Confining Wi-Fi Coverage: A Crowdsourced Method Using Physical Layer Information

Bingxian Lu*, Zhicheng Zeng*, Lei Wang*, Brian Peck†, Daji Qiao†, and Michael Segal‡

*Key Laboratory for Ubiquitous Network and Service Software of Liaoning Province
School of Software, Dalian University of Technology, China
†Department of Electrical and Computer Engineering, Iowa State University, USA
‡Communication Systems Engineering Department, Ben-Gurion University of the Negev, Israel

***Abstract*—Many small businesses and public areas offer free Wi-Fi access, but may wish to restrict network access only to their customers or patrons inside the physical property. Unfortunately, due to the nature of wireless networks, this is difficult to accomplish. We develop and implement CLAC, a Crowdsourced Location aware Access Control scheme using physical layer information to address this challenge. It crowdsources both channel state information (CSI) and received signal strength (RSS) of already validated users to classify future users. We propose and use two CSI metrics in CLAC: CSI Cross-Antenna Stability Metric and CSI Cross-Frame Stability Metric, which summarize well the spatial and temporal CSI characteristics respectively. CLAC is evaluated in an office and a classroom. Evaluation results show that CLAC performs well in both environments, allowing most valid users inside the area to access the network, while the chance that invalid users outside the boundary may access the network is small.**

***Index Terms*—Crowdsourcing, Machine Learning, User Validation, IEEE 802.11 WLAN**

I. Introduction

Smartphone and mobile device usage among consumers is currently at an all-time high, and the current growth rate shows no signs of slowing down. As a result, billions of users around the world are using the Internet on the go, whether from cellular data or through Wi-Fi Access Points (APs). Currently, many businesses and public areas seek to attract customers and patrons by offering Internet access through these Wi-Fi APs. However, providing this access is not cheap, as a business has to purchase enough bandwidth to satisfy customers, as well as install enough APs to provide coverage throughout the physical space. Due to these costs, a business may wish to restrict Wi-Fi access only to paying customers, or restrict access to customers inside the physical space.

A. Motivation

As an example scenario, we deliberate on a scene as depicted in Figure 1. The room is equipped with an AP and two additional monitor devices labelled $M1$ and $M2$. The shaded circle around the AP is the region in direct proximity. The figure also shows multiple locations that a user may dwell, labelled 1 through 20. Consider a user sitting at Location 3, directly next to the AP. This user should certainly receive Wi-Fi access, since they are definitely inside the region. While another user outside the room, at Location 19. This user should not receive Wi-Fi access. Finally, inspect a user at Location 10. This user should receive access, since he is inside the room, even though he is not directly near the AP. We design and implement CLAC, which is capable of properly classifying customers as being inside a given area (hereafter referred to as *valid users*). In this example scenario, CLAC would allow the users at Locations 3 and 10 to access the Wi-Fi network, while preventing access to a user at Location 19.

Lei Wang (Email: lei.wang@dlut.edu.cn) is the corresponding author of this work.

B. Intuition behind the CLAC Design

CLAC uses a variety of factors to determine if a given user should be granted access to the network. Most basically, CLAC uses channel state information (CSI) and received signal strength (RSS) to classify a user. CLAC first uses CSI data to determine if a user is in direct proximity of an AP; if so, the user is classified as a valid user. We also collect RSS data from valid user to form a fingerprint. As valid users move around the area, we feed their RSS data into a machine learning algorithm, which quickly populates a training set.

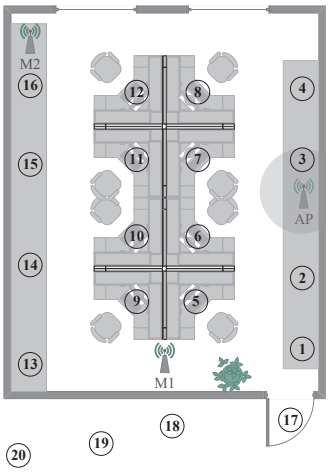


Figure 1: Topology for observations and experiments. The scene has one AP, and two Monitors $M1$ and $M2$ (to be used in the proposed CLAC scheme). Each numbered circle represents a specific location that a user may dwell at.

This crowdsourcing allows CLAC to quickly identify future valid users. Finally, users that are incorrectly classified as being outside the area (False Negatives) could then obtain an access code which would add their device to a whitelist, granting them accesses on future validation attempts.

Some existing schemes use similar fingerprinting techniques to classify users. In most cases, however, extensive war-driving is required to create a database of fingerprints before the system is usable. CLAC, on the other hand, utilizes crowdsourcing from users already validated to learn how to classify future users. Specifically, CLAC will learn over time how to classify users by building a training set of valid fingerprints. Initially, only users in the shaded circle in the immediate vicinity of the AP will be allowed access. This effectively bootstraps the system, allowing users in direct proximity of the AP to populate the training set. As these users move around the area and additional users are validated via an access code or the direct proximity test, more fingerprints will be added to the training set, allowing the algorithm to learn.

C. Contributions

In this work, we make the following contributions:

- Based on extensive indoor and outdoor measurements and observations, we propose two CSI metrics: CSI Cross-Antenna Stability Metric and CSI Cross-Frame Stability Metric, which serve well to summarize the spatial and temporal CSI characteristics respectively.
- We design CLAC, a zero configuration scheme which crowdsources valid users' CSI and RSS information to populate a training set, which is fed into a machine learning algorithm to classify future users in a given area.
- CLAC determines whether a user is in direct proximity of an AP by leveraging the rapid de-correlation of CSI between multiple antennae as the multipath effect gets worse, which, for example, may be caused by increased distance or presence of obstacles between the user and the AP.
- We develop a complete working implementation of CLAC on commercial-off-the-shelf devices and validate its usefulness through experimentation. Results demonstrate the effectiveness of CLAC's crowdsourcing mechanism to allow most valid users to access the network, while the chance that invalid users (outside the room) may access the network is small.

D. Paper Organization

The remainder of this paper is organized as follows: We offer some exploratory observations and analysis to motivate our design in Section II, and describe the design in Section III. Section IV describes the implementation of our system, and Section V shows the results of evaluation. Section VI briefly surveys related work in this area. We conclude in Section VII.

II. Observations and Design Ideas

A. Observations

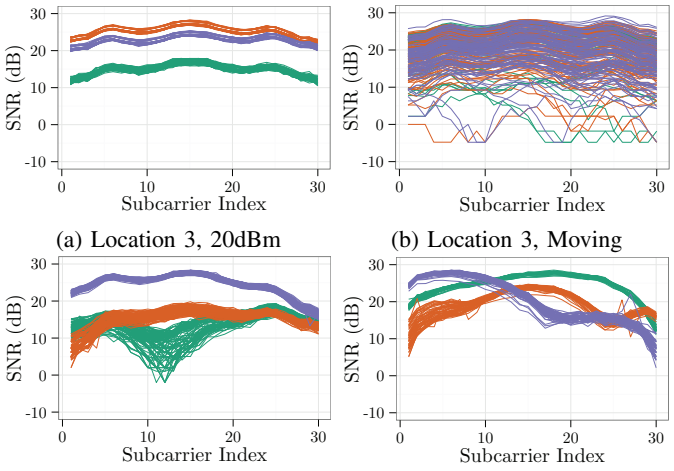
A few brief observations regarding CSI and RSS will help to explain the motivation and intuition behind CLAC.

1) CSI: CSI variance correlates with multipath effect and movement status. To observe this, we conduct a simple experiment with a single AP and two users at Locations 3 and 12 according to the topology given in Figure 1. The AP records the CSI of every frame from each user, including SNR values across each antenna and subcarrier.[1] The results are shown in Figure 2. The SNR values for Location 3 in Figure 2a are very high for each antenna and subcarrier, with little variance. Further, there is a strong similarity between the curves for each antenna. This tells us that we have a strong, stable connection, and the user at Location 3 is likely close to the AP.

The user at Location 12 is a bit further away. As a result, we notice in Figure 2c that there is greater separation between the lines making up each antenna indicating additional variance, and informing us that Location 12 is not directly at the AP. Additionally, the overall shape of each line is not consistent between antennae, showing little correlation.

We also study the effect of movement. We repeat the experiment for Location 3, this time moving around in a small area around the original point. We can clearly see in Figure 2b that there is a high degree of variance between each frame. Thus, we can learn that CSI variance relates closely to movement status.

Finally, we inspect the effect of a high-powered antenna. We repeat the experiment for Location 12, replacing the standard antenna with a high-powered antenna. We notice that the curves within each antenna are more similar, but each antenna has a very distinct curve. In this way, we recognize that the variance within CSI data correlates strongly to proximity, regardless of the transmitting power used. Thus, when comparing CSI values, CLAC uses a metric that measures stability and variance, as opposed to absolute values.



(a) Location 3, 20dBm (b) Location 3, Moving

(c) Location 12, 20dBm (d) Location 12, 30dBm

Figure 2: CSI collected by the AP from different sources indicated in Figure 1. Clearly, the most stable connections involve stationary, proximal sources.

[1] In the experiments, we use Thinkpad T400 with Intel 5300 NIC as the AP and montiors. It has three antennae and reports CSI of 30 subcarriers on each antenna.

2) RSS and Multiple Receivers: Comparing with CSI data, RSS values correlate better with the distance. To explore this, we study four users at Locations 3, 12, 13, and 19. We then send a number of packets (∼200) from the users and record the RSS as measured by the AP. The results of this experiment are shown in Figure 3. As we can see, closer locations have higher absolute RSS values. Therefore, it seems plausible that an easier classification technique would be to simply compare the received RSS to some threshold. However, such a technique is easily circumvented with a high-powered antenna. In this case, a user outside the given area could use a high transmitting power such that the RSS measured by the AP exceeds the threshold. If we raise the threshold to counteract this, we may prevent legitimate users with a normal transmitting power from accessing the network.

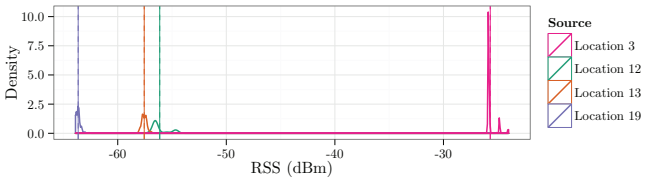


Figure 3: RSS densities for packets from various location to the AP. Location 3 is closest to the AP, and Location 19 is furthest. Dashed horizontal lines are RSS means.

To deal with this, CLAC uses multiple receivers to filter out invalid users by adding additional monitoring devices. Consider an example where a user at Location 19 is equipped with a high-powered antenna, such that it appears to be close to the AP at Location 3. If we also collect RSS readings at Monitor $M2$, we would think that the user was also close to the monitor due to the high-powered antenna. However, this would be inconsistent with a valid user at Location 3, which would have a low RSS value at Monitor $M2$, and we could then reject this user. Due to this intuition, CLAC uses readings from multiple locations to learn about valid users. Adding additional monitors can further increase system confidence.

3) Crowdsourcing: Many current localization schemes rely on extensive training sets acquired through war-driving before the system is operational. CLAC, however, uses crowdsourced CSI and RSS data from valid users to build a training set in real-time. Specifically, we use RSS data of valid users to populate the training set, but only when the users are determined to be stationary based on their CSI data, thus preventing potentially noisy data from movemement to contaminate the training set. Through the use of machine learning, we are able to perform implicit training instead of costly explicit training.

B. Observation Summary and Design Ideas

We summarize the above observations with the following:

- CSI variance between antennae correlates well to proximity and multipath effect, but not necessarily to distance.
- CSI variance between successive frames correlates well to movement status.
- RSS values correlate to the distance better than CSI, but may need multiple receivers to increase confidence and avoid ambiguity. It also needs CSI, to overcome the mobility situations.

Based on the above observations, we seek to design CLAC with the following ideas:

- Our scheme should use both CSI and RSS cooperatively to properly classify users.
- The scheme should make use of multiple APs or monitors to improve classification reliability.
- CLAC should utilize crowdsourcing to learn how to classify users in real-time, rather than rely on extensive pre-determined training sets.

III. System Design

A. System Overview

The design of CLAC is outlined in Figure 4. The system begins when a user attempts to join the network, at which point the system records the user's MAC address. Initially, this MAC address is not included in the Access List, and thus the default status of *Denied* is given to the user, such that he does not have Internet access. At this point, the user's RSS and CSI data is collected and fed into a model which outputs a binary indicator I whether the user should be classified as a valid user. If $I = 0$, access is denied and the user is informed of the decision and instructed to either move closer to the center of the store, or to acquire an access code from an employee. If $I = 1$, the user is accepted, the user's MAC is added to the Access List and their status is officially set as *Allowed*, such that they can start receiving Internet access. Finally, we also run periodic updates of each user's RSS and CSI data. As long as the Access List is non-empty, we collect RSS and CSI data for each admitted user and update the training set. In this manner, our training set will always contain the most recent data available.

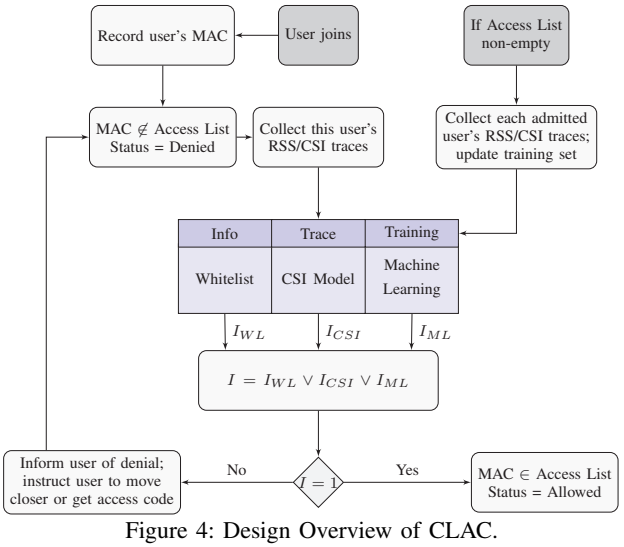


Figure 4: Design Overview of CLAC.

The core of CLAC is a set of three parallel prediction components, where each determines an indicator whether a

particular user is valid. The three components are a Whitelist, a CSI Model, and a Machine Learning algorithm. The results of all components are then combined as:

$$I = I_{WL} \vee I_{CSI} \vee I_{ML}, \tag{1}$$

where each I is the indicator of positive classification for each component. Three components can be briefly summarized as:

1) *WhiteList* - The Whitelist component allows a user to be specifically counted as a valid user.
2) *CSI* - The CSI component uses CSI stability metric to determine if a user is in direct proximity of an Access Point. It effectively bootstraps the system, allowing users that are clearly in the area to initially populate the training set for machine learning. As these valid users move around the area, CLAC can use the information to learn.
3) *Machine Learning* - The Machine Learning component utilizes crowdsourcing to allow additional users to gain access. This allows valid users (as determined by other components) to quickly populate the training set, allowing future users to be immediately validated.

The remainder of this section will explain each component in greater detail.

B. Whitelist Component

The Whitelist component simply outputs an indicator I_{WL} to signify whether or not a user has been added to a whitelist. If a user enters an access code during any point of the process, we know that he is a valid user of the system and should be given access, and thus they are added to a whitelist. This component will then output $I_{WL} = 1$, resulting in $I = 1$ and effectively overriding the other components. If the user is not on the whitelist, then $I_{WL} = 0$.

C. CSI Component

The CSI component outputs an indicator I_{CSI} whether a user is in direct proximity of an AP, and thus in the target area. To do this, we use a simple CSI model based on data collected at the AP. In an OFDM system, there are multiple streams of CSI information, with 30 subcarriers in each stream. Each data point is the received SNR value and is recorded as $H_j^{i,k}$, which represents the j-th frame collected by the i-th antenna on the k-th subcarrier. A group of each SNR values across all subcarriers for one frame j on a single antenna i ($i = 1, 2, 3$) can be represented as:

$$CSI_j^i = \langle H_j^{i,1}, H_j^{i,2}, \ldots, H_j^{i,30} \rangle. \tag{2}$$

During one measurement period (a system parameter with a default value of 10 seconds), the AP receives N frames, thus $j = 1, 2, \ldots, N$. Each frame has three curves (SNR vs subcarrier), one for each antenna.

1) CSI Cross-Antenna Stability Metric: According to our observations, when a user is stationary and close enough to an AP without obstacles between them, the corresponding CSI values on different antennae will be similar and stable, as in Figure 2a, where each line in a given color (antenna) represents a single frame. Thus, we propose a CSI cross-antenna stability metric to measure a user's proximity to the AP. It is calculated as follows. We first measure the correlation of CSI values between antennae using the Pearson correlation coefficient, as in:

$$CSI_{corr,j}^{a,b} = \frac{\text{cov}(CSI_j^a, CSI_j^b)}{\sigma_{CSI_j^a} \cdot \sigma_{CSI_j^b}}, \forall a, b \in \{1, 2, 3\}, \tag{3}$$

where cov is the covariance between CSI_j^i readings, and σ is the standard deviation of the respective values. This yields three correlation coefficients, one for each combination of antenna pair. We then use the minimum of these values as our final CSI cross-antenna stability metric for a given frame:

$$CSI_{CA-corr,j} = \min_{\forall a,b \in \{1,2,3\}} CSI_{corr,j}^{a,b}. \tag{4}$$

As an example, we consider a user at Location 3 which is very close to the AP. We previously showed all frames from Location 3 to the AP in Figure 2a, where all lines are very similar. If we calculate the correlation coefficient between all antenna pairs for all frames, we find that $CSI_{CA-corr}$ is at least 0.89, which is very high.

To determine a conservative cutoff for which we are confident that a $CSI_{CA-corr}$ value belongs to a valid user, we perform a series of short experiments. We collect CSI and distance from various locations around the room, and plot $CSI_{CA-corr}$ versus distance in Figure 5. The results show that $CSI_{CA-corr}$ drops off rapidly once we move further than one meter from the AP or outside the room. As a result, we use a threshold of 0.5 to determine if a user is in direct proximity of an AP.

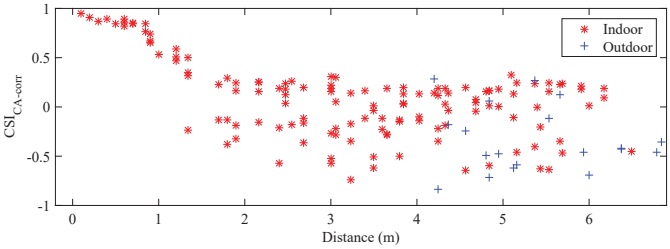


Figure 5: $CSI_{CA-corr}$ versus distance for locations around the room. Correlation drops off rapidly after one meter, signifying that 0.5 is a suitable threshold for determining direct proximity.

We also collect CSI from four locations with various transmitting power, and plot $CSI_{CA-corr}$ versus transmitting power in Figure 6. It shows that transmitting power has little effect on $CSI_{CA-corr}$. Although a high RSS value also may indicate a short distance between user and AP, it also may be caused by a high transmitting power at the user side. Thus, $CSI_{CA-corr}$ seems to be a better choice than RSS as the metric to measure a user's proximity to the AP.

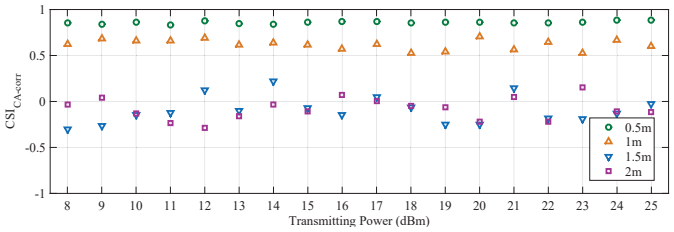


Figure 6: $CSI_{CA-corr}$ versus transmitting power for four locations at different distances from the AP.

2) User Recognition: When a user connects to an AP, the AP will measure the CSI and give a $CSI_{CA-corr}$ for each frame according to the above equations. Intuitively, we expect $CSI_{CA-corr}$ to be large (close to 1) when the user is near the AP. We calculate the average $CSI_{CA-corr}$ of frames in a measurement period, and use it to determine whether a user is close to the AP as follows:

$$I_{CSI} = \begin{cases} 1, & \text{if } \frac{\sum_{j=1}^{N} CSI_{CA-corr,j}}{N} > 0.5, \\ 0, & \text{otherwise,} \end{cases} \quad (5)$$

where N is the number of frames in the measurement period.

We continue the example with the user at Location 3, and look at all frames shown in Figure 2a. In this case, $CSI_{CA-corr}$ is above the threshold for every frame, and thus $I_{CSI} = 1$. We further examine a user at Location 12, who is not immediately near the AP as shown in Figure 1. As seen from the CSI information from Figure 2c, the shapes of the lines are not similar; thus, the $CSI_{CA-corr}$ values are low and $I_{CSI} = 0$.

D. Machine Learning Component

The Machine Learning component outputs an indicator I_{ML} whether a user is in the target area according to calculations by a machine learning algorithm. We use the One-Class Support Vector Machine (OSVM) algorithm, where our one class is valid users. The whole process can be described in the following steps.

1) Training Set: Selected RSS data from valid users is used to populate the training set. Each sample consists of the time when the frame is received, the sender's MAC address, and the RSS data from the AP and monitors. We represent the training set as

$$X = \{x_1, x_2, \ldots, x_n\}, \quad (6)$$

where n is the total number of samples in the training set, and each sample is described by the RSS as

$$x_j = \langle RSS_j^{AP}, RSS_j^{mon\_1}, RSS_j^{mon\_2}, \ldots \rangle. \quad (7)$$

2) Training Set Maintenance based on CSI Cross-Frame Stability Metric: CLAC uses crowdsourced information from users to build a training set in real-time. The following three conditions are both necessary and sufficient for data to be added to the training set:

- *Status = Allowed*: The user must be considered as valid.
- $t \in [t_c - W_T, t_c]$: The data must be within a training time window W_T, and t_c is the current time. This ensures that our scheme can recover from the training set, if contaminated, within the time window W_T. In practice, we use a default W_T value of two hours, such that we only use data from the last two hours.
- *Stationary*: The user must be stationary, which minimizes potentially noisy data from movement to be added to the training set. To determine if the user is moving or not, we leverage the rapid spatial de-correlation of CSI as in [1]. Specifically, we use the correlation coefficient between successive CSI samples on the same antenna to determine user movement. Since de-correlation occurs in all antennae, we define $CSI_{CF-corr}$ as the maximum cross-frame correlation coefficient among all antennae, and use it as the stability metric to determine whether a user is moving. It is calculated as:

$$CSI_{CF-corr,j} = \max_{\forall a \in \{1,2,3\}} \frac{\text{cov}(CSI_j^a, CSI_{j+1}^a)}{\sigma_{CSI_j^a} \cdot \sigma_{CSI_{j+1}^a}}. \quad (8)$$

The reason for taking the maxiumum is based on the consideration that it is unlikely for any antenna to maintain a high cross-frame correlation if a user is moving. We plot $CSI_{CF-corr}$ values of a user in Figure 7. It is clear that $CSI_{CF-corr}$ stays high (close to 1) when the user is stationary (during the shaded regions). In CLAC, we start to add the RSS data of a valid user into the training set, if the average $CSI_{CF-corr}$ value of 20 consecutive frames is above 0.5, and stop including it if any $CSI_{CF-corr}$ value drops below 0.5.

These conditions ensure that we only use data from valid users that are stationary.

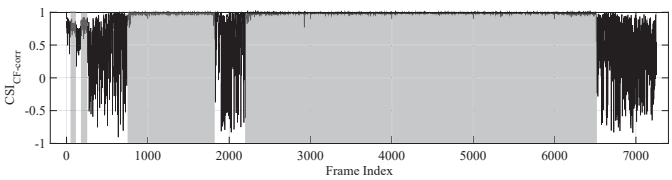


Figure 7: $CSI_{CF-corr}$ versus frame index. A high and stable value signifies that there is high correlation between successive samples, and thus the user is likely stationary, as indicated by shaded regions.

3) One-Class SVM Classification: CLAC uses a One-Class SVM (OSVM) classifier to determine if each frame matches the current training set. In CLAC, OSVM detects invalid users by finding a proper hypersurface in the non-linear space, which we find by starting with:

$$(\omega, \xi, b) = \arg\min \left(\frac{1}{2}\|\omega\|^2 - b + \frac{1}{vn}\sum_{j=1}^{n} \xi_j \right) \quad (9)$$
$$\text{s.t. } \forall \xi_j \geq 0,\ \omega \cdot \phi(x_j) - b \geq \xi_j,$$

where $\phi(x_j)$ is a kernel mapping [2] of x_j, ξ is the slack variable, which enables the model with some tolerance, and the constant v is set to 0.01 to control the tolerance. Among all the

training samples, the frame x_j^* subject to $(\omega \cdot \phi(x_j^*) - b = \xi_j)$ is the so-called support vector, which is located on the margin of the classifier. By solving the problem we can get a classifier:

$$f(x) = \text{sgn}(\omega \cdot \phi(x) - b). \quad (10)$$

The case $f(x) = 1$ means that the frame x comes from a valid user; otherwise $f(x) = -1$. For the support vector x_j^*, $f(x_j^*) = 0$. In our application, when a user device attempts to join the network, it sends out multiple frames in a measurement period. We calculate the average RSS value of the frames when the user is determined to be stationary based on $CSI_{CF-corr}$, and uses it as the input to the Machine Learning component. Then, the Machine Learning component outputs an indicator I_{ML} to indicate whether the user is valid.

IV. Implementation Details

We provide a working implementation of CLAC in order to fully evaluate our scheme. The CLAC implementation consists of two complementary components: *CLAC Access Point and Monitor*, and *CLAC Server*. As depicted in Figure 8, in addition to the normal operations of serving data to users through Wi-Fi and bridging, CLAC has introduced a few custom function blocks. The custom function blocks are indicated with shaded boxes in the figure, while the white boxes indicate non-CLAC specific operations.

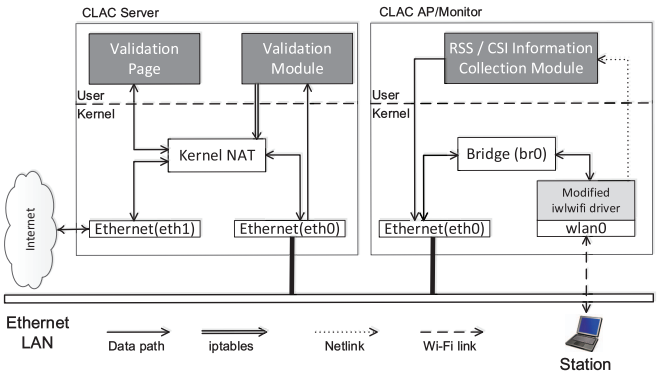


Figure 8: System Architecture: The box on the left shows the CLAC server to coordinate all decisions, and the box on the right shows the CLAC AP/monitor to collect information and provide network access.

A. CLAC AP and Monitor

The CLAC AP provides Wi-Fi service to users using a modified version of the `hostapd` software. The AP, along with passive CLAC monitors, also collects RSS and CSI information using a modified `iwlwifi` driver by employing the Linux 802.11n CSI Tool [3]. Part of our modifications include adding support to the `iwl_priv` structure to collect the MAC address of the user in addition to the RSS and CSI information. The information is buffered in the kernel process and then transmitted to a user space process, called the *RSS/CSI Information Collection Module*, via a `netlink` socket. The collected information is stored locally in a file, and then sent to the CLAC server through the Ethernet communication backbone. The file is updated periodically to ensure that the latest data is available.

B. CLAC Server

The CLAC server is implemented on a router that provides the DHCP service and connects Ethernet to the Internet. The key function block added to the router is the *Validation Module*, which is responsible for collecting data from all CLAC APs and monitors to provide a final classification decision for each user. The server primarily uses `iptables` to control access for connected users. Valid users will be assigned a valid IP address by the DHCP server such that they can access the Internet, while other users will be directed to a local website which informs them their validation has failed, along with a recommendation to either move closer to the center of the store or to acquire an access code.

V. Evaluation Results

A. Experimental Setup

We evaluate CLAC using experiments. In the experiments, we use three Lenovo ThinkPad T400 laptops as the AP and monitors, as shown in Figure 9. Each laptop runs a 32-bit Ubuntu Server 10.04 (LTS) operating system, and is configured with the Intel 5300 802.11n wireless network card. The AP (or monitor) uses interface `wlan0` operating on Channel 6 of the 2.4GHz band to communicate with users, and interface `eth0` to communicate with the server. The server uses `eth1` to connect to the Internet. We set up a testbed in a $6.5m \times 8m$ laboratory with 8 work cubes and 8 tables as shown in Figure 1. The laboratory has a concrete wall of 0.5 meters of thickness, and the door is open throughout the experiments.

Figure 9: Thinkpad T400 with Intel 5300 NIC, which has three antennae and reports CSI of 30 subcarriers on each antenna.

We then collect the CSI and RSS data from the smartphones of 30 different users as traces. The path followed for each trace is shown in Table I. Each user (except user 26), enters with 20 $\rightarrow$ 19 $\rightarrow$ 18 $\rightarrow$ 17 before following the listed path, and then leaves with 17 $\rightarrow$ 18 $\rightarrow$ 19 $\rightarrow$ 20. User 26 merely travels 20 $\rightarrow$ 19 $\rightarrow$ 18 $\rightarrow$ 17 $\rightarrow$ 18 $\rightarrow$ 19 $\rightarrow$ 20. All users walk at a speed of 1 m/s and dwell at the final location listed in the table for between 20 and 30 minutes. To simulate network traffic, each user sends `ping` packets with an interval of 0.2 seconds. Furthermore, in CLAC, the OSVM classifier is recalibrated every 5 minutes with the updated training set.

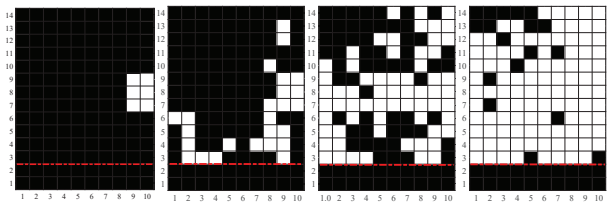
Table I: Paths of various users. All users enter with 20 → 19 → 18 → 17, and leave in the opposite order, except for User 26 who only travels as far as Location 17. All users also stay at the final location listed for the majority of their time.

User	Arrival (hrs)	Path	User	Arrival (hrs)	Path
1	0.06	9 → 3 → 9	16	1.49	3 → 15
2	0.08	3 → 13	17	1.53	3 → 11
3	0.10	3 → 5	18	1.58	3 → 4
4	0.25	3 → 10	19	1.72	3 → 12
5	0.29	3 → 7	20	1.80	3 → 16
6	0.53	3	21	2.04	3 → 4
7	0.80	3 → 5	22	2.14	3 → 2
8	0.84	3 → 16	23	2.21	3 → 12
9	0.85	3 → 13	24	2.28	3 → 7
10	0.91	3 → 8	25	2.28	3 → 11
11	1.05	3	26	2.33	dwells at 17
12	1.21	3 → 1	27	2.58	3 → 14
13	1.26	3 → 10	28	2.40	3 → 2
14	1.28	3 → 6	29	2.85	3 → 13
15	1.38	3 → 8	30	2.85	9

B. Single Trace Results

Initially, we play back the traces in the order listed, with arrival times following a Poisson process for a total simulation time of 3 hours. We use a training window (W_T) of 2 hours, and a measurement period of 10 seconds.

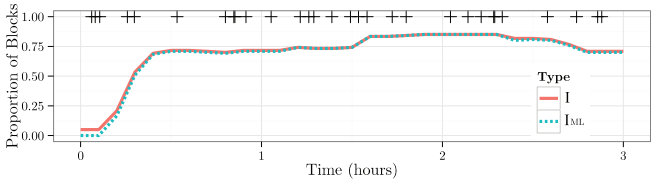
We partition up the area into 140 $60cm \times 60cm$ blocks, as shown in Figure 10, with 120 blocks inside the room (top 12 rows above the dotted line), and 20 blocks outside the room (bottom two rows below the dotted line). We show the validation results at each test block at 0, 10, 20, and 120 minutes. Initially, the CSI model outputs indicator $I_{CSI} = 1$ near the AP, allowing the entire scheme to output indicator $I = 1$. This effectively bootstraps the system for the Machine Learning component to begin working, which learns of valid locations as users travel around the room over time. As a result, the Machine Learning algorithm, and thus the entire algorithm, output indicator $I = 1$ for an increasing number of blocks. After two hours, nearly the entire room is correctly identified. Additionally, the 20 blocks outside the room always have $I = 0$. This confirms that CLAC is able to correctly identify the boundary precisely around the room.



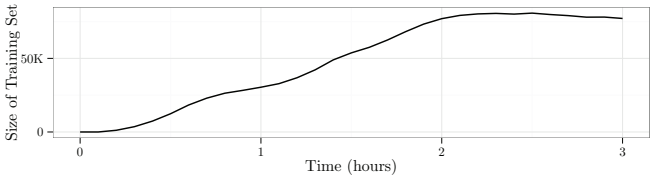
(a) 0 min (b) 10 mins (c) 20 mins (d) 120 mins

Figure 10: Validation results for each test block. White color signifies $I = 1$ and black color represents $I = 0$.

Given the blocks above, we show the proportion of blocks with $I_{ML} = 1$ and $I = 1$ in Figure 11a. We also show the size of the training set over time in Figure 11b. In both cases, we see that the blocks with indicator $I = 1$ increases along with the size of the training set. Specifically, the proportion increases very rapidly at the beginning of the simulation, as many users are dwelling in new locations. After the initial period, the proportion remains relatively stable until new locations are visited around 1.5 hours. Finally, after about two hours, the proportion stabilizes since the training window W_T is two hours.

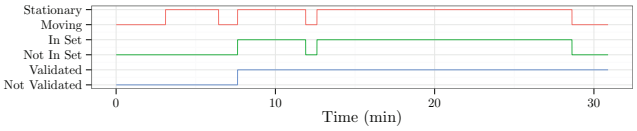


(a) Proportion for both I and I_{ML}. The black ticks represent user arrival times.

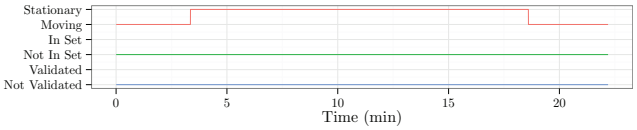


(b) Training Set Size

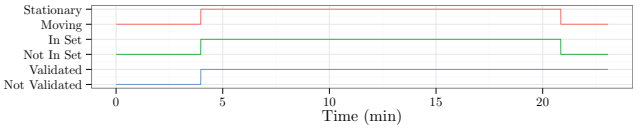
Figure 11: Additional results with 30 users in given order.



(a) User 1



(b) User 26



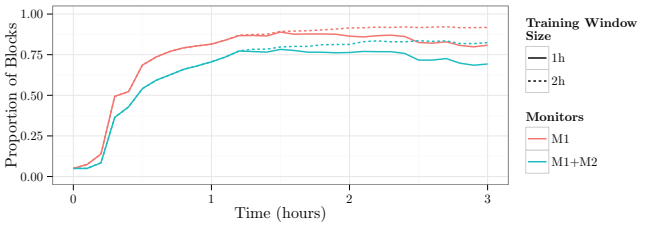
(c) User 30

Figure 12: Detailed status of three users.

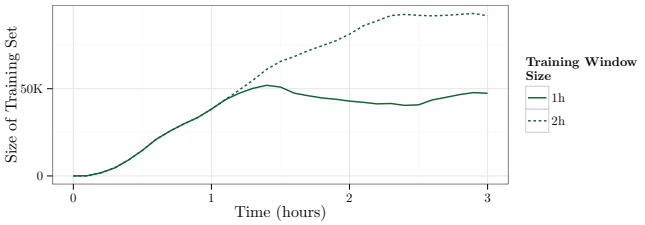
We now look at detailed information from three users as an example in Figure 12. At first, User 1 sits at Location 9, but is not validated since the training set is initially empty. User 1 then walks to the AP to be validated by the CSI model, and then returns to Location 9 where they can begin using the network. User 26 simply stands at Location 17 and never enters the room, and is thus never validated. Finally, User 30 sits at Location 9, but is classified as a valid user immediately due to the Machine Learning component.

C. Aggregate Trace Results

We now repeat the previous simulation multiple times, but each time with a randomized order of users (except User 1



(a) Proportion of blocks validated for various conditions.



(b) Training Set Size for various conditions. The size grows much larger with a longer training window.

Figure 13: Aggregate results for multiple simulations with random user orders.

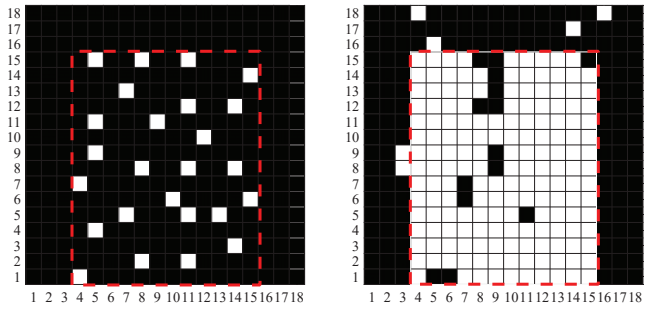
to start and User 30 to end), following the paths specified in Table I. This allows us to observe expected results for an arbitrary ordering of users. We can then alter parameters such as which monitors are used, and the length of the training window. We show the results in Figure 13.

We first notice that using an AP together with one additional monitor results in a higher proportion of validated blocks, as opposed to two additional monitors. When two monitors are used, a particular block must have appropriate CSI and RSS values at both monitors to be correctly identified by the Machine Learning component, thus providing a more stringent challenge. As a result, using a single additional monitor performs very well, which allows CLAC to be used with minimal infrastructure overhead.

We also notice that increasing the size of the training window allows for the proportion of blocks found to remain high over time, as the size of the training set is much higher. If we use a short training window such as one hour, then after an hour we stop using traces from the earliest users, and no longer validate the blocks they travelled through. However, if the environmental conditions change rapidly, using a smaller training window size would be beneficial, as older users with values from old environmental conditions would no longer contribute to the training set. In this manner, we can tune CLAC according to the conditions present.

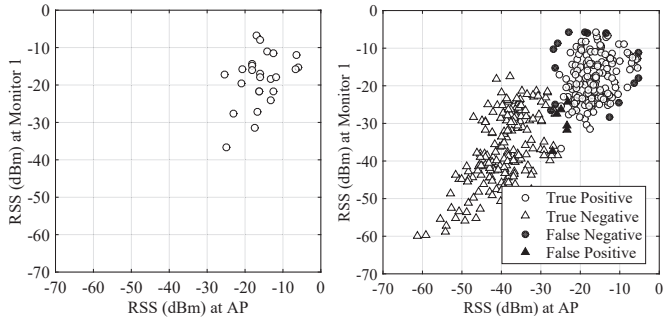
D. Additional Results on Classification Accuracy

We have conducted additional experiments in a larger $7.2m \times 9m$ classroom (also with a concrete wall of 0.5 meters of thickness) to further evaluate the performance of CLAC's Machine Learning component. We partition up the area into 324 $60cm \times 60cm$ blocks, with 180 blocks inside the wall (shown as the dotted line in Figure 14) and 144 blocks outside the room.



(a) Training set (b) Classification results

Figure 14: Additional evaluation results in a classroom. Dotted line represents the wall. White color signifies $I_{ML} = 1$ and black color signifies $I_{ML} = 0$.



(a) Training set (b) Classification results

Figure 15: Average RSS values from various blocks to the AP and the monitor.

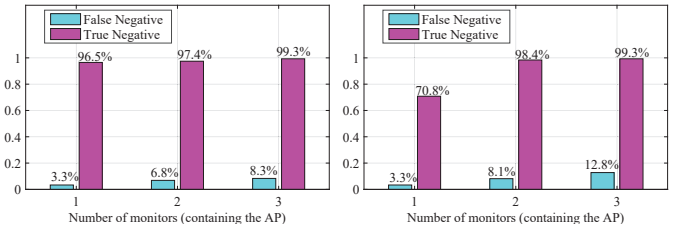
One AP is set up at the center of the room, and one monitor is set up along the wall. Figure 14a shows the locations where valid users sit and we use their RSS data to form a training set. Figure 14b shows the output of the Machine Learning component at all blocks. Results show that, with a reasonable-size training set, CLAC is able to produce correct classification results most of the time. Figure 15a plots the average RSS values of each valid user in the training set. Figure 15b plots the RSS values collected in each test block and the corresponding classification result:

- "○" ("●") – True Positive (False Negative): a test block inside the room has been classified correctly (incorrectly);
- "△" ("▲") – True Negative (False Positive): a test block outside the room has been classifid correctly (incorrectly).

Ideally, we would like to have 0% false negative and 100% true negative rates. In our experiments, as shown in the figure, CLAC yields about 7.8% false negative rate and 95.8% true negative rate; only very few isolated or boundary points were misclassified.

We also vary the number of monitors (counting the AP) to explore how it may affect CLAC's classification performance. Figure 16a shows the true positive and false positive rates when the AP is at the center of the room, while Figure 16b shows the results when the AP is along the wall. All montiors are set up along the wall. Results show that CLAC yields consistently low false negative and high true negative rates

with the AP at the center. However, when the AP is along the wall, at least one additional monitor is needed to aid the AP to produce better classification results.



(a) AP in the center (b) AP in the boundary

Figure 16: Average false negative and true negative rates with different numbers of monitors (counting the AP).

VI. Related Work

Numerous indoor localization schemes have been proposed in the past. Recently proposed solutions include [4]–[10]. In order to improve localization accuracy, they either require additional infrastructure support such as a large number of APs [4], or knowledge of building or floor map [5], or cumbersome war-driving [6], or inertial sensor measurements [5]–[10], or explicit user cooperation such as spinning [9].

Different from indoor localization solutions, CLAC attempts to address a distinct but related problem, which is to accomplish precise Wi-Fi access control within a given area, while not requiring accurate localization of each user. In other words, a user is granted access to the network as long as the user is inside the given area, while the exact location of the user is less important. Therefore, compared with indoor localization schemes, CLAC is a lightweight solution which does not require expensive localization operations or supporting devices. It also is based on crowdsourcing which does not require explicit training to set up a fingerprint database *a priori*.

Recently, physical layer information such as CSI has been exploited to address various problems such as user authentication [11]–[18]. CLAC also is a CSI-based scheme. In fact, it exploits both CSI and RSS, and achieves the goal of precise Wi-Fi access control with a combination of CSI-based proximity model which is based on cross-antenna correlation, and CSI/RSS-based classification scheme which is based on cross-frame correlation and machine learning.

Geo-fencing is a system proposed in [19] that addresses a similar problem of ours. However, it requires multiple APs, each with an electronically steerable directional antenna to confine the Wi-Fi coverage. In comparison, CLAC does not require special hardware support and works well with off-the-shelf wireless devices equipped with omnidirectional antennae.

VII. Conclusions

With the numbers of smartphones and mobile devices constantly increasing, users are increasingly interested in connecting to the Internet via Wi-Fi while away from home and work. We present and implement CLAC, which enables a location to offer Wi-Fi access to customers inside a particular boundary, while preventing users outside the boundary to connect to the network. Specifically, CLAC first recognizes valid users within direct proximity of the AP by leveraging the rapid decorrelation of CSI between multiple antennae as the multipath effect gets worse which may be caused by increased distance or presence of obstacles between the user and the AP. Using this data, we then utilize crowdsourcing to build a training set from valid users, based on which to recognize additional valid users over time. As a result, CLAC is able to successfully recognize valid users, while denying access to invalid users. Future work includes adding better boundary detection for varied settings and environments.

Acknowledgment

This work is supported by Natural Science Foundation of China under Grants No. 61272524 and by Dalian University of Technology under the Haitian Scholar grant. Work by Michael Segal has been supported by Israel Science Foundation (grant No. 317/15), by IBM Corporation and by Israel Ministry of Economy and Industry. Linlin Guo, Jialin Liu, and Wei Zhou are participated in this work.

References

[1] Z.-P. Jiang, W. Xi, X. Li, S. Tang, J.-Z. Zhao, J.-S. Han, K. Zhao, Z. Wang, and B. Xiao, "Communicating is crowdsourcing: Wi-Fi indoor localization with CSI-based speed estimation," *Journal of Computer Science and Technology*, vol. 29, no. 4, 2014.

[2] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Computation*, vol. 13, no. 7, 2001.

[3] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, "Tool Release: Gathering 802.11n traces with channel state information," *ACM SIGCOMM CCR*, vol. 41, no. 1, 2011.

[4] S. Yang, P. Dessai, M. Verma, and M. Gerla, "FreeLoc: Calibration-free crowdsourced indoor localization," in *IEEE INFOCOM*, 2013.

[5] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan, and R. Sen, "Zee: Zero-effort crowdsourcing for indoor localization," in *ACM MobiCom*, 2012.

[6] A. T. Mariakakis, S. Sen, J. Lee, and K.-H. Kim, "SAIL: Single access point-based indoor localization," in *ACM MobiSys*, 2014.

[7] H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, and R. R. Choudhury, "No need to war-drive: Unsupervised indoor localization," in *ACM MobiSys*, 2012.

[8] Z. Yang, C. Wu, and Y. Liu, "Locating in fingerprint space: Wireless indoor localization with little human intervention," in *ACM MobiCom*, 2012.

[9] S. Sen, R. R. Choudhury, and S. Nelakuditi, "SpinLoc: Spin once to know your location," in *ACM HotMobile*, 2012.

[10] C. Luo, H. Hong, and M. C. Chan, "PiLoc: A self-calibrating participatory indoor localization system," in *IEEE IPSN*, 2014.

[11] J. Tugnait and H. Kim, "A channel-based hypothesis testing approach to enhance user authentication in wireless networks," in *IEEE International Conference on Communication Systems and Networks (COMSNETS)*, 2010.

[12] J. Tugnait, "Wireless user authentication via comparison of power spectral densities," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 9, 2013.

[13] L. Xiao, L. Greenstein, N. B. Mandayam, and W. Trappe, "Using the physical layer for wireless authentication in time-variant channels," *IEEE Transactions on Wireless Communications*, vol. 7, no. 7, 2008.

[14] ——, "Channel-based spoofing detection in frequency-selective rayleigh channels," *IEEE Transactions on Wireless Communications*, vol. 8, no. 12, 2009.

[15] N. Goergen, W. Lin, K. Liu, and T. Clancy, "Extrinsic channel-like fingerprint embedding for authenticating MIMO systems," *IEEE Transactions on Wireless Communications*, vol. 10, no. 12, 2011.

[16] Y. Liu and P. Ning, "Enhanced wireless channel authentication using time-synched link signature," in *IEEE INFOCOM*, 2012.

[17] D. Shan, K. Zeng, W. Xiang, P. Richardson, and Y. Dong, "PHY-CRAM: Physical layer challenge-response authentication mechanism for wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 9, 2013.

[18] Z. Jiang, J. Zhao, X.-Y. Li, J. Han, and W. Xi, "Rejecting the attack: Source authentication for Wi-Fi management frames using csi information," in *IEEE INFOCOM*, 2013.

[19] A. Sheth, S. Seshan, and D. Wetherall, "Geo-fencing: Confining Wi-Fi coverage to physical boundaries," in *Springer Pervasive, LNCS*, vol. 5538, 2009.