Journal of Computer and System Sciences 77 (2011) 869–883

Contents lists available at ScienceDirect

Journal of Computer and System Sciences

www.elsevier.com/locate/jcss

An empirical framework for user mobility models: Refining and modeling user registration patterns

Jinpyo Hong, Hwangnam Kim *

School of Electrical Engineering, Korea University, Seoul, Republic of Korea

ARTICLE INFO

Article history:
Received 20 January 2010
Received in revised form 23 February 2010
Accepted 9 August 2010
Available online 17 August 2010

Keywords:
WLAN
802.11
Empirical mobility model
Network performance

ABSTRACT

In this paper, we examine user registration patterns in empirical WLAN traces, identify elusive patterns that are abused as user movements in constructing empirical mobility models, and analyze them to build up a realistic user mobility model. The examination shows that about 38–90% of transitions are irrelevant to actual user movements. In order to refine the elusive movements, we investigate the geographical relationships among APs and propose a filtering framework for removing them from the trace data. We then analyze the impact of the *false-positive* movements on an empirical mobility model. The numerical results indicate that the proposed framework improves the fidelity of the empirical mobility model. Finally, we devise an analytical model for characterizing realistic user movements, based on the analysis on the elusive user registration patterns, which emulates elusive user registration patterns and generates true user mobile patterns.

1. Introduction

As portable devices become popular, such as Personal Digital Assistants (PDAs), Portable Media Players (PMPs), and MP3 player, and many enhanced service features incorporated into the devices are available through Wi–Fi interfaces, such as real-time video streaming, network games, and web browsing, Wi–Fi Hot Spots gain more responsibility for connecting nomadic people to the Internet. On the other hand, many wireless applications use location information to carry out hand-offs, provide location-based services, or improve the quality of services. However, since IEEE 802.11-based Wi–Fi Hot Spots do not fully support user mobility and cannot provide location information for users, mobile users may experience inferior connectivity due to inappropriate hand-off and service unavailability due to inaccurate paging (or positioning). Hence, we need to predict user mobile patterns, at least the association patterns of users or their wireless devices, since the accurately predicted mobile scenario assists us to precisely pinpoint the current location of the user, and it also directs us to carry out seamless hand-off procedure, so that the prediction consequently impacts on the connectivity and the performance of wireless networks.

At the initial stage of the research when no empirical trace data was available, theoretical models have been developed, many of which relied on random theory such as random walk, random waypoint and random direction model, assuming the users decide their next move in a random manner. But since random theory does not fit well in reality, other supplementary theoretical models such as Gauss–Markov model or Obstacle mobility model have emerged as well. However, once Dartmouth College released a large set of WLAN trace data to the wireless community [1], empirical trace-driven mobility models have been developed, such as in [2–5], and also several empirical trace sets have been available to develop more

* Corresponding author.
E-mail addresses: jpsps@korea.ac.kr (J. Hong), hnkim@korea.ac.kr (H. Kim).

0022-0000/$ – see front matter
doi:10.1016/j.jcss.2010.08.005

realistic mobility model for WLAN users. In comparison with the theoretical models, the empirical mobility models are superior in capturing the real aspects of user movements. For example, the models in [6] and [2] discovered that users are most likely to stay in only one access point (AP) while connecting to the Internet, and even if they roam around in the network, their associations are usually confined within a geographically close area.

In this paper, we firstly present that these WLAN empirical traces include elusive user registration patterns that degenerates the fidelity of mobility model, and then build a filtering framework that can effectively remove false user registrations (re-associations). Note that the *false-positive* user registrations result from the dense deployment of APs, and so, as the density of APs increases up to a certain level, the number of elusive registration increases and consequently the quality of empirical models proportionally degrades. In order to remove the *false-positive* registrations, we suggest to take into account the spatial relationships among APs in dealing with the real trace, especially in the dense WLANs. The spatial relationship actually verifies if the transition between any pair of two APs is geographically possible when it appears in the trace. The investigation shows that about 90% of the Dartmouth College trace data and 38% of the UCSD trace data [7] transitions are irrelevant with actual user movements. The difference between the two traces comes from the method of generating the traces: the UCSD trace was a polling-based trace while the Dartmouth College trace was an event-based one, so that the former produced a relatively low proportion of irrelevant transitions. Based on this observation, we propose a filtering framework for extracting true user registration patterns from the trace data. In the framework, we classify the re-associations into *positive* and *false-positive* registrations and use only *positive* data in building up an empirical mobility model, where *positive* re-associations are considered as real user movements and *false-positive* re-associations are considered as transitions between APs caused by external effect (such as signal fluctuation) rather than actual user movements. We use a large collection of user registration patterns gathered from the Dartmouth College campus network collected from April 2001 to July 2004 [8] to validate the effectiveness of the proposed framework. Then we analyze the impact of the *false-positive* registrations on an empirical model. For the evaluation, we use Model T [2]. The numerical results indicate that the *refined* trace data produces a more accurate empirical model in perspective of statistical accuracy. Based on the examination and analysis on user registration patterns, we develop a simple empirical user mobility model for estimating the future mobile or association patterns of the users. The proposed model characterizes the effect of *false-positive* registrations in addition to *positive* registrations on realistic registration scenario of the users. Especially, the emulation for *false-positive* registration is important because the dense WLAN environment inevitably creates such *false-positive* transitions and continues to last for a quite long (or whole) period of time. In order to build up realistic empirical model, we utilize the SNMP log from Dartmouth College collected from November 2003 to February 2004 [9], from which we observed that fluctuating channel state creates the *false-positive* transitions and then extracted topological features of the transitions when these transitions happen.

The rest of the paper is organized as follows. We briefly discuss about existing empirical mobility models in Section 2. In Section 3, we introduce the Dartmouth trace data and explain the proposed framework of refining the trace data. We then examine the impact of *false-positive* registration on actual mobility model in Section 4, where we also use the UCSD trace data in addition to the Dartmouth trace data. Next, we propose a simple analytical model for modeling user association patterns and evaluate the model based on prediction rate in Section 5. Finally, we conclude the paper in Section 6.

2. Related work

In this section, we will briefly introduce several trace data, empirical mobility models, and the previous work that have been developed to extract *positive* registration patterns from the real trace.

Trace data. The Dartmouth College trace data [1] consists of association, re-association, and disassociation *events* of mobile users collected from distinct APs in the campus-wide WLANs. Trace data in MIT [10] and UCSD [7] used *polling* method to record user associations at every periodic interval. Although there is no explicit work to present that event-based trace is better than the polling-based trace, we use the Dartmouth college data for building up the filtering framework since we do not want to miss any short timescale movement which may not be captured in the interval used in the polling-based method. But we do use the UCSD trace data to verify the effect of the proposed filtering framework. Recently, Rhee et al. have released human mobility track logs using GPS from 44 participants from five different sites [5]. In vehicular networks, there is a publicly available data set of GPS traces collected by 500 taxi cabs in San Francisco for over two years [11].

Mobility models. The Model T [2] is an empirical model that can generate a synthetic trace with statistic information based on the Dartmouth College trace data (from April 2001 to March 2003). In the model, the whole set of APs is clustered using MCL (Markov Clustering) graph clustering algorithm [12] according to the number of transitions between APs, showing a distinct hierarchy in user registration patterns. Its subsequent work in [13] extends the model by including the transitions to and from OFF states of wireless devices, and it also provides a method for determining the residence time at each APs based on its *popularity gradient*, which is determined by the popularity of previously and currently associated APs. Lee et al. [3] characterized steady-state and transient behaviors of user mobility, and represented the mobility by a semi-Markov process. It mainly investigates how user mobility is correlated in time at different time scales. Yoon et al. [4] built up a realistic mobility model by combining association data (between WLAN users and APs) with an actual map of Dartmouth College

campus, where the map is converted into a graph representation. Kim et al. [14] introduced a predictive association control algorithm based on past mobility patterns. In addition, Lee et al. [15] produces synthetic human walk traces by expressing the patterns that involves (i) truncated power-law distributions of flights, pause times and inter-contact times, (ii) fractal waypoints and (iii) heterogeneously defined areas of individual mobility.

Optimizing registration patterns. The work presented in [16,17] identified *ping-pong* phenomenon in the trace data. The phenomenon occurs when a WLAN device is located within multiple transmission ranges of APs and the unreliable wireless channel continuously triggers the re-association procedure among them even if there is no real user movements. The work in [3] and [4] proposed detailed methods for removing the transitions occurred due to *ping-pong* phenomenon. Lee et al. [3] devised a method for eliminating them in the way that all the APs involved to a *ping-pong* phenomenon are replaced with dominant APs. In the method, transition is classified as a ping-pong transition if the sequence of transitions form a round trip between two or three APs. Yoon et al. [4] filtered out the *ping-pong* re-association with an average moving filter, which produces similar results to those obtained in [3]. Even though both works deal with *ping-pong* phenomenon to construct better mobility models, they do not verify if the transitions in a trace are really possible in perspective of geographical relationships among AP locations and/or human mobile speed.

3. Filtering framework

In this section, we first explain the empirical data from Dartmouth College to give a better understanding about the proposed framework. We then explain how to devise the framework for filtering out *false-positive* user registration patterns.

3.1. Empirical trace data

We use the trace data that the Dartmouth College collected from April 2001 through July 2004 [8]. The data covers 602 APs in 161 buildings and there are 13 888 different WLAN users in total. The users can be divided into Wi–Fi users and VoIP users according the Operating System type. However, we use the whole user data without using the classification. The campus area is about 200 acres which indicates a sufficiently dense WLAN environment (about 743 APs/km^2). The trace data consists of association, re-association and disassociation events, each of which occurs when a user associates from OFF state to specific AP, when (s)he re-associates from an AP to another AP, and when (s)he disassociates from an AP going to OFF state, respectively.

In addition to the trace data, we also use the location information of APs [18] in this study consisting of `x`, `y` and `z` coordinates, where the `z` coordinate indicates the floor of the AP, and name of the building at which the AP is located. Note that we have assumed that all buildings in the campus stand on a flat ground and each floor has equal height of 10 ft.

Since only 500 APs do have information about their locations, we use only the trace data involved to those APs and users who visited to them. This reduces the number of mobile users to 5845 users, but the number is still large enough for our analysis. In order to verify the AP locations, we overlap the actual map of the Dartmouth College [19] and the MATLAB plot of the location data, which is shown in Fig. 1. The black squares in the figure fit well within the buildings (brown shaded areas) which shows the locations of APs match well with the locations of buildings.

3.2. Filtering-out false-positive transitions

In this section, we define *false-positive* user registration patterns for elusive transitions that are abused as user's physical movements, and propose a framework for filtering them out from the original trace data. Let i denote the AP with which a user was previously associated, j denote the AP with which the user plan to associate, and T_f represent the *false-positive* transitions. Firstly, we divide T_f into three classes: (1) transition from i to j where both APs are in the same building but there are one or more floors between them where at least one AP is located ($T_{f,s}$); (2) transition from i to j where each AP is in different building from each other but the transition is not geographically possible since there is no connecting infrastructure between two buildings, or another AP exists between the two APs ($T_{f,d}$); (3) ping-pong transition ($T_{f,p}$).

In order to detect transitions in first two cases, we use only building name and floor information since we do not have any obstacle information inside the buildings. Recall we assume that a user will associate another AP in the same floor since the barrier between floors is assumed to prevent any association through it. Additionally, the Cisco specification states that the transmission range of the AP indoors is from 130 to 350 feet, which makes it reasonable to assume that a user will associate APs in the same floor. Note that some buildings are connected to each other through a skywalk or a connecting infrastructure at all floors even though they are different buildings in the trace data. For example, *ResBldg83* and *AcadBldg18* are connected to each other at all floors and thus can be regarded as the same building as shown in Fig. 2(a). In order to detect this, we employ the Google Earth [20]. As shown in Fig. 2(b), the Google Earth offers 3-D building presentation in a great detail so that we can confirm not only the existence of the connection but also the location of skywalk. Note that 28 buildings are fully connected to each other and 7 buildings has a skywalk connected to another building at the Dartmouth campus.

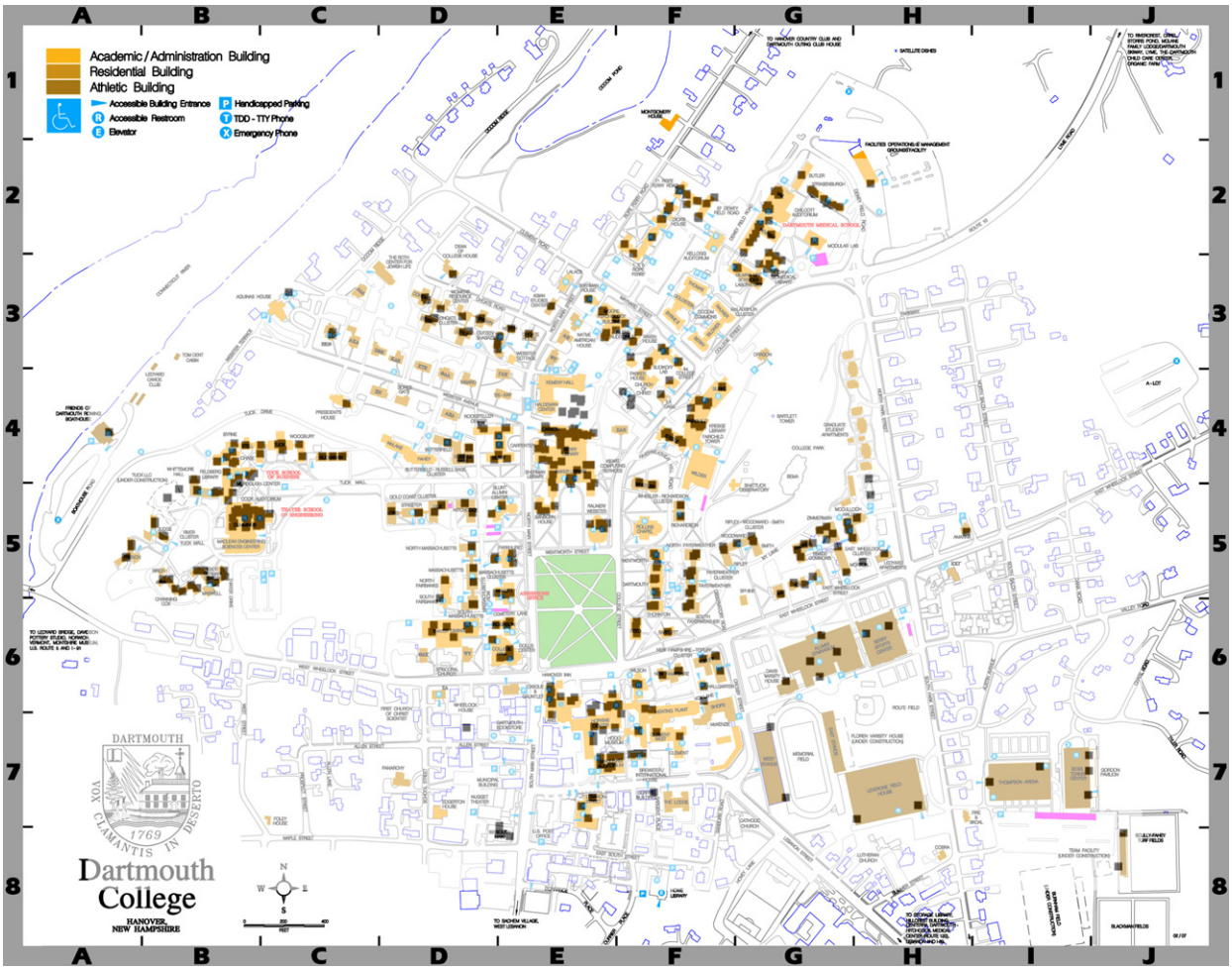
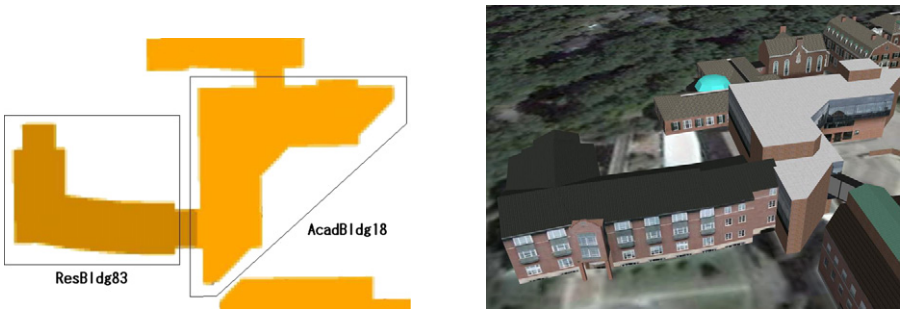


Fig. 1. The AP locations plotted on Dartmouth College map where the symbol of black squares indicate individual APs.



(a) (b)

Fig. 2. An example of connected but differently marked buildings. (a) Resbldg83 and AcadBldg13 are connected to each other according to the map. (b) The Google Earth presents how Resbldg83 and AcadBldg13 are connected to each other.

As for the third case, we define ping-pong transition when transitions are satisfied with:

(a) $j \in B_{n_u}$ and
(b) $p_{n_u} = p_{n_{max}}$,

where

(i) B_{n_u} is the set of APs in n_uth transition of user u, which contains at most l different recently associated APs.
(ii) p_{n_u} is the ping-pong counter operating in what follows:

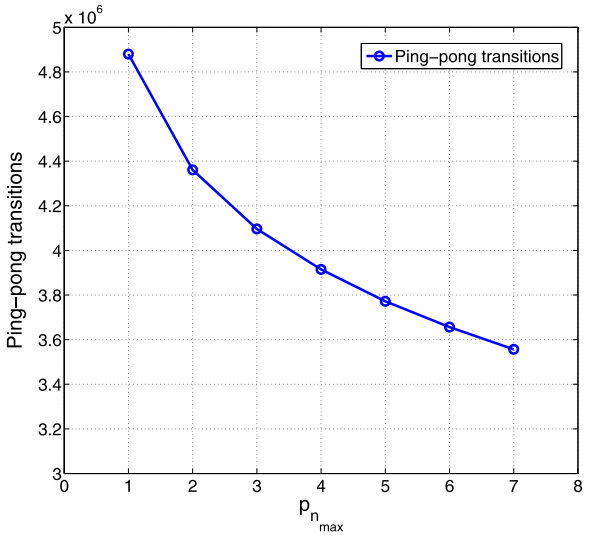


Fig. 3. Number of ping-pong transitions versus $p_{n_{maxF}}$.

$$p_{n_u} = \begin{cases} 0 & \text{if } 1_{B_n}(j) = 0 \text{ and } p_{n_u-1} = 0, \\ & \text{or } n_u \text{ is a transition to the OFF state,} \\ p_{n_u-1} - 1 & \text{if } 1_{B_n}(j) = 0 \text{ and } 0 < p_{n_u-1}, \\ p_{n_u-1} + 1 & \text{if } 1_{B_n}(j) = 1 \text{ and } p_{n_u-1} < p_{n_{max}}, \\ p_{n_{max}} & \text{if } 1_{B_n}(j) = 1 \text{ and } p_{n_u-1} = p_{n_{max}}, \end{cases} \tag{1}$$

where $p_{n_{max}}$ is the maximum value of p_{n_u} and $1_A(\cdot)$ is the indicator function which is

$$1_A(x) = \begin{cases} 0 & \text{if } x \notin A, \\ 1 & \text{if } x \in A. \end{cases} \tag{2}$$

In the above algorithm of identifying *ping-pong* transitions, the l value indicates how many APs are involved to *ping-pong* transitions. Admitting that the *ping-pong* transitions mostly occur between two APs, we observed that they occur in a fair amount between up to four APs in dense area of APs. In order to accommodate the *ping-pong* transitions involved to multiple APs and simultaneously avoid removing *positive* re-associations, we employ the variable of p_{n_u} in the algorithm. As shown in (1), p_{n_u} increases when the current associated AP already exists in B_{n_u} and decreases when it does not. The current transition is counted as *ping-pong* transition only when p_{n_u} has reached the maximum value $p_{n_{max}}$. The reason why we introduce $p_{n_{max}}$ in Eq. (1) is because we would like to give a tolerance threshold to transitions between a pair of APs before they are regarded as *ping-pong* transitions from the trace data. We can henceforth effectively remove *ping-pong* transitions by setting the $p_{n_{max}}$ value appropriately. Note that larger (smaller) $p_{n_{max}}$ value reduces (increases) *ping-pong* decisions while increasing (decreasing) correctness of the decision. To find an optimal $p_{n_{max}}$ value, we first identified a *ping-pong* session as a sequence of consecutive *ping-pong* transitions, and then analyzed the relationship between the lengths and number of sessions according to $p_{n_{max}}$ values. Finally, we can predict to what extent the $p_{n_{max}}$ value influences on estimating the number of *ping-pong* transitions.

Fig. 3 presents the number of *ping-pong* transitions as $p_{n_{max}}$ increases. We observe from the figure that the number sharply decreases from $p_{n_{max}} = 1$ to $p_{n_{max}} = 2$, but after that slowly decreases. So, we use $l = 4$ and $p_{n_{max}} = 2$ for the proposed framework.

Note that the algorithm sets p_{n_u} to 0 when it goes to OFF state since a user is likely to move to a different location when (s)he turns on his (her) device again after (s)he terminates his (her) session.

Finally, we determine the transition count for each class of T_f as follows:

$$|T_{f,s}| = \sum_{u=1}^{|U|} \sum_{n_u=1}^{N_u} E(A_{i,j}), \quad b_i = b_j, \tag{3}$$

$$|T_{f,d}| = \sum_{u=1}^{|U|} \sum_{n_u=1}^{N_u} E(A_{i,j}), \quad b_i \neq b_j, \tag{4}$$

$$|T_{f,p}| = \sum_{u=1}^{|U|} \sum_{n_u=1}^{N_u} 1_{B_{n_u}}(j) \cdot \left\lfloor \frac{p_{n_u}}{p_{n_{max}}} \right\rfloor, \tag{5}$$

where u is a WLAN user index, U is the set of WLAN users, n_u is the transition index of user u's trace, N_u is the total number of transitions in user u's trace, $A_{i,j}$ is the set of APs positioned over the shortest path from i to j with respect to the number of passing floors, b_i is the building name at which i is located, and $E(\cdot)$ is the step function defined as

$$E(A) = \begin{cases} 0 & \text{if } |A| = 0, \\ 1 & \text{if } |A| > 0. \end{cases} \tag{6}$$

When we use (3)–(5) to analyze the Dartmouth College trace data, we found out that each *false-positive* transition, $T_{f,s}$, $T_{f,d}$, and $T_{f,p}$ occupies 280 611, 1 345 309 and 4 360 764 in total 5 138 484 transitions, respectively. Since *false-positive* transitions occupy more than 90% ($\approx 90.36\%$, after removing duplications) of total re-association transitions, we expected *false-positive* transitions to impose a great impact on empirical mobility models. Note that about 38% of total transitions in UCSD trace were *false-positive* where only $T_{f,p}$ was measured due to lack of location information of UCSD APs.

4. Performance evaluation on the filtering framework

In this section, we analyze the impact of *false-positive* transitions on the accuracy of empirical mobility models. For the analysis, we employed Model T [2] to compare the accuracy between two empirical models, each of which was made from the original and *refined* trace,[1] respectively. Since Model T in [2] was constructed with a different data from what we used, we performed the whole analytical procedure specified in [2] again with the data sets used for developing the proposed filtering framework (Dart04) and trace data from UCSD.

As the metrics of comparison, we used: (i) *transition probability distributions*, (ii) *cluster size versus Weibull parameters*, and (iii) *characteristics of AP clustering* since these metrics play the important role of deriving the Model T. As mentioned in Section 2, clustering the APs according to the number of transitions shows that user registration patterns form a distinct hierarchy within each cluster and between clusters. Note that *transitions within a cluster* are called *intra-cluster transitions* and *transitions between clusters* are called *inter-cluster transitions*. The readers are referred to [2] for a detailed account of this clustering.

4.1. Transition probability distributions

For a cluster C, we created a $|C| \times |C|$ intra-cluster transition matrix N to obtain intra-cluster transition probability distribution as defined in [2] where $N(u, v)$ is the number of intra-cluster transitions from AP u to AP v in C. The intra-cluster transition matrix M for C is then

$$M(u, v) = \frac{N(u, v)}{\sum_i N(u, i)}, \tag{7}$$

where we included the OFF state as stated in [13].

Inter-cluster transition probability matrix was made in the similar manner except that each AP is replaced by individual cluster and the whole network is treated as one large cluster. The resulting distributions were fitted into Weibull CDF which is:

$$F(x) = 1 - e^{-(\frac{x}{a})^b}, \tag{8}$$

with *non-linear least squares fit* with *robust option* in the MATLAB curve fitting tool.

Finally, we obtained the transition probabilities for both the original and *refined* traces. The results for intra-cluster ($|C| = 12$ for Dart04 trace and $|C| = 7$ for UCSD trace) and inter-cluster transitions are shown in Figs. 4 and 5, respectively. Specific numerical results in the perspective of the goodness-of-fit measures for Dartmouth trace data in all cluster sizes are shown in Tables 1 and 2. Table 1 shows the results for intra-cluster transitions while Table 2 shows those results for inter-cluster cases. Note that some entries in Table 1 were not available because the distribution of clusters for the original and *refined* trace were different from each other and thus produced different output with MCL tool. As we can see for both tables, the Weibull CDF fits much better for the *refined* trace in both intra- and inter-cluster cases. This result means that the *false-positive* transitions make the fitting errors in Model T. The results for UCSD trace data also showed improvements in goodness-of-fit measure for both intra- and inter-cluster cases.

[1] We use the terminology of "*refined* trace" for the trace that we obtained by removing *false-positive* transitions in the original trace.

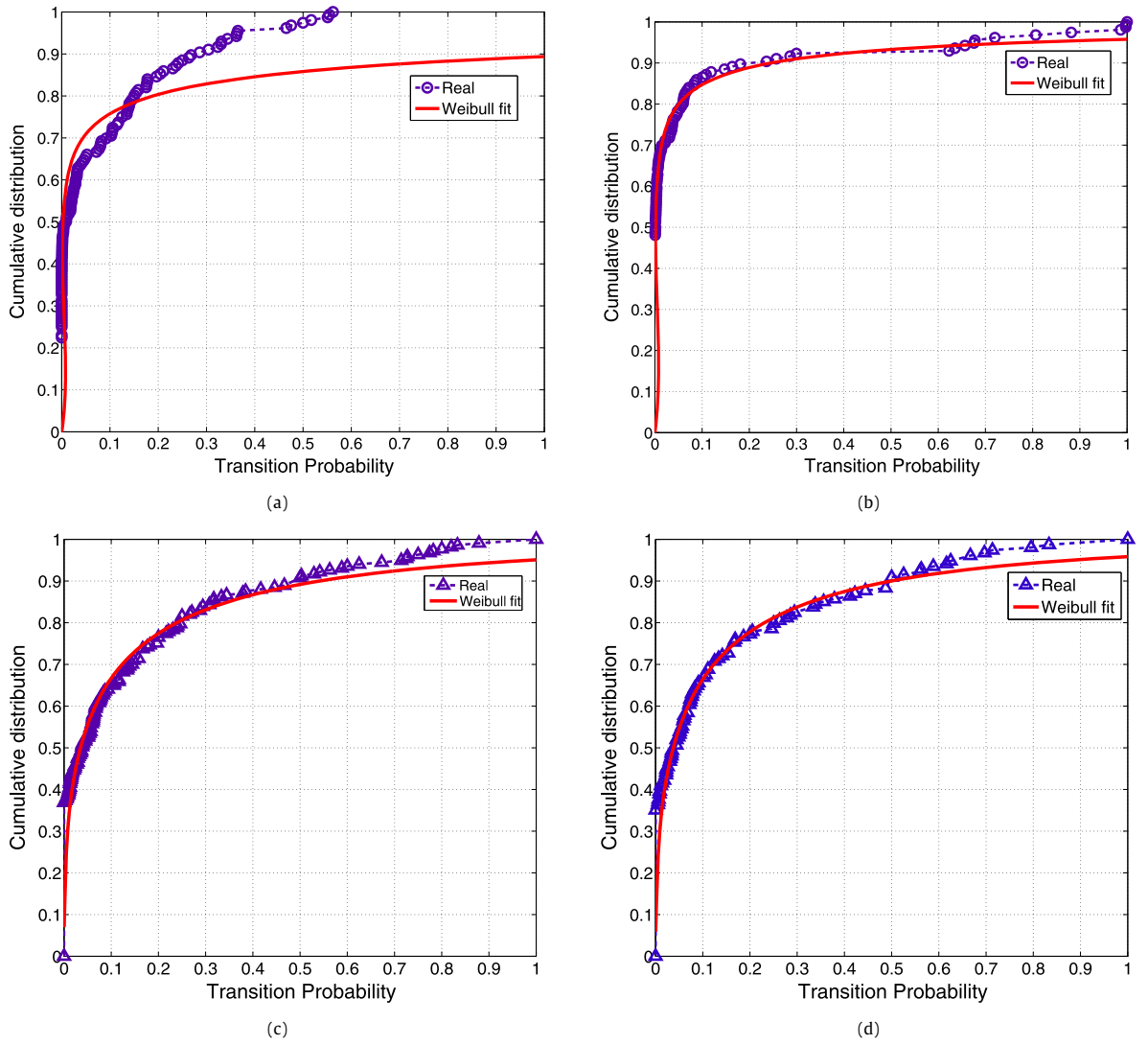


Fig. 4. Comparison between intra-cluster transition probabilities. (a) Empirical CDF and its Weibull fit for intra-cluster transition probabilities when $|C| = 12$ in the original trace (Dart04 trace). (b) Empirical CDF and its Weibull fit for intra-cluster transition probabilities when $|C| = 12$ in the *refined* trace (Dart04 trace). (c) Empirical CDF and its Weibull fit for intra-cluster transition probabilities when $|C| = 7$ in the original trace (UCSD trace). (d) Empirical CDF and its Weibull fit for intra-cluster transition probabilities when $|C| = 7$ in the *refined* trace (UCSD trace).

4.2. Cluster size vs. Weibull parameters

Multiple CDF-based modeling corresponding to each cluster size produces a huge amount of modeling parameters since clustering creates tens of cluster sizes and we need two Weibull parameters (a and b) for each cluster size. Model T [2] provides a higher level of abstraction that reduces the number of these parameters, which fits each Weibull parameter with the following exponential equation:

$$y = p_1 e^{-p_2 x} + p_3. \tag{9}$$

Before proceeding to detailed modeling, we note the characteristics of each Weibull parameters as follows in order to give the insight about its relation with cluster size. The first parameter a is called the scale parameter and the second parameter b is called the shape parameter. The former stretches out the Weibull pdf when it increases while the latter (b) is fixed, which relaxes the steepness of the Weibull CDF. The latter changes the shape of its pdf from a monotonically decreasing form to a bell-shaped one when it increases from 0 while the former (a) is fixed. According to Eq. (7), the transition probabilities between each pair of APs are expected to decrease as cluster size increases, resulting a continuous decrement of parameter a. However, the effect of varying cluster size on parameter b remains ambiguous.

Fitting the parameters of original and *refined* trace data into Eq. (9) gives a better insight of these parameters. We employed the same fit options as used in Section 4.1, and presented the results in both Table 3 and Fig. 6. As expected,

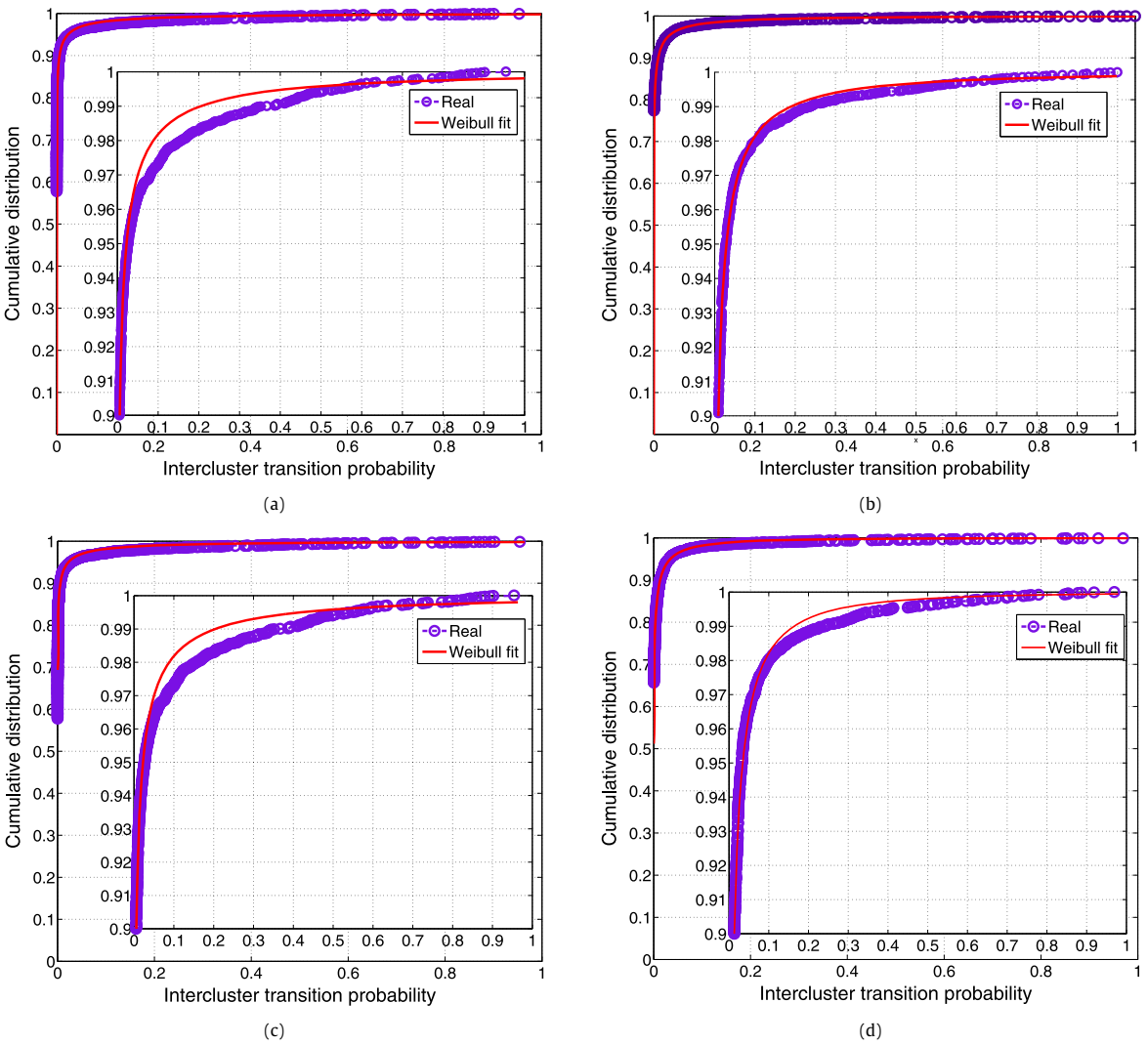


Fig. 5. Comparison between inter-cluster transition probabilities. (a) Empirical CDF and its Weibull fit for inter-cluster transition probabilities in the original trace (Dart04 trace). (b) Empirical CDF and its Weibull fit for inter-cluster transition probabilities in the *refined* trace (Dart04 trace). (c) Empirical CDF and its Weibull fit for inter-cluster transition probabilities in the original trace (UCSD trace). (d) Empirical CDF and its Weibull fit for inter-cluster transition probabilities in the *refined* trace (UCSD trace).

Table 1
Goodness-of-fit measures for intra-cluster transitions between the original and *refined* trace.

$\|C\|$	rmse, original	rmse, *refined*	R^2, original	R^2, *refined*
3	0.0378	0.0179	0.9822	0.9960
4	0.0440	0.0185	0.9731	0.9956
5	0.0429	0.0152	0.9743	0.9964
6	0.0453	0.0181	0.9689	0.9949
7	0.0609	0.0235	0.9331	0.9922
8	0.0378	0.0216	0.9822	0.9887
9	0.0320	0.0099	0.9854	0.9983
10	0.0454	0.0183	0.9553	0.9911
11	0.0118	0.0383	0.9981	0.9826
12	0.0705	0.0207	0.9078	0.9861
13	0.0454	0.0196	0.9562	0.9913
14	0.0164	N/A	0.9931	N/A
15	0.0286	N/A	0.9865	N/A
16	N/A	0.0174	N/A	0.9916
17	0.0473	N/A	0.8786	N/A
21	0.0294	N/A	0.9816	N/A
22	N/A	0.0230	N/A	0.9903

Table 2
Goodness-of-fit measures for inter-cluster transitions between the original and *refined* trace.

	rmse, original	rmse, *refined*	R^2, original	R^2, *refined*
Inter	0.00504	0.00277	0.9978	0.9981

Table 3
Parameter values and Goodness-of-fit measures for fitting Weibull parameters.

original				*refined*			
a		b		a		b	
p_1	1.129	p_1	2.222	p_1	1.506	p_1	0.6334
p_2	0.3864	p_2	0.3871	p_2	0.5213	p_2	0.2603
p_3	0.01332	p_3	0.2651	p_3	0.009609	p_3	0.2902
rmse	0.0169	rmse	0.1074	rmse	0.0106	rmse	0.1007
R^2	0.9778	R^2	0.8043	R^2	0.9875	R^2	0.4664

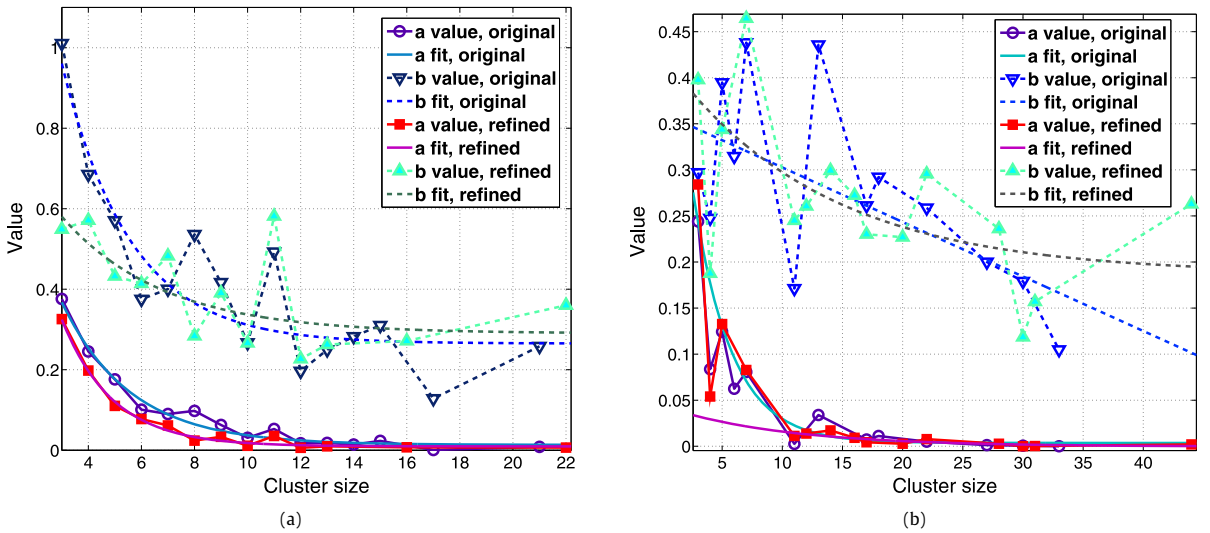


Fig. 6. Distribution of Weibull parameters and its fitted curves. (a) Dartmouth College trace data. (b) UCSD trace data.

parameter a is fitted well with the monotonically decreasing exponential curve. However, parameter b seems to be set with any value in the range of $[0, 1]$ rather than being well fitted with the curve. This can be interpreted in the way that the shape of Weibull CDF per cluster size is similar to each other.

4.3. Characteristics of AP clustering

Once we obtained clusters, the original trace of Dartmouth College had clusters of sizes $\{2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 17, 21\}$ and the *refined* trace had $\{2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 16, 22\}$. Even though two distributions of cluster sizes do not differ much from each other, the APs in each cluster became more geographically packed after removing *false-positive* transitions. The average distances between APs in each clusters were computed with following equation:

$$\frac{\sum_{i,j\in C}\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}{|C|\cdot|C-1|/2}, \quad i \neq j, \tag{10}$$

where the denominator indicates the number of AP pairs in cluster C and the nominator indicates the sum of distances of the AP pairs in C.

By averaging the results for all clusters, the average distance for the original trace was 348.03 feet and that for the *refined* trace was 203.49 feet. The result is shown in Fig. 7. Clusters of the same size have been averaged.

One interesting result after refining the original trace is that 50 clusters have the same APs that belongs to specific buildings (including connected building). For example, the APs in the cluster of $|C| = 22$ in the *refined* trace exactly matched with the APs in *AcadBldg10*. This is noticeable since only 11 clusters have those exact matchings in the original trace. Combined with the result from average distances, we can interpret this result as the evidence that many transitions are irrelevant to actual user movement.

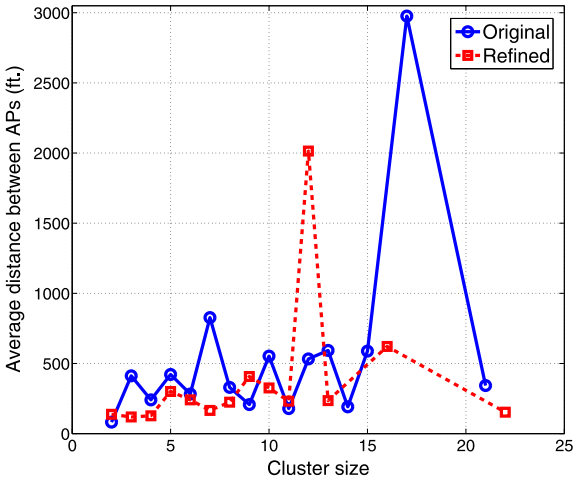


Fig. 7. Average distance between APs.

5. Realistic user registration model

In this section, we propose an analytical model for user registration patterns, inclusive of *false-positive* and *positive* user registrations, by which we can improve the accuracy in predicting (estimating or modeling) user association patterns in WLANs. Note that *false-positive* transitions occur even when the user device is almost immobile in a dense WLAN environment (as discussed in Section 3), so that they should be incorporated into user mobility model, especially when the network environment is assumed to be dense. Definitely, *positive* transitions are considered as real user movements.

In order to model the *false-positive* transitions, we use the SNMP log provided from Dartmouth College [9] where we extracted the principal features of the transitions because it provides the RSSI values of WLAN devices polled from APs in every five minutes. This enabled us to directly see how a WLAN device changes its association when it is within the transmission range of several contending APs. As for the *positive* transitions, we exploit the Bayesian classifier with a small modification to generate registration patterns for each user according to each user specific mobile patterns in the given geographical property. Different from many mobility models, including [2], which focus on the probability distributions from total registration statistics to determine the next association for an individual user, we capture the mobile patterns for each user with a recent registration history of each individual user.

5.1. Modeling false-positive user registrations

As discussed in Section 3, in case of large number of APs in the WLAN, such high density of APs incurs a *false-positive* transition even though the user is stationary. Therefore, we need to include the *false-positive* registration patterns in empirical mobility model. Note that a recent update on the locations of APs in Dartmouth College [21] supports this argument since 70% more number of APs has been installed at the campus. Additionally, the user density per AP also becomes severe in indoor environments because the area to which users can move around is very limited, compared to outdoor environment. Therefore, by including the *false-positive* patterns, we realistically emulate user mobile patterns (or registration patterns) for WLAN users, and also we can use it to develop intelligent hand-off algorithm and/or load-balancing algorithm.

The fundamental question is how we can identify *false-positive* registrations to reflect them into a prospective mobility model, and the next question is what types of *false-positive* registrations we need to emulate in the prospective model.

How to decide false-positive *registrations.* Since one of main criteria that determines the link state is the received signal strength (RSS) and the distance between an AP and a user device can be estimated with the RSS value, we decide to use the RSS value to determine *false-positive* registrations. We collect a set of RSS values for each user (or WLAN device), each of which is measured from each neighboring AP. Based on these information, we then classify the different pattern of residence time at each AP during the period of *false-positive* registrations, which is used to determine what AP dominates to connect with the user during the period. For example, Fig. 8 presents two different distributions of user residence time: the total residence time of AP 362 and 364 in Fig. 8(a) (the numbers on the rightmost side denote the residence times) is almost equal to each other, whereas those time in Fig. 8(b) are varying over APs. We make use of SNMP log data from Dartmouth College collected from November 2003 to February 2004 which includes RSSI values for each user measured from the latest packet received from APs. But, since the data was collected by a polling method at every five minutes, there exists inconsistency with the syslog trace data. Note that the terminology "*syslog* trace data" is identical to "the syslog trace data" and is used to be distinguished from the SNMP log data, and that each record in Fig. 8 consists of date, time, a day of the week, AP identifier, and the residence time. To address this inconsistency problem, we average the RSSI values measured

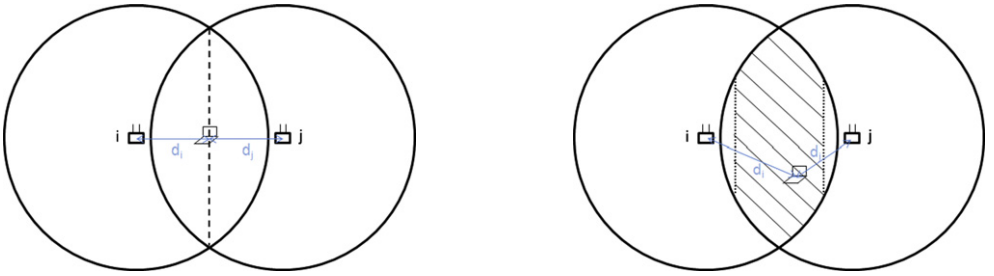
```
2003-08-26 08:32:51 Tuesday [364]   507
2003-08-26 08:41:18 Tuesday [362]   240
2003-08-26 08:45:18 Tuesday [364]   516
2003-08-26 08:53:54 Tuesday [362]   242
2003-08-26 08:57:56 Tuesday [364]   244
2003-08-26 09:02:00 Tuesday [362]   241
2003-08-26 09:06:01 Tuesday [364]   273
2003-08-26 09:10:34 Tuesday [362]   240
2003-08-26 09:14:34 Tuesday [364]   252
2003-08-26 09:18:46 Tuesday [362]   185
2003-08-26 09:21:51 Tuesday [364]   911
2003-08-26 09:37:02 Tuesday [362]   239
2003-08-26 09:41:01 Tuesday [364]   453
2003-08-26 09:48:34 Tuesday [362]   239
2003-08-26 09:52:33 Tuesday [364]   559
```

(a)

```
2002-09-26 10:13:03 Thursday [366]   238
2002-09-26 10:17:01 Thursday [367]   4
2002-09-26 10:17:05 Thursday [366]   1536
2002-09-26 10:42:41 Thursday [367]   3
2002-09-26 10:42:44 Thursday [366]   31
2002-09-26 10:43:15 Thursday [367]   3
2002-09-26 10:43:18 Thursday [366]   2779
2002-09-26 11:29:37 Thursday [367]   42
2002-09-26 11:30:19 Thursday [366]   950
2002-09-26 11:46:09 Thursday [367]   3
2002-09-26 11:46:12 Thursday [366]   1003
2002-09-26 12:02:55 Thursday [367]   7
2002-09-26 12:03:02 Thursday [484]   4
2002-09-26 12:03:06 Thursday [367]   151
2002-09-26 12:05:37 Thursday [366]   305
```

(b)

Fig. 8. Examples for presenting distinct patterns of false-positive transitions. (a) Syslog trace of a user who has almost equal distribution of residence time between two APs. (b) Syslog trace of a user who has different distribution of residence time between two APs.



(a) (b)

Fig. 9. The area of incurring false-positive registrations. (a) A set of theoretical points when *false-positive* transitions occur ($d_i = d_j$). (b) A realistic area when *false-positive* transitions occur ($|d_i - d_j| < \alpha$).

during the residence time to represent the RSSI value for each association. Additionally, there exists another inconsistency in the RSSI values from January 2004 to February 2004. The inconsistency is that two types of measurement units for RSSI values appear in the period, which actually happens when Cisco 350 APs migrated from running `VxWorks` to the Cisco `IOS`. Cisco `IOS` recorded the RSSI values in dBm unit where `VxWorks` recorded those values in normalized value ranging from 1 to 100. For this reason, we excluded these two months SNMP data and used the data from the remaining two months.

We made the following assumptions in the course of developing user registration model for *false-positive* user registrations:

(i) the transmission range (or power) of all APs are identical;
(ii) *false-positive* transitions occur between two APs;
(iii) no obstacle is assumed to exist;
(iv) the channel noise that affects the RSSI value is constant.

As the first step, we consider the most simple case when there are only two contending APs i and j, and the WLAN device is located at the same distance from i and j, namely d_i and d_j. The configuration is presented in Fig. 9(a), where the dashed line indicates that the user can reside within any spot along the line to satisfy the above assumption. Note that this configuration presents the simplest environment in order for a WLAN device to generate *false-positive* transitions because the theoretical RSSI value of each AP is equal to each other and the fluctuation of the channel state directly affects the associations.

We then extend the positions over the dashed line by giving a tolerance value of α to the difference between d_i and d_j considering various radio parameters:

$$|d_i - d_j| < \alpha. \tag{11}$$

This extension is described with the shaded area of Fig. 9(b).

In the configuration of Fig. 9(b), one of the APs starts to connect to the user device, but, due to the unreliable channel state due to various types of interference, it loses its connection with the device after for some period of time, where this time interval is inversely proportional to α. That means the farther the user moves away from AP i (AP j) and approaches AP j (AP i), the less AP i (AP j) gets the chance to associate with the user. This can be explained with the following Friis formula that specifies the relationship between the RSSI value and the transmission power of AP:

$$\frac{P_r}{P_t} = G_t G_r \left(\frac{\lambda}{4\pi d}\right)^{\theta}, \tag{12}$$

where P_r is the power received by the receiving antenna, P_t is the power input to the transmitting antenna, G_t and G_r are the antenna gain of the transmitting and receiving antennas, respectively, λ is the wavelength and d is the distance between AP and the WLAN device, and θ is the path loss parameter in between 2 and 4. If we remove all the constants and the fixed variable P_t, only the relationship between P_r and d remains ($P_r \propto 1/d^{\theta}$). Thus, α is dependent on P_t, where P_t increases, α also increases. Since deciding the exact value of α requires the knowledge of all the variables, including P_t and d_i, d_j, which we cannot infer even from the SNMP log data, we leave it as a variable.

How to classify false-positive *transitions w.r.t. residence time.* Then, we will discuss about the residence time ratio, r_{res}, in the area that may incur *false-positive* transitions. Note that the residence time ratio, r_{res}, is defined as the following value:

$$r_{res} = \frac{\min(s_i, s_j)}{s_i + s_j}, \tag{13}$$

where s_i and s_j indicates the residence time in AP i and j, respectively. By intuition, r_{res} will decrease as $|d_i - d_j|$ increases because the dominant AP is more likely to seize the association with the WLAN device for a longer period of time that the other APs, and so the recessive AP becomes the numerator in Eq. (13).

We now empirically derive a model for r_{res} with the SNMP log data. Since we do not have any location data of WLAN users, which is required to determine the distances between each user and APs, we will infer $|d_i(\tau_\rho) - d_k(\tau_\rho)|$ from $|RSSI_i(\tau_\rho) - RSSI_k(\tau_\rho)|$, where ρ is the index for a SNMP time instance, $d_i(\tau_\rho)$ is the distance between AP i and the user at the time instance τ_ρ and $RSSI_i(\tau_\rho)$ is the measured RSSI value at the time instance τ_ρ. Note that AP k is the AP that contends with AP i for the next association with the user and is not AP j. Also we assume that $|RSSI_i - RSSI_j|$ and $|d_i - d_j|$ have a positive relationship from Eq. (14), which is:

$$|RSSI_i - RSSI_k| \propto \left|\frac{1}{d_i^{\theta}} - \frac{1}{d_k^{\theta}}\right|. \tag{14}$$

We first extracted T_f for all users as described in Section 3 but only between two APs in this case (recall the second assumption). Then, we track the syslog trace and SNMP log data simultaneously to judge *false-positive* user registrations and obtain the residence time of APs and RSSI values of WLAN devices. Note that in this procedure of tracking, we need to synchronize the registration pattern in the syslog trace and RSSI values in SNMP log because the model basically relies on the syslog trace and the SNMP log is only referred to obtain RSSI value when *false-positive* registration is detected. When we detect a *false-positive* transition, we collect all the RSSI values of AP i and AP k while the user is associated with AP i. In the case that we observed multiple RSSI values from the same AP, we averaged those values. Thus, $|RSSI_i(\tau_\rho) - RSSI_k(\tau_\rho)|$ value(s) measured during the period of association with AP i can be described by the following normalized RSSI value (in syslog trace):

$$RSSI_{ik}(t_i, t_j) = \frac{1}{M_{t_i,t_j,(i,k)}} \sum_{h=1}^{M_{t_i,t_j,(i,k)}} \left|RSSI_i(\tau_\rho) - RSSI_k(\tau_\rho)\right|, \tag{15}$$

where t_i denotes the association time instance of i and $M_{t_i,t_j,(i,k)}$ denotes the number of available RSSI sets $(RSSI_i, RSSI_k)$ that are measured between $[t_i, t_j]$. We then compute r_{res} using the relationship $s_i = t_j - t_i$. In the procedure of discovering the relation between r_{res} and $RSSI_{ik}(t_i, t_j)$, we arrange 10 number of bins by equally dividing the value of $RSSI_{ik}(t_i, t_j)$, assign the computed r_{res} value to an appropriate bin indexed with the $\lfloor \frac{RSSI_{ik}(t_i,t_j)}{10} \rfloor$, and average the r_{res} values in the collected set in the end. This approximation is acceptable because we simply want to discover the relationship between $RSSI_{ik}(t_i, t_j)$ and r_{res}. The empirical results for this classification is presented in Fig. 10.

In Fig. 10, the blue line tagged with cross marks presents the results of the above calculations. We can see that until $RSSI_{ik}(t_i, t_j) \leqslant 20$, r_{res} does not vary much. This results from (1) little difference between $RSSI_i$ and $RSSI_k$ values and (2) the effect of threshold value of $|RSSI_i - RSSI_k|$ set in WLAN cards to prevent frequent hand-offs. After that point, however, we can see that r_{res} drops in a steady manner except the last r_{res} value, which was distorted by one WLAN device that continuously associated to AP i with $1 \leqslant RSSI_i \leqslant 10$ instead of AP k with $60 \leqslant RSSI_k$. We thus deliberately removed this special case form the data set. After this removal, we have more stable results: the green dashed line tagged with circle marks in Fig. 10 shows the result. With the results presented with the green dashed line, we can fit the residence time with the exponential equation in Eq. (9). When $p1 = 0.272$, $p2 = 0.05627$ and $p3 = 0.1779$, the fitting result showed goodness-of-fit measures $R^2 = 0.9889$ and $rmse = 0.003679$. This fitting was proved to be the best in our investigation among other distributions, such as power law and linear polynomial.

5.2. Modeling positive user registrations

The main objective of modeling *positive* transitions is to obtain real user movements, which is different from *false-positive* transitions in that the distances between a user and APs are physically changed over time. We basically use the

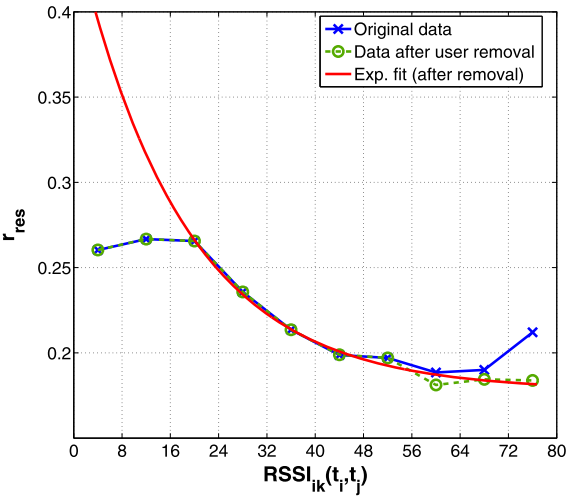


Fig. 10. $RSSI_{ik}(t_i, t_j)$ vs. r_{res}.

Algorithm 1 Algorithm for generating positive transitions

Require: i is the current associated AP,
j is the next associating AP,
reg_data_u is the registration data of u,
$\vec{x}$ is the vector storing the CDF data.

```
|U| ← user input
N_u ← user input or a given probability distribution
for u in U do
    i ← Pick from any AP that has P(i) > 0
    for 1 to N_u do
        x⃗ ← Create_CDF(reg_data_u, i)
        rand_no ← Create_rand(0, 1)
        j ← AP that has closest CDF value with rand_no
    end for
end for
```

Bayesian approach to achieve this goal. The reason why we use this approach is because each user has his (or her) own specific association patterns, which varies greatly depending on user characteristics and topological (and/or geographical) environment. Specifically, among the Bayesian classification, we use the Minimum–Error–Rate classification which is as follows:

$$\text{Decide } j \text{ if } \quad P(j|i) > P\left(j'|i\right), \quad \forall j' \neq i,\ j' \in J_i, \tag{16}$$

where $P(j)$ is the probability that the user associates with AP j after current association with AP i is released and the J_i is the set of all candidate APs with which the user can associate after the current association. Note that $P(j|i)$ is the *posterior* probability which is mainly used for the decision. Assuming we have sufficient data at the starting point of modeling, it is possible to generate synthetic trace for each user by calculating $P(j'|i)$, $j' \in J_i$, in Eq. (16). However, applying Eq. (16) directly to the association pattern data gives fixed result for the same input i and u since the classifier will always choose j which has the maximum $P(j|i)$ value. We thus modify the classifier to build a CDF from $P(j'|i)$ values creating $F(index_{j'})$ and let the classifier pick one as j, where $F(index_j)$ has the closest value to a randomly picked number in range $[0, 1]$. This will give other APs a chance to be selected even though their *posterior* value is small.

Positive transition generating algorithm. Based on the modified Bayesian classifier, we define an algorithm to model *positive* user registration, which actually generates *positive* user registrations. The detailed algorithm is referred to Algorithm 1. The algorithm assumes that the registration history data of user u is given in advance, and uses the following functions:

(1) $Create\_CDF(reg\_data_u, i)$: creates CDF of $P(j'|i)$ values extracted from reg_data_u;
(2) $Create\_rand(0, 1)$: creates a random number drawn between the range [0,1].

In the algorithm, the number of users $|U|$ can be determined according to the network scale, and the number of transitions of each user N_u has a distribution. From our statistical result, an example of well fitted distribution of N_u is Gamma or Weibull distribution.

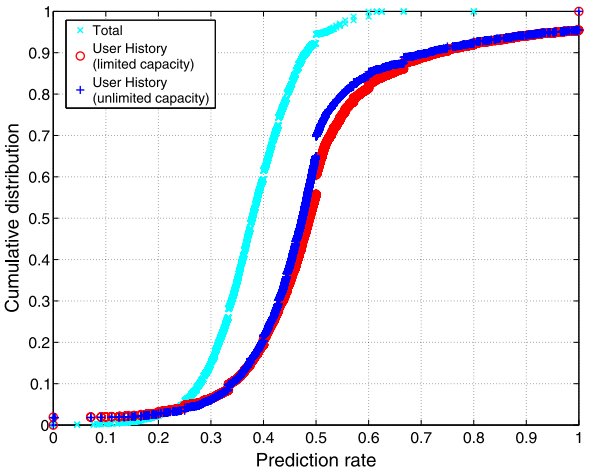


Fig. 11. Comparison between prediction algorithms.

The modified Bayesian classifier can also be applied to predict the next AP for user's reassociation, which can effectively assist any hand-off procedures. The specific method for the prediction is similar to Algorithm 1 but we need to use the real trace data in order to follow the sequence of real trace and verify the correctness of the prediction. Another major difference with Algorithm 1 is that we need to update reg_data_u whenever there is a transition from h to i, where h is the previously associated AP. Note that since we simulate the WLAN device, there is a limit on the capacity of reg_data_u regarding the computational overload.

We compare the proposed prediction method with two other methods. One method is when we remove the capacity constraint of res_data_u, and the other one is when we use the whole transition statistics from the real trace and build a transition matrix that stores all the transitions from trace data. The latter actually removes the effect of dealing each user separately.

The result is shown in Fig. 11, where *total* denotes the prediction made from total statistics and others used user-specific approach with (or without) the size limit of reg_data_u. The figure indicates that user-specific approach gives a much better result compared to the *total* method. Surprisingly, limiting the capacity of the registration data gives a slightly better result, implying that locality of user association patterns plays an important role.

6. Conclusion

In this paper, we investigated the trace data collected from the Dartmouth College WLAN networks, and identified elusive user transitions that are abused as actual user movements. We then proposed a framework of filtering out the *false-positive* transitions in order to retain only actual user movements in empirical mobility models. We employed Model T and its extension to verify the effect of the proposed framework on real mobility model. The evaluation has been done by comparing the transition probability distribution and cluster characteristics between the original and *refined* Dartmouth College trace, and also done with UCSD trace. Similar statistical results from the two trace data shows that removing elusive registration patterns improves the fidelity of empirical mobility models. Based on the *refined* trace and the examination on the user registration patterns in the dense Wi–Fi networks, we proposed a realistic mobility model, inclusive of *false-positive* and *positive* registrations. Especially, one part of the modeling that emulates *positive* registrations can predict the prospective position of individual users based on a short-term history of user registration patterns. We presented that the prediction statistically is better than when we fully use the whole set of trace data.

We have several research directions for future research. We plan to investigate additional set of parameters to determine *false-positive* registration patterns and also to elaborate the algorithm for emulating *positive* registrations in the way of reducing the complexity of the algorithm. We also would like to integrate both registration models into a singe integrated framework in order to produce more realistic user mobile patterns in real Wi–Fi HotSpot networks.

Acknowledgments

This work was supported in part by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) (No. 2010-0014060), in part by the IT R&D program of MKE/KEIT [KI001822, Research on Ubiquitous Mobility Management Methods for Higher Service Availability], and in part by the KCC (Korea Communications Commission), Korea, under the R&D program supervised by the KCA (Korea Communications Agency) (KCA-2011-09913-04003).

References

[1] D. Kotz, K. Essien, Analysis of a campus-wide wireless network, in: Proceedings of ACM MobiCom, September 2002.
[2] R. Jain, D. Lelescu, M. Balakrishnan, Model T: An empirical model for user registration patterns in a campus wireless LAN, in: Proceedings of ACM MobiCom, August 2005.
[3] J. Lee, J.C. Hou, Modeling steady-state and transient behaviors of user mobility: Formulation, analysis, and application, in: Proceedings of ACM MobiHoc, May 2006.
[4] J. Yoon, B.D. Noble, M. Liu, M. Kim, Building realistic mobility models from coarse-grained traces, in: Proceedings of ACM MobiSys, June 2006.
[5] I. Rhee, M. Shin, S. Hong, K. Lee, S. Chong, On the Levy-walk nature of human mobility, in: Proceedings of IEEE INFOCOM, April 2008.
[6] W. Hsu, A. Helmy, IMPACT: Investigation of mobile-user patterns across university campuses using WLAN trace analysis, in: Proceedings of IEEE Int'l Workshop on Wireless Network Measurement (WiNMee), April 2006.
[7] M. McNett, G. Voelker, Access and mobility of wireless PDA users, in: ACM SIGMOBILE Mobile Computing and Communications Review, vol. 7, October 2003.
[8] D. Kotz, T. Henderson, I. Abyzov, J. Yeo, CRAWDAD trace dartmouth/campus/syslog/01_04 (v. 2004–12–18), http://crawdad.cs.dartmouth.edu/dartmouth/campus/syslog/01_04, December 2004.
[9] D. Kotz, T. Henderson, I. Abyzov, J. Yeo, CRAWDAD trace set dartmouth/campus/snmp (v. 2004–11–09), http://crawdad.cs.dartmouth.edu/dartmouth/campus/snmp, November 2004.
[10] M. Balazinska, P. Castro, Characterizing mobility and network usage in a corporate wireless local-area network, in: Proceedings of ACM MobiSys, May 2003.
[11] http://cabspotting.org/.
[12] S.v. Dongen, Graph clustering by flow simulation, University of Utrecht, PhD dissertation ed., May 2000.
[13] D. Lelescu, U. Kozat, R. Jain, M. Balakrishnan, Model T++: An empirical joint space-time registration model, in: Proceedings of ACM MobiHoc, May 2006.
[14] M. Kim, Z. Liu, S. Parthasarathy, D. Pendarakis, H. Yang, Association control in mobile wireless networks, in: Proceedings of IEEE INFOCOM, April 2008.
[15] K. Lee, S. Hong, S. Kim, I. Rhee, S. Chong, SLAW: A mobility model for human walks, in: Proceedings of IEEE INFOCOM, April 2009.
[16] T. Henderson, D. Kotz, I. Abyzov, The changing usage of a mature campus-wide wireless network, in: Proceedings of ACM MobiCom, September 2004.
[17] L. Song, D. Kotz, R. Jain, X. He, Evaluating location predictors with extensive Wi–Fi mobility data, in: Proceedings of IEEE INFOCOM, March 2004.
[18] Dartmouth College, Location information of access points in Dartmouth College, http://www.crawdad.org/download/dartmouth/campus/movement, November 2004.
[19] Dartmouth College, The Dartmouth College campus map, http://www.dartmouth.edu/~maps/docs/map0207.pdf.
[20] Google Inc., The Google Earth, http://earth.google.com.
[21] J. Yeo, CRAWDAD trace dartmouth/campus/syslog/aplocations_2008 (v. 2009–09–09), http://crawdad.cs.dartmouth.edu/dartmouth/campus/syslog/aplocations_2008, September 2009.