0.1652	0
idle	0.0590	0.9064	0.0343	0.0002
sleeping	0.0885	0.0459	0.8651	0.0003
gone	0.0007	0.0001	0.0004	0.9994

(b) All training data combined, sleepy category, Tx

	active	idle	sleeping	gone
active	0.7786	0.2211	0	0
idle	0.0315	0.9678	0	0.0004
sleeping	0	0	0	0
gone	0.0001	0.0009	0	0.9994

(c) All training data combined, sleepless category, Rx

	active	idle	sleeping	gone
active	0.7254	0.1382	0.1363	0
idle	0.0443	0.9135	0.0417	0.0002
sleeping	0.0648	0.0589	0.8757	0.0003
gone	0.0003	0.0005	0.0004	0.9994

(d) All training data combined, sleepy category, Rx

TABLE II: Model Transition Matrices.

the Rx and Tx matrices for any case, we see that they are very similar (e.g., (a), (c) for sleepless and (b), (d) for sleepy with all users combined). In order to quantify this similarity we perform the following data analysis. First, we associate a numeric value to each of the four states: active – 1, idle – 2, sleeping – 3, gone – 4. Then, for each user, we consider the Rx and the Tx traces and obtain two vectors containing numbers from the set $S = \{1, 2, 3, 4\}$. We compute a correlation coefficient of these two vectors. The mean correlation between Rx and Tx states averaged over all 131 users is 0.7609 (median 0.9034). Similarly, if we construct a vector of number of packets received every $t = 1$s and another for packets transmitted for each user and determine a correlation coefficient for these vectors, we obtain a mean value of 0.6357 (median is 0.7760) – the correlation is very high in almost all cases. This implies that a user's behavior *at the level of abstraction of our model* is application independent since users who are predominantly receiving data are similar to those who are not. This is a very useful feature of our model since it allows a high-level compact representation of users.

Another feature worth noting is that users have a high-probability of staying in the state they are already in. The probability of staying in the same state (i.e. $\bar{P}_k = Pr[q_{i+1} = S_k | q_i = S_k]$) averaged accross all categories is $\{\bar{P}_1, \bar{P}_2, \bar{P}_3, \bar{P}_4\} = \{0.471, 0.838, 0.305, 0.443\}$ for Tx and $\{0.386, 0.863, 0.301, 0.382\}$ for Rx. This behaviour is easily translated to the concept of "thinking" time, used in internet traffic modeling [7]. Users are typically active for a couple of seconds, and then idle for several seconds, and then active for a couple of seconds and so on. This pattern is especially interesting as current powersaving schemes are agnostic to such behavior (as discussed in [13] and others).

C. Modeling Residing Time

Given transition probabilities as in Table II, we can generate traces for synthetic users through simple Markov simulation. However, in order to know *how long* to make the traces, we also need a robust model for residing time. We used the data at [2], collected over several years at Dartmouth, to fit a model to residing time, extending the work of [3], [4], [14]. The result, which is well fitted with a generlized pareto function, is in table III.

Censored Portion (s)	Distribution	Parameters	Goodness
$t = 0$	Gen. Pareto	$k = 3.5663$ $\sigma = 14.1917$ $\theta = 0$	$\lambda^2 = -0.0148$ $KS = 0.9934$ $\chi^2 = 6879700$
$60 \geq t \geq 86400$	Gen. Pareto	$k = 1.3$ $\sigma = 1121.9$ $\theta = 0$	$\lambda^2 = -0.0186$ $KS = 0.7920$ $\chi^2 = 3944700$

TABLE III: Residing-Time Model Parameters and Statistics.

IV. Validation of the Derived User Models

A. Methodology

In order to validate our model, *we must show that it successfully describes the data from which it is derived*. Here we take the classic approach of randomly halving the data into a training set and a test set. The training set is used to prepare a model, and then this model is validated against the test set. Our full data-set contains 131 users from the 5 site traces. Hence, we randomly select 66 of these to train from, and 65 to test with. The trained model is utilized to produce 65 synthetic user traces which can be compared to the test set.

We have adopted a stochastic method of model validation introduced in dynamic human control strategy modeling [18]. This method of model validation utilizes hidden markov model optimization via the Baum-Welch expectation-maximization (EM) algorithm.

Given some observation sequence O_i of T_i symbols, there must be a model $\lambda_i = \{A_i, B_i, \pi_i\}$ which optimizes $P(\lambda_i|O_i)$, the probability of the model given the observations. Each hidden Markov model (λ) is defined by a state transition matrix (A), a set of probabilties for each output symbol and each state (B), and a set of initial probabilities (π). In our model, state i always outputs the symbol i, so we set $B = I$ (the identity matrix).

The Baum-Welch algorithm works by iteratively maximizing $P(\lambda_i|O_i)$. At some point, the probability will cease to improve, and since Baum et al. have shown that $P(\lambda_i|O_i)$ is necessarily monotonically increasing with successive iterations, when $P(\lambda_i|O_i)$ ceases to improve, the model must be locally optimized [5].

Nechyba et al. utilize this final probability value to define a similarity function:

$$\sigma(O_i, O_j) = \sqrt{\frac{P_{21}P_{12}}{P_{11}P_{22}}} \quad (1)$$

Where

$$P_{ij} = \hat{P}(O_i|\lambda_j) = P(O_i|\lambda_j)^{1/T_i} \quad (2)$$

As is shown in [18], this metric is quite well-behaved as it exhibits the following useful properties:

$$\sigma(O_i, O_j) = \sigma(O_j, O_i) \quad (3)$$

$$0 < \sigma(O_i, O_j) <= 1 \quad (4)$$

$$\sigma(O_i, O_j) = 1 \textit{ if } \lambda_i \sim \lambda_j \textit{ or } O_i = O_j \quad (5)$$

To implement this metric, we made use of Kevin Murphy's hidden Markov model toolbox for Matlab [17] which contains an implementation of the Baum-Welch algorithm[2]. Murphy's toolbox implements the Baum-Welch algorithm with a few modifications suggested in [20]. These changes include the addition of scaling, to avoid underruns for small probability values, and a method for optimizing the model over several observation sequences. This second modification is essential for our ability to compare groups of user traces. Another practical problem which must be dealt with, is that $P(O_i|\lambda_j)$ is often out of the dynamic range of the machine running the algorithm. The solution is to use $LL_{ij} = log(P(O_i|\lambda_j))$ as the parameter to optimize. This is problematic for our similarity algorithm, as we need $P(O_i|\lambda_j)$ itself, but must also avoid underruns. The solution we used is as follows:

$$LL_{ij} = log(P(O_i|\lambda_j)) \Rightarrow 10^{LL_{ij}} = P(O_i|\lambda_j) \quad (6)$$

We can then make this substitution in 2 as follows:

$$P_{ij} = (10^{LL_{ij}})^{1/T_i} = 10^{LL_{ij}/T_i} \quad (7)$$

It is worth noting that initial parameter estimates are selected at random (as suggested by [20]) and then optimized. We use the obvious choice of 4 hidden states, but also tried many experiments with 8 states as well (the most used in [18]) to see if there was notable variance – there was not.

B. Results

In order to validate our derived user model, we follow the following procedure:

1) Of the 131 user traces for Tx (the same thing is done for Rx traces as well) we randomly select 65 user traces and call them the *test set*.
2) The remaining 66 user traces are called the *training set* and are used to train the model:
   a) First categorize the 66 users into sleepy and sleepless sets.
   b) For each set, concatenate the user traces and derive the 4x4 transition matrices as well as the steady-state probabilities.
3) Generate 65 synthetic user traces using the above transition matrices in the *same proportion* of sleepy and sleepless users. The length of each trace is a random variable selected from the residing time distribution of Table III and the starting state is chosen using the steady state probabilities.
4) We now run the similarity metric to compare the 65 *test set* traces with the 65 synthetic user traces. The process followed is:
   a) Each user trace (in each set of 65) is treated as an independent observation sequence O_i.
   b) We run the Baum-Welch EM Algorithm on each set of 65 to derive two different models (referred to as λ above), one for the synthetic set and one for the test set.
   c) We now compute the value of σ (equation 1) for these two models.

We run steps 3 and 4 repeatedly (note that in step 3 we probabilistically generate one set of 65 synthetic users). The average σ value we obtain is 0.8571 for Tx and 0.9164 for Rx after 10 repititions. Figure 2 plots the CDFs of these 10 σ values. As we can see, there is a high similarity between the derived model and the actual traces for both Rx and Tx (although Rx is generally more similar) which lets us conclude that the user model derived in step 2 is representative of the user population.

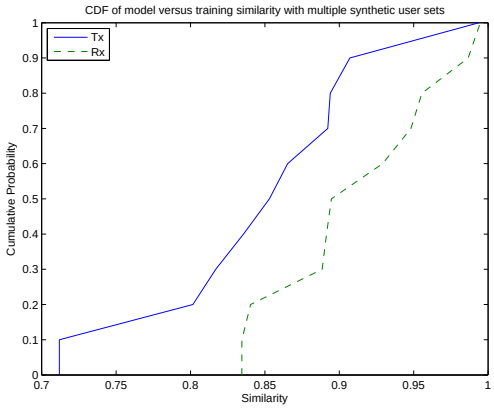

Fig. 2: CDFs of similarity values for multiple sets of synthetic users as compared to the training data.

V. Comparison of Different Traces With Each Other

Over the past several years, various researchers have collected WLAN usage data from a wide variety of domains and these traces have been analyzed either individually or in some combinations. While these forms of analysis and comparison are valuable, there is still the larger question of *classification*.

[2]Our implementation can be downloaded at http://web.cecs.pdx.edu/~singh/software.html

	uw	non-uw	cafe	library	powells	ug
uw	1.0000	0.9660	0.9352	0.9503	0.3798	0.3357
non-uw	0.9660	1.0000	0.9231	0.9747	0.8494	0.8857
cafe	0.9352	0.9231	1.0000	0.8820	0.7066	0.6452
library	0.9503	0.9747	0.8820	1.0000	0.7578	0.8830
powells	0.3798	0.8494	0.7066	0.7578	1.0000	0.8960
ug	0.7603	0.9803	0.7192	0.9904	0.8933	1.0000

(a) Tx Similarity

	uw	non-uw	cafe	library	powells	ug
uw	1.0000	0.6764	0.9971	0.9131	0.2121	0.7005
non-uw	0.6764	1.0000	0.9352	0.7701	0.8585	0.9594
cafe	0.9971	0.9352	1.0000	0.9951	0.6813	0.8201
library	0.9131	0.7701	0.9951	1.0000	0.4119	0.4795
powells	0.2121	0.8585	0.6813	0.4119	1.0000	0.7864
ug	0.7005	0.9594	0.8201	0.4795	0.7864	1.0000

(b) Rx Similarity

TABLE IV: Similarity between traces. The most similar and disimilar traces traces have been highlighted.

Classification is required to tame complexity of analysis as the database of traces available grows.

We utilize the technique discussed in the previous section to classify our studied traces. The question we ask is, are the five traces we study in this paper similar or distinct? and by what degree are they similar or distinct? To answer these questions, we compute σ values for each pair of traces. The results are presented in Table IV, where the item "non-uw" consists of a concatenation of all the traces except the uw trace (we did this because the uw trace was taken at a conference and generally exhibited high load – very different from the remaining four traces).

The main conclusions we can draw from the data are that the two traces most unlike one another are the powells and the uw trace. However, the uw trace is similar to the other three traces and almost identical to the cafe trace (0.9352 for Tx and 0.9971 for Rx). Considering that the uw trace was from SIGCOMM 2004 where the load was consistently high and the cafe trace was taken at a lightly loaded coffee shop, it is remarkable that the user models are so similar. The overall conclusion we can draw is that one or at most two user models are representative of all five traces giving us a very compact representation of all this data.

VI. Conclusion

In this paper we developed a wireless user model based on five different traces. The major conclusions we draw are:

1) By and large, the user models are similar across all five traces even though the traces were collected at different venues (library, coffee shops, conference). This implies that one or a small set of user models can be used to describe most WLAN users.
2) We know that some users are mainly receivers (e.g., web users) whereas others transmit as well as receive equally (e.g., p2p applications). However, we note that at the level of abstraction at which we construct the user models, all these users are similar. This is an interesting finding because our models are independent of the underlying applications.
3) The technique we use for model validation as well as comparison can be generally applied to compare and classify other traces as well.

References

[1] "Veriwave wt20, http://www.veriwave.com/," July 2006.

[2] "Archive for wireless data and papers, http://crawdad.cs.dartmouth.edu," May 2007.

[3] A. Balachandran, G. V. abd P. Bahl, and P. V. Rangan, "Characterizing user behavior and network performance in a public wireless lan," in *ACM SIGMETRICS*, June 2002, pp. 195 – 205.

[4] M. Balazinska and P. Castro, "Characterizing mobility and network usage in a corporate wireless local-area network," in *MOBISYS*, May 2003.

[5] L. E. Baum, T. Petrie, G. Soules, and N. Weiss, "A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains," *The Annals of Mathematical Statistics*, vol. 41, no. 1, pp. 164–171, 1970.

[6] Y.-C. Cheng, J. Bellardo, and P. Benk, "Jigsaw: Solving the puzzle of enterprise 802.11 analysis," in *ACM SIGCOMM*, September 2006.

[7] M. E. Crovella and A. Bestavros, "Self-similarity in world wide web traffic: Evidence and possible causes," *IEEE/ACM Transactions on Networking*, vol. 5, no. 6, pp. 835 – 846, December 1997.

[8] S. Irani, S. Shukla, and R. Gupta, "Online strategies for dynamic power management in systems with multiple power-saving states," *ACM Transactions on Embedded Computing Systems*, vol. 2, no. 3, pp. 325 – 346, August 2003.

[9] A. A. J. Yeo, M. Youssef, "A framework for wireless lan monitoring and its applications," in *WISE*, Oct 2004.

[10] R. Jain, D. Lelescu, and M. Balakrishnan, "Model t: An empirical model for user registration patterns in a campus wireless lan," in *ACM MOBICOM*, 2005, pp. 170 – 184.

[11] A. P. Jardosh, K. N. Ramachandran, K. C. Almeroth, and E. M. Belding-Royer, "Understanding congestion in ieee 802.11b wireless networks," in *Proceedings of the Internet Measurement Conference 2005 on Internet Measurement Conference*, Berkeley, CA, 2005.

[12] D. Kotz and K. Essien, "Analysis of a campus-wide wireless network," *Wireless Networks*, vol. 11, pp. 115–133, 2005.

[13] R. Krashinsky and H. Balakrishnan, "Minimizing energy for wireless web access with bounded slowdown," in *ACM MOBICOM*, September 2002.

[14] J.-K. Lee and J. C. Hou, "Modeling steady-state and transient behaviors of user mobility: Formulation, analysis, and application," in *ACM MobiHoc*, May 2006.

[15] I. S. MacKenzie, *HCI models, theories, and frameworks: Toward a multidisciplinary science*, J. M. Carroll, Ed. San Francisco: Morgan Kaufmann, 2003.

[16] R. Mahajan, M. Rodrig, D. Wetherall, and J. Zahorjan, "Analyzing the mac-level behavior of wireless networks in the wild," in *ACM SIGCOMM*, Sept 11 – 15 2006.

[17] K. Murphy, "Hidden markov model toolbox for matlab, http://www.cs.ubc.ca/ murphyk/software/hmm/hmm.html," May 2007.

[18] M. C. Nechyba and Y. Xu, "Stochastic similarity for validating human control strategy models," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 3, pp. 437 – 451, June 1998.

[19] C. Phillips and S. Singh, "Analysis of wlan traffic in the wild," in *IFIP NETWORKING*, 2007, pp. 1173 – 1178.

[20] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," vol. 77, February 1989, pp. 257 – 286.

[21] C. Reis, R. Mahajan, M. Rodrig, D. Wetherall, and J. Zahorian, "Measurement-based models of delivery and inteference in static wireless networks," in *ACM SIGCOMM*, September 2006.

[22] M. Rodrig, C. Reis, R. Mahajan, D. Wetherall, and J. Zahorian, "Measurement-based characterization of 802.11 in a hotspot setting," in *Proceedings of the ACM SIGCOMM 2005 Workshop on experimental approaches to wireless network design and analysis (E-WIND-05)*, 2005.