RICE UNIVERSITY

Golden-Eye: A Server-Side Location-Sensing System for Wireless LANs

by

Ping Tao

A THESIS SUBMITTED
IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE

Master of Science

APPROVED, THESIS COMMITTEE:

Dan S. Wallach, Chair, Assistant Professor
Computer Science

David B. Johnson, Associate Professor
Computer Science

Edward W. Knightly, Associate Professor
Electrical and Computer Engineering

HOUSTON, TEXAS

FEBRUARY, 2004
INFORMATION TO USERS

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleed-through, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.
Golden-Eye: A Server-Side Location-Sensing System for Wireless LANs

Ping Tao

Abstract
Determining the location of a wireless client is a key problem for location-aware systems and for security applications. Many recent studies have used Bayesian methods to determine location from wireless LAN signals, but such methods have the drawback that a model must first be built from training data. The introduction of model error can drastically reduce the robustness of the location estimates, rendering most models incapable of tolerating hardware differences, channel variations, and intentional interferences from malicious users. This thesis describes the design, implementation and analysis of Golden-Eye, a server-side wireless LAN location-sensing system that uses new techniques to address this problem. By fitting training data into Gaussian distributions and using relative signal strength, Golden-Eye works independent of the client’s wireless LAN implementation or transmission power level, making it suitable even for tracking clients that might be trying to hide their locations.
Acknowledgments

I would like to thank my advisor Dr. Dan S. Wallach for his guidance throughout this thesis work. His support and criticism were among the most valuable resources I had during this work. I would also like to thank other members of my thesis committee, Dr. David B. Johnson and Dr. Edward W. Knightly, for their reviews and suggestions that greatly improved the quality of this thesis. Thank you also to Algis Rudys for his assistance in getting part of this work published in WiSE '03 [24], and to Shu Du for volunteering his time to set up some of the experiments. Last but not least, thanks to my family and friends for their continuous support and encouragement.
Contents

Abstract ii
Acknowledgments iii
List of Illustrations vi

1 Introduction 1
   1.1 Motivation ........................................ 1
   1.2 Contributions ..................................... 2
   1.3 Related Work ..................................... 2
   1.4 Organization ..................................... 4

2 Golden-Eye System Architecture 5
   2.1 System Overview ................................. 6
   2.2 Snoopers ......................................... 7
   2.3 Target Machine .................................. 9
   2.4 Server ........................................... 9

3 Observing RF Signals 10
   3.1 Channel Variations ............................. 10
   3.2 Transmission Power ............................. 14

4 Location-Sensing Algorithms 15
   4.1 Histogram Method ............................... 16
   4.2 Gaussian Method ............................... 16
   4.3 Difference Method .............................. 17
5 Results
5.1 Experimental Setup .................................................. 20
5.2 Basic Localization ...................................................... 24
5.3 Varying Transmission Power Level ................................. 26
5.4 Varying Transmitter ..................................................... 27
5.5 Untrained Locations .................................................... 27

6 Discussion ................................................................. 30
6.1 Methodology Revisited .................................................. 30
  6.1.1 Weighting Scheme vs. Probability ............................... 30
  6.1.2 Inference Window Size .......................................... 31
  6.1.3 Limitations .......................................................... 32
6.2 Comparison to Prior Work ............................................. 32
6.3 Privacy Concerns ....................................................... 33

7 Conclusion ............................................................... 35

Bibliography ............................................................... 36
Illustrations

2.1 Golden-Eye system components ........................................ 6
2.2 A segment of snooper trace data ........................................ 8

3.1 Observed signal measurements over time from a laptop to fixed AP1 ........................................ 11
3.2 Observed signal measurements over time from a laptop to fixed AP2 ........................................ 12
3.3 Observed signal measurements over time from a laptop to fixed AP3 ........................................ 13
3.4 Signal strength readings from four different receivers of signal from a single transmitter, with the transmitter varying its transmission power ........................................ 14

4.1 AP1, AP2, AP3 signal strength distribution fitting ........................................ 18

5.1 Experimental area map ........................................ 21
5.2 Training data in the long hallways ........................................ 22
5.3 Training data in the short hallways ........................................ 23
5.4 Cumulative error over the four hallways, where the target machine and WLAN card are the same as used in training ........................................ 24
5.5 Cumulative error over the four hallways, where the target machine and WLAN card are the same as used in training, but the transmission power level is reduced when localization occurs ........................................ 25
5.6 Cumulative error over the four hallways, where the transmission power level changes for every packet ........................................ 25
5.7 Cumulative error over the four hallways, where different WLAN cards were used for localization and training. ........................................ 26
5.8 Cumulative error of untrained poses ........................................ 29
5.9 Localization results outside the training area .......................... 29

6.1 The effect of varying inference window size on the localization accuracy ... 31
Chapter 1

Introduction

1.1 Motivation

IEEE 802.11a/b/g wireless LAN (WLAN) systems have been enthusiastically adopted in business offices, homes, hotels, cafés, and other spaces, both public and private, for wireless local network connectivity. The widespread deployment of WLANs has fostered a growing interest in location-aware systems and services [18, 19, 11]. However, WLAN itself does not provide the necessary localization primitives needed for location-aware applications. Over the years, a wide variety of techniques have been developed to address this need, both using custom hardware devices, such as sonar [20, 21] or infrared sensors [26], and using RF signal-strength measurements that can be performed by existing WLAN cards. The latter provides more ubiquitous coverage and does not require additional hardware for user location determination, thereby becoming more popular in recent studies [2, 1, 16, 30]. However, these localization systems are designed to work on the client-side or with client collaborations. They do not consider the need to locate WLAN intruders who desire not to be found, and thus will not participate in the localization process.

If an intruder connects to a traditional, wired network and begins transmitting packets, those packets can usually be physically traced to the port where they entered, and the rogue machine can be physically located and disconnected. In the wireless case, however, the network administrators only know that the rogue machine is associating with a given access point. This rarely provides enough information to physically identify the rogue machine. Moreover, wireless network technologies such as WLAN are subject to additional classes of attacks that may be exploited by such a rogue machine [3], making the identification of their physical location a priority for administrators.
In these situations, the localization must be performed by other agents in the system, such as customized access points. The localizing agents do not know what WLAN hardware the rogue machine is using, and they do not know the power level at which the rogue machine is transmitting. Indeed, the rogue machine could vary its transmit power for every transmitted packet. In order to track such a target, a location sensing system must be sufficiently independent of the differences between mobile device configurations and must be able to overcome active interference by the intruder designed to confound this system.

1.2 Contributions

This thesis presents a server-side in-building location-sensing system, Golden-Eye, that requires no hardware or software on the client being tracked other than an off-the-shelf WLAN card. By fitting training data into Gaussian distributions and using relative signal strength, Golden-Eye is robust against differences in the transmitter and modulation of transmission power levels by the intruder, making it suitable for tracking rogue machines. Furthermore, these techniques are robust against variations in the quality of the wireless channel, which is time varying due to the movements of the objects and the dynamics of the changing environments [10]. Training data are collected using WLAN cards with a customized driver capable of measuring the signal-power level, signal-to-noise ratio, and signal-to-interference ratio for every packet. I present experimental data suggesting that variations in hardware and transmission power can be handled without explicitly training for such variations.

1.3 Related Work

The field of location-aware computing [11, 6] deals with two principal tasks, first, determining and tracking the position of a mobile device, and second, providing useful user functionality that makes use of a location primitive. WLAN location sensing deals with executing the first task for devices such as laptops or PDAs equipped with wireless network
hardware.

The scheme presented in this thesis is an extension to the WLAN location sensing system previously developed at Rice University by Ladd et al [16]. This system uses the signal strength read off WLAN cards as a sensor and implements the Markov localization algorithm commonly used in various robotics applications [25, 8, 4]. Following this technique, conditional probability distributions are built correlating sensor readings to position space by sampling. This is an off-line phase referred to as the training or learning phase.

During the on-line phase, measurements are integrated and a probability distribution is built over position space. A maximum likelihood estimate is then used to determine position. A sequence of estimates can be integrated over time using various sensor fusion techniques. A Hidden Markov Model (HMM) can be used for this purpose [16, 14]. A general survey on probabilistic methods can be found in a comprehensive paper by Thrun [25]. Some interesting developments are discussed in a recent paper [9], which experimentally compares Kalman filtering, grid-based Markov localization, Monte Carlo localization, and combinations thereof.

Several early location-aware computing schemes used specialized hardware such as ultrasound transmitters or cameras to detect location [27, 21, 15]. Early schemes for wireless location sensing also relied on specialized transmitters or base stations [28, 12]. A number of systems have been built using probabilistic techniques to determine location based on RF signal strength for cellular telephone systems [17, 29]. The first system to use signal strength from off-the-shelf WLAN cards to detect location was RADAR [2, 1], which used nearest neighborhood techniques to infer the user location.

Recently, there has been a flurry of activity in testing if probabilistic localization techniques can be applied to WLAN-based location sensing [5, 16, 30, 23, 31]. At the core of these various methods is the construction of conditional probability distributions relating sensor values to positions. This relation is constructed during a training phase as previously described. All these studies reach similar conclusions, that robust 1–2 meter accuracy is achievable, probabilistic methods effectively combat noise, and it is difficult to automate
training and parameter tuning. Operator, hardware, and power variation is not discussed in any of these works.

The techniques proposed in this thesis involve filtering the raw input of the sensor before inputting the data into the localization engine. In mobile robotics localization applications, filters are commonly used to reject outliers, reduce noise, relativize and reduce data, or fit the data into some statistical model. Various filters can be employed as a pre-processing phase on sensors ranging from sonar to cameras for these purposes [7, 9, 25].

1.4 Organization

The rest of the thesis is structured as follows. In Chapter 2, I describe the architecture of the Golden-Eye location-sensing system. In Chapter 3, I present several observations of the RF signal measurements that helped the design of the robust location-sensing algorithms. In Chapter 4, I describe these algorithms in details. Then, I explain the results from tracking experiments in Chapter 5. Finally, I discuss these results and future work in Chapter 6, and present my conclusions in Chapter 7.
Chapter 2

Golden-Eye System Architecture

In prior work on WLAN localization [16], there was a mobile laptop measuring the signals from fixed access points (APs). An initial training phase took measurements at positions spaced approximately every 1.5 meters on the third floor of Duncan Hall, the building housing Rice University's Computer Science department. This training data was used to create a Bayesian network that could take subsequent observations of signal strengths and yield a probability distribution of where the mobile device might be in the building.

In contrast, Golden-Eye is a server-side location-sensing system, where signal strengths are measured by the wireless infrastructure and localization can take place without notifying the client. This is beneficial because the client does not have to download and run a localization program every time it enters a new network, and it does not have to waste CPU cycles and battery power on localization. On some devices, such as PDAs, processing power may be expensive. Of course, this also raises privacy concerns. See Chapter 6 for more discussion.

Mathematically, there is no difference between a laptop measuring the observed signal strengths of APs and APs collaboratively measuring the observed signal strength of a laptop. However, a server-side architecture allows Golden-Eye to localize a laptop independent of any specialized hardware or software on the laptop. In this way, it can be tested against situations where a laptop is using unknown hardware or adjusting its transmission power to evade detection.
Figure 2.1: Illustration of the server-side location-sensing system Golden-Eye. The snoope-
ers can monitor packet signal from any WLAN transmitter within their range. The Golden-
Eye server controls the snoppers, gathers signal property data and processes them during
training and location-sensing.

2.1 System Overview

A Golden-Eye location-sensing system consists of a centralized server and a number of
snoppers, as shown in Figure 2.1. Snoppers provide overlapping coverage for the target
area, much like APs do in a production network. In IEEE 802.11b [13] networks, one
AP provides services on only one of the 11 channels defined by the IEEE standard in the
2.4GHz band in the United States. Typically, neighboring APs use different channels to
minimize interference. A WLAN station associates with one of the APs that service its lo-
cation and communicates on the chosen AP’s channel. In order to measure signal strengths
of all stations within a snopper’s coverage area, the snopper is capable of switching to any
specific channel to take measurements.

During training and localization, the server notifies the snappers of the target MAC
address, the channel number, and the listening period. The snappers will then tune in to
the target channel, record the signal strength of all packets received that match the server’s
query, and transmit those results to the server, which can then infer the physical location of
the target using algorithms described in Chapter 4.

In the current prototype, the server communicates with the snappers using the preex-
isting in-building WLAN. In a future production environment, the snooper functionality might be integrated directly into existing WLAN APs.

2.2 Snoopers

Snoopers are responsible for observing the signal strength of packets transmitted by the target machine. In my experiments, I used five laptops from various manufacturers, all running Windows XP, and using D-Link Systems AirPlus DWL-650+ WLAN PCMCIA cards containing the Texas Instruments ACX100 single-chip IEEE 802.11b [13] WLAN implementation. I modified the device driver, allowing me to extract the signal strength, noise power, and multipath power of each received packet. Figure 2.2 shows some measurements collected by a snooper.

The signal strength readings are taken from the receiver’s automatic gain control (AGC) register. The controller updates this 8-bit register for every packet after receiving the packet’s IEEE 802.11b PLCP header [13]. As I understand the system, this value should stay constant, even if the transmitter changes to a different bit-rate or makes other low-level protocol changes. I do not know how the AGC register’s values map to the actual signal intensity.

The noise power and multipath power readings are also taken from baseband registers. These values can be used to calculate signal-to-noise ratio (SNR) and signal-to-interference ratio (SIR). I do not use these values for this research.

Upon request, a snooper temporarily goes to promiscuous mode, switches channel if necessary, measures the target station’s signal strength, and returns back to normal network operations. I use this technique to allow the snoopers to perform tracking and communicate with the central server through the same WLAN card.
<table>
<thead>
<tr>
<th>MAC Address</th>
<th>Seq #</th>
<th>Signal Strength</th>
<th>Noise Power</th>
<th>Multipath Power</th>
</tr>
</thead>
<tbody>
<tr>
<td>00-02-2D-2F-42-ED</td>
<td>E3D0</td>
<td>90</td>
<td>40</td>
<td>24</td>
</tr>
<tr>
<td>00-02-2D-21-9B-32</td>
<td>EBD0</td>
<td>132</td>
<td>40</td>
<td>24</td>
</tr>
<tr>
<td>00-02-2D-2B-86-3A</td>
<td>E210</td>
<td>52</td>
<td>40</td>
<td>24</td>
</tr>
<tr>
<td>00-02-2D-2F-85-3E</td>
<td>3410</td>
<td>47</td>
<td>40</td>
<td>24</td>
</tr>
<tr>
<td>00-02-2D-2F-42-3C</td>
<td>32B0</td>
<td>58</td>
<td>40</td>
<td>24</td>
</tr>
<tr>
<td>00-02-2D-2F-42-3F</td>
<td>DB20</td>
<td>106</td>
<td>40</td>
<td>24</td>
</tr>
<tr>
<td>00-02-2D-2B-82-4F</td>
<td>4550</td>
<td>81</td>
<td>40</td>
<td>24</td>
</tr>
<tr>
<td>00-02-2D-2F-42-ED</td>
<td>ED70</td>
<td>88</td>
<td>51</td>
<td>19</td>
</tr>
<tr>
<td>00-02-2D-21-9F-0B</td>
<td>B7A0</td>
<td>50</td>
<td>51</td>
<td>19</td>
</tr>
<tr>
<td>00-02-2D-21-9B-32</td>
<td>BDC0</td>
<td>104</td>
<td>51</td>
<td>19</td>
</tr>
<tr>
<td>00-02-2D-2B-86-3A</td>
<td>E260</td>
<td>46</td>
<td>51</td>
<td>19</td>
</tr>
<tr>
<td>00-02-2D-2F-42-3C</td>
<td>22E0</td>
<td>62</td>
<td>51</td>
<td>19</td>
</tr>
<tr>
<td>00-02-2D-2B-85-6E</td>
<td>7D20</td>
<td>51</td>
<td>51</td>
<td>19</td>
</tr>
<tr>
<td>00-02-2D-2F-42-3F</td>
<td>DB50</td>
<td>94</td>
<td>51</td>
<td>19</td>
</tr>
<tr>
<td>00-02-2D-2B-82-4F</td>
<td>4580</td>
<td>80</td>
<td>51</td>
<td>19</td>
</tr>
<tr>
<td>00-02-2D-2F-42-ED</td>
<td>ED80</td>
<td>88</td>
<td>39</td>
<td>17</td>
</tr>
<tr>
<td>00-02-2D-21-9B-32</td>
<td>BDD0</td>
<td>134</td>
<td>45</td>
<td>26</td>
</tr>
<tr>
<td>00-02-2D-2B-86-3A</td>
<td>E280</td>
<td>46</td>
<td>45</td>
<td>26</td>
</tr>
<tr>
<td>00-02-2D-2F-42-3C</td>
<td>22F0</td>
<td>64</td>
<td>45</td>
<td>26</td>
</tr>
<tr>
<td>00-02-2D-2B-85-6E</td>
<td>7D30</td>
<td>48</td>
<td>45</td>
<td>26</td>
</tr>
<tr>
<td>00-02-2D-2F-42-3F</td>
<td>DB60</td>
<td>105</td>
<td>45</td>
<td>26</td>
</tr>
<tr>
<td>00-02-2D-2B-82-4F</td>
<td>4590</td>
<td>79</td>
<td>45</td>
<td>26</td>
</tr>
<tr>
<td>00-02-2D-2F-42-ED</td>
<td>ED90</td>
<td>93</td>
<td>45</td>
<td>26</td>
</tr>
<tr>
<td>00-02-2D-21-9B-32</td>
<td>BDE0</td>
<td>133</td>
<td>45</td>
<td>26</td>
</tr>
<tr>
<td>00-02-2D-2B-86-3A</td>
<td>E290</td>
<td>51</td>
<td>45</td>
<td>26</td>
</tr>
<tr>
<td>00-02-2D-2B-85-6E</td>
<td>3460</td>
<td>46</td>
<td>45</td>
<td>26</td>
</tr>
<tr>
<td>00-02-2D-2F-42-3C</td>
<td>2300</td>
<td>64</td>
<td>45</td>
<td>26</td>
</tr>
<tr>
<td>00-02-2D-2B-85-6E</td>
<td>7D40</td>
<td>50</td>
<td>45</td>
<td>26</td>
</tr>
<tr>
<td>00-02-2D-2F-42-3F</td>
<td>DB70</td>
<td>91</td>
<td>45</td>
<td>26</td>
</tr>
<tr>
<td>00-02-2D-2B-82-4F</td>
<td>45A0</td>
<td>78</td>
<td>45</td>
<td>26</td>
</tr>
<tr>
<td>00-02-2D-2F-42-ED</td>
<td>EDA0</td>
<td>91</td>
<td>40</td>
<td>30</td>
</tr>
<tr>
<td>00-02-2D-21-9B-32</td>
<td>BDF0</td>
<td>133</td>
<td>40</td>
<td>30</td>
</tr>
<tr>
<td>00-02-2D-2B-86-3A</td>
<td>E2B0</td>
<td>52</td>
<td>40</td>
<td>30</td>
</tr>
<tr>
<td>00-02-2D-2F-42-3C</td>
<td>2310</td>
<td>65</td>
<td>40</td>
<td>30</td>
</tr>
<tr>
<td>00-02-2D-2B-85-6E</td>
<td>7D50</td>
<td>44</td>
<td>40</td>
<td>30</td>
</tr>
</tbody>
</table>

Figure 2.2: A segment of snooper trace data. Measurements are collected by monitoring a specific channel for 0.5 second. The first column is the MAC address of the source station; the second column is the sequence number field of the packet's MAC header; the last three columns are the signal strength, noise power and multipath power values, in that order. The values in the first two columns are hexadecimal. Others are decimal.
2.3 Target Machine

For training and testing, I used a Dell Latitude X200 sub-notebook. The X200 has a built-in antenna to support an internal miniPCI WLAN card. A normal PCMCIA slot is also available. Most of the tests and training used a miniPCI version of the aforementioned D-Link card with the internal antenna. I used the custom device driver with this card, which allowed me to vary the card’s transmission power. To ensure that my results were independent of this particular card, I also used a Linksys WPC11 PCMCIA card, which has an antenna built into the card. The Linksys card uses the Intersil Prism2 chipset, a completely different WLAN implementation. Accordingly, it also uses a different set of software drivers.

2.4 Server

The server is a Java program that communicates with the snoopers to collect signal strength measurements on packets observed from the target machine. The server must have sufficient memory and CPU power to contain the Bayesian network. In practice, any modern laptop has more than enough power to do this in real time. In a production environment, however, the server might be tracking a large number of users simultaneously, and thus would likely run on a dedicated machine.
Chapter 3

Observing RF Signals

3.1 Channel Variations

Previous studies [5, 16, 30, 23, 31] in this area directly used the signal strength histogram obtained at each training point to infer a user's location, based on the observation that the signal strength distributions were non-Gaussian, and thus did not necessarily yield meaningful "average" values. By doing this, they essentially assumed that the histograms were not changing over time. However, subsequent experiments, as shown in Figures 3.1, 3.2, and 3.3, indicate that the signal strength histograms vary noticeably over time, with significantly more noise during daytime when more people are in the building. Room temperature, humidity, and interferences from other electronic devices may also be major contributors of such changes.

This experiment was done by placing a sniffer in an office in Duncan Hall and measuring all beacon packets from surrounding APs during a week-long period. Figures 3.1, 3.2, and 3.3 are the measurements for three of the APs during a 24-hour period. The signal strength readings can change dramatically over time. For example, a signal strength histogram generated at 7am may not fit the signal strength distribution at 7pm. Also, in Figure 3.1, the histogram from the first night had two modes, whereas the histogram from the second night had only one mode. These, and other similar measurements, suggest that the average signal strength is a more robust characteristic to use across different days and times of the day.

Also, as shown in Figures 3.1, 3.2, and 3.3, the noise power and multipath power readings are almost identical for all three APs in that experiment. This is because the ACX100 WLAN chip adapts these values across several packets, as oppose to adjusting for each
Figure 3.1: Observed signal measurements over time from a laptop to fixed AP1.
Figure 3.2: Observed signal measurements over time from a laptop to fixed AP2.
Figure 3.3: Observed signal measurements over time from a laptop to fixed AP3.
Figure 3.4: Signal strength readings from four different receivers of signal from a single transmitter, with the transmitter varying its transmission power. Higher values on the x-axis reflect higher transmission power. Higher values on the y-axis reflect a stronger signal at a receiver.

packet. So, for this type of WLAN cards, these values are unusable for location-sensing.

3.2 Transmission Power

A key problem for localizing rogue machines is being robust when the rogue machine uses a transmit power level different from the power used during training. Likewise, the system must also be robust if the rogue machine deliberately changes its transmit power for every packet.

Even though signal strength, in general, is not well correlated to distance in an indoor environment, in most cases the attenuations, reflections, and distortions do scale with transmission power level. So, I expect that at fixed locations relative signal strength should scale well with initial power level. Figure 3.4 shows how the observed signal strength changes as transmission power is varied. The relative ordering of observed signal strengths remains constant with different power levels. Lowering the transmission power seems to reduce the observed power levels in a linear fashion. Most important, the differences in received signal strengths does not vary dramatically as the transmission power changes.
Chapter 4

Location-Sensing Algorithms

The Golden-Eye system uses a Bayesian inference scheme to locate a WLAN user. It models the world as a finite position space \( \{p_1, \ldots, p_n\} \) with a finite observation space \( \{o_1, \ldots, o_m\} \). The sensor model \( Pr(o_j|p_i) \) is a learned model of the conditional probability of seeing observation \( o_j \) at position \( p_i \). A state vector \( \pi \) is a probability vector over the various positions (i.e., \( \pi_i \) represents the probability that position \( p_i \) is the current position). Given a prior estimate \( \pi \), after observing \( o_j \) the new state vector \( \pi' \) can be estimated by calculating the individual conditional probabilities \( \pi'_i \) for each \( i \in \{1, \ldots, n\} \) using Bayes’ rule,

\[
\pi'_i = \frac{\pi_i \cdot Pr(o_j|p_i)}{\sum_{\alpha=1}^{n} \pi_{\alpha} \cdot Pr(o_j|p_{\alpha})}.
\]

Golden-Eye combines these \( \pi'_i \)s into the new estimate of the state, \( \pi' \). It then chooses the most likely position as the representative position from this state vector. In case there’s no one position having significantly higher probability than the others, the localization accuracy can be improved by combining multiple probable positions, unless, as in some rare cases, they are far apart. In subsequence experiments, if the two most likely positions \( p_1 \) and \( p_2 \) are close to each other and have similar probabilities (i.e., \( \pi'_1 \) is at most 3 times \( \pi'_2 \)), Golden-Eye considers the current position to be between positions \( p_1 \) and \( p_2 \), offset proportional to the magnitude of the difference between \( \pi'_1 \) and \( \pi'_2 \).

Each position \( p_i \) is a tuple \( (x_i, y_i, z_i, \theta_i) \), describing a user’s location and orientation. During the training phase, for each \( p_i \), the system takes signal strength measurements from the snoopers and learns the probability distribution of \( Pr(o_j|p_i) \). It then uses a number of localization methods described in Sections 4.1, 4.2, and 4.3, each varying in the filters.
applied to the raw sensor data before being input into the localization system. This has the
effect of varying how the probability $Pr(o_j|p_i)$ is defined.

4.1 Histogram Method

This section summarizes the localization method used by Ladd et al [16]. It serves as a
baseline algorithm to compare against the new algorithms introduced in subsequent sec-
tions.

An observation is defined as a vector of signal strength readings over $k$ snoopers,

$$o_j = (\lambda_1, \ldots, \lambda_k),$$

where $\lambda_p$ is the signal strength measured by snooper $p$. Then $Pr(o_j|p_i)$ is defined as

$$Pr(o_j|p_i) = \prod_{\rho=1}^{k} Pr(\lambda_{p}|p_i).$$

This method directly uses the signal strength histogram obtained from training to get each
$Pr(\lambda_p|p_i)$, so I call it the histogram method, or briefly, Histo.

As I have shown in Section 3.1, the signal strength histogram changes over time, so it’s
not reliable for localization. Indeed, as my results in Chapter 5 show, the Histo method
combines poorly to my other methods.

4.2 Gaussian Method

Since I’ve observed that average signal strength is quite stable as the histogram changes, I
define an observation as a vector of average signal strengths over $k$ snoopers,

$$o_j = (\bar{\lambda}_1, \ldots, \bar{\lambda}_k).$$

Each snooper captures readings from several packets over a short time window, and each
$\bar{\lambda}_p$ is the average signal strength measured by snooper $p$ over this time window.

In radio channel modeling, Rayleigh distribution and Ricean distribution are commonly
used statistical models [22]. However, in order to use the Bayesian inference scheme, the
real signal strength distribution is not as relevant as the observed signal strength distribution. The histograms obtained during the training phase should be used for the prior knowledge. As shown in Figure 4.1, the histograms appear similar to Gaussian distributions $N(\mu, \sigma^2)$, with the average $\mu$ being the average signal strength obtained during training. To confirm this, I plotted the observed quantiles versus the theoretical quantiles in the normal quantile-quantile plots. The distributions fit nicely with the Gaussian distributions. On average, the standard deviation $\sigma$ is 4. This value accommodates sampling errors and the variations of average signal strength during daytime.

The probability $Pr(o_j|p_i)$ is computed as follows

$$Pr(o_j|p_i) = \prod_{j=1}^{k} Pr(\lambda_j|p_i).$$

Once normalized, these values were used in the Bayesian framework to derive a location estimate. This algorithm is called the Gaussian method, or briefly, Gauss.

My experiments showed that a fairly accurate average could be obtained with just a few signal strength readings. The results in Chapter 5 were obtained with an inference window of 2 seconds, which contained about 10 samples from each snooper. I will discuss the effects of different inference window sizes in Section 6.1.2.

### 4.3 Difference Method

The Gauss method assumes the observed average signal strength approximates the trained average signal strength in all cases. This is not necessarily true when the target client is using a WLAN card different from the one used to train the system, or when the target client is intentionally altering its transmission power. In order to accommodate such variations, I developed another localization algorithm, one where only the relative signal strengths are used. Based on the observations described in Section 3.2, I expect that the differences of the signal strength measurements by different snoopers should remain constant as the transmission power changes. So, these difference values should be good metrics to characterize a position.
Figure 4.1: AP1, AP2, AP3 signal strength distribution fitting.
I stored the training data as the differences in signal strength between each pair of snoopers, fitted to the Gaussian distribution $N(\mu, \sigma^2)$, with the average $\mu$ being the average difference in signal strengths. The standard deviation $\sigma$ was chosen as 4 to accommodate sampling errors and the variations of channel conditions over the course of a day.

As each snooper receives a given packet and reports the signal strengths to the localization server, the server computes the difference in signal strengths of consecutive snooper’s measurements. An observation is defined as a vector of differences in signal strength over $k$ snoopers

$$ o_j = (\overline{\lambda_{i_1} - \lambda_{i_2}}, \overline{\lambda_{i_2} - \lambda_{i_3}}, \ldots, \overline{\lambda_{i_{(k-1)}} - \lambda_{i_k}}). $$

During each inference window, the server receives several such observations, and can then compute the average difference in signal strength $\overline{(\lambda_{\rho} - \lambda_{\rho'})}$ for each pair of snoopers $\rho$ and $\rho'$. Through extensive experiments, as described in Chapter 5, I found that using a weighting scheme where the conditional probability of each difference in signal strength was added to the probability for that location gave a more accurate estimate than multiplying them together. The weights $Pr(o_j|p_i)$ were computed as follows:

$$ Pr(o_j|p_i) = \sum_{\rho=1}^{k-1} \sum_{\rho' = \rho+1}^{k} Pr((\overline{\lambda_{\rho} - \lambda_{\rho'}})|p_i). $$

Once normalized, these weights were used in the Bayesian framework to derive a location estimate. This algorithm is called the Difference method, or briefly, Diff.
Chapter 5

Results

In this chapter, I examine the accuracy of Golden-Eye location-sensing system in a number of different scenarios. I first examine straightforward localization using the miniPCI D-Link card with which I trained the system. I then examine localization of a hypothetical rogue machine that varied its transmission power to avoid detection. I also examine localization when using a Linksys PCMCIA card, a totally different form factor, with different transmitter and antenna from the D-Link card used in training.

5.1 Experimental Setup

I conducted the experiments on the third floor of Duncan Hall at Rice University, using four hallways as shown in Figure 5.1. Hallways 1 and 2 are narrow, long enclosed hallways with fiberglass ceiling tiles, carpeted concrete floors, and painted drywall with occasional concrete structural pillars. Hallways 3 and 4 are open to the ceiling of the building some 30 feet overhead. Furthermore, hallway 4 is adjacent to an open-air atrium overlooking the building’s lobby. While the accuracy of my techniques would certainly differ in other buildings, I believe this building offers a diversity of materials and architectural styles that provides a significant challenge to accurate localization.

I placed 5 snoopers (S1 through S5) under the five APs shown in the map. While I could have placed them anywhere, this allowed me to simulate what the current APs could do, if they were augmented to support snooping. It also represented a reasonable distribution to observe all packets on the third floor. The snoopers are between 22 and 50 meters away from each other, with no line of sight between any two snoopers.

During the training phase, I measured signal strengths at positions every 1.42 meters
Figure 5.1: Map of the region of the Duncan Hall where I conducted my tests. There are 5 wall-mounted access points (AP1-5) providing overlapping coverage for this area. I placed one snooper (S1-5) under each AP to measure the signal strength from target machine. The x marks on the map are the points where the training took place.

(most offices in the building are 2.84 meters wide) from one end of each hallway to the other, facing in both directions. Figures 5.2 and 5.3 show the average signal strength recorded for each training point at two opposite directions in the four hallways. Although I actually captured the full histogram of signal strengths observed from each location, I present only the averages of those values to simplify the graph. These graphs show a trend in signal strength variations as I moved from one end of a hallway to the other, and they also show that changing orientation yields different signal measurements.

Each experiment consisted of walking around the loop formed by the four hallways of the test area in a counterclockwise direction, down hallways 1, 3, 2, and 4 in that order. At each training point and between every two training points, I stopped for 15 seconds to allow the localizer to attempt to determine my position. Using the same set of trace data, I
Figure 5.2: Training data in the long hallways. Measurements are taken at 22 positions for each of the two orientations. Missing segments for S1 indicate that signals from those positions were too weak to be detected by snooper S1.
Figure 5.3: Training data in the short hallways. Measurements are taken at 11 positions for each of the two orientations. Missing segments for S1 indicate that signals from those positions were too weak to be detected by snoopers S1.
Figure 5.4: Cumulative error over the four hallways, where the target machine and WLAN card are the same as used in training.

compare the localization accuracy for three different algorithms: Histo, Gauss, and Diff, as described in Chapter 4.

5.2 Basic Localization

When Golden-Eye localized using the same WLAN card as it had used in training, the accuracy was quite high. Over the four hallways, the localization error was at most 2 meters 60% of the time for both the Diff method as well as the Gauss method. The Histo method achieved 2 meter accuracy only 46% of the time. The Diff and Gauss methods are 30% more accurate than the Histo method in this case. The cumulative probabilities of the three methods for error distances within 15 meters are illustrated in Figure 5.4. For the Histo method, the cumulative probability was only 0.88 at 15 meters error distance. That means about 12% of the time, it localized a location that was more than 15 meters away from the actual location of the target machine.
Figure 5.5: Cumulative error over the four hallways, where the target machine and WLAN card are the same as used in training, but the transmission power level is reduced when localization occurs.

Figure 5.6: Cumulative error over the four hallways, where the transmission power level changes for every packet.
5.3 Varying Transmission Power Level

The performance differences among the three algorithms become more evident when localizing a transmitter that transmits at a reduced power level. My results show the Diff method achieving the same accuracy (2 meters, 60% of the time) with lower-power transmission as it achieved with default-power transmission. The Gauss method achieved 2 meter accuracy only 41% of the time, and Histo method achieved 2 meter accuracy only 32% of the time. In this case, the Diff method is 88% more accurate than the Histo method. The cumulative probabilities of the three methods for error distances within 15 meters are illustrated in Figure 5.5.

I ran another test in which I varied the transmission power randomly for each transmitted packet. Golden-Eye achieved accuracy of 2 meters 36% of the time using the Diff method. The Gauss and Histo methods achieved 2 meters accuracy only 12 and 14% of the time, respectively. In this test, the Diff method has more than doubled the accuracy of the other two methods. This result shows that, although an attacker may have some success varying her transmission power, the Diff method vastly improves the system’s ability to detect her. The cumulative probabilities of the three methods for error distances within 15
meters are illustrated in Figure 5.6. Also note that all three cumulative error curves fail to reach 1.0 at 15 meters point. This shows that even the best performer, the *Diff* method, has about 20% of the time localized a position with a error distance more than 15 meters.

### 5.4 Varying Transmitter

When Golden-Eye localized using a different WLAN card from the one it used in training, it maintained acceptable accuracy when using the *Diff* and *Gauss* methods. An error of at most 2 meters was achieved at 45% of the sampled positions. The *Histo* method achieved this accuracy for 39% of the sampled positions. The cumulative probabilities of the three methods for error distances within 15 meters are illustrated in Figure 5.7. Although this result is not sufficient to prove that my mechanism can handle every WLAN card, it shows that, comparing to the *Histo* method, the *Diff* and *Gauss* methods are more robust against transmitter differences.

### 5.5 Untrained Locations

Any localization method which relies on training can only localize targets to points within the trained area. Moreover, small deviations from the trained positions can confuse a training-based localization system. Although the *Diff* method performs the best in the face of modulation of power levels and unmodeled hardware, it still can not correctly localize a mobile device anywhere outside the four hallways. I tested how it would perform when localizing a laptop in untrained positions along the training hallways.

The first test involved localizing a laptop which was very close to the wall in a trained hallway; the original training data was taken in the center. *Diff* achieved an accuracy of 2 meters at 49% of sampled points. Comparing to 60% accuracy in the center (as described in Section 5.2), this represents a performance degradation of 18%.

I also tried to localize a laptop which was facing perpendicular to a trained direction. *Diff* achieves an accuracy of 2 meters at 50% of sampled points (versus 60% facing trained
directions), a performance degradation of 17%.

The results from both cases are illustrated in Figure 5.8. They are comparable to other cases which break the model, including varying power level and varying transmitter tests.

Finally, I tested how the Diff method performed when the target was not even in one of the trained hallways. I sampled five test positions on the third floor of Duncan Hall, two positions in rooms along the trained hallways and three positions in nearby hallways, as shown in Figure 5.9. In all of those five tests, the Diff method estimated locations within the training area that were less than 2 meters away from the best possible matches.

I also tested several positions on the second floor. The error distances were much larger than on the third floor. For position below hallway 1, 3, and 4, the Diff method localized to the correct respective hallways, although not always the position directly above. There is no hallway on the second floor directly below hallway 2, however there is a hallway parallel to hallway 2 that is open on one side to the open area that cuts across the middle of the building. Attempting to localize positions on that hallway, the Diff method concluded that I was on hallway 3. Although I was physically closer to hallway 2, I had line-of-sight with hallway 3, so this conclusion is not surprising.

Of course, no training-based method will return locations outside of its training set. However, as these results show, the Diff method can often derive positions in the original training set that are meaningfully “close” to the target’s actual location. Therefore, in practice, it should be sufficient to train in major hallways and open areas to yield useful localization results, even if the target machine is actually in a side office.
Figure 5.8: Cumulative error of tests on untrained poses comparing to that of the trained ones. All of the four hallways were tested, and the positions were estimated by using the Diff method.

Figure 5.9: Localization results when the Diff method was used to locate a laptop outside the training area. A, B, C, D, and E are the tested locations. A’, B’ C’, D’, and E’ are the locations that Golden-Eye estimated using the Diff method.
Chapter 6

Discussion

This chapter discusses a number of issues regarding the work described in this thesis. I first address some methodology aspects of my work. Then, I discuss the performance of this work in relation to the performance of previous work in wireless localization. I also discuss some related privacy concerns.

6.1 Methodology Revisited

6.1.1 Weighting Scheme vs. Probability

With the Gauss method, Golden-Eye got the best location-sensing results by computing the conditional conjunctive probability. By contrast, with the Diff method, when Golden-Eye was using relative sensor readings, it got the best results by computing relative weights, summing up the conditional probabilities, rather than multiplying them. Part of the reason is that with the Diff method, computing the conditional conjunctive probability is not simply a matter of multiplying the individual conditional probabilities. Since it is computing the probability of differences, a single signal strength reading will appear in two different probability values, as shown in the definition of $o_j$ in Section 4.3. This means the probability values are not independent. Multiplying them together would be meaningless and the result does not converge. So, I chose to use a weighting scheme in that method. Although still does not represent a meaningful probability value, the sum is much more stable.
Figure 6.1: The effect of varying inference window size on the localization accuracy. Shown here are the likelihoods of error within 2 meters for all three algorithms when locating a target transmitting at a reduced power level.

6.1.2 Inference Window Size

Due to sampling errors and environmental factors in an in-building WLAN, meaningful localization results can only be obtained over a number of signal strength samples. An inference window is a short period of time during which a series of samples is taken and combined together to infer a single location estimation. A larger window size can potentially increase localization accuracy for stationary targets, but it will decrease the responsiveness of the system and degrade the ability to track mobile targets. Concerns arise especially for the Gauss and Diff methods, in which average signal strength readings are used. My experiments show that these methods did not require a large window size to perform well. In fact, localization accuracies only improved slightly beyond two seconds, which contained about 10 samples in my experiments. As an example, Figure 6.1 shows the effect of varying the inference window size when locating a target transmitting at a reduced power level. The results in Chapter 5 were all generated with inference window size of two seconds. In reality, however, if very few packets are transmitted by the target, the localization system may need to adjust for a larger window size, or the localization results may become less accurate.
6.1.3 Limitations

One major limitation of the Bayesian approach is that these methods can only localize targets within the trained area. For instance, Golden-Eye will not generate any location estimation beyond the four hallways. While this may not be a problem for most location-aware applications, it could be a serious weakness when tracking down a malicious WLAN machine, as an intruder could be hiding anywhere in the WLAN service area. To fundamentally solve this problem, it may be necessary to combine a signal propagation model into the current system.

This thesis suggests that it may be very difficult to physically hide a wireless network node that is actively transmitting packets. However, it does not consider coalitions of hostile nodes, working in concert to attack a network. Such coalitions might be able to confuse the localization system by presenting the illusion of an attacker being simultaneously in multiple locations, particularly if some of the hostile nodes are actively moving. A more robust localizer might try to determine the number and locations of hostile nodes by using clustered sensors. Youssef et al [31] provides one example of this approach.

A further concern is attackers operating at a significant distance, using parabolic or otherwise non-standard antennas. It is still an open question as to whether the in-building snooper network could be trained to identify, at a minimum, the compass direction from the building to the attacker. Such information could greatly reduce the effort necessary to find the attacker.

Finally, this thesis focuses only on static location-sensing. No target movements are considered. In order to track a moving target, it may require postprocess filtering or sensor fusion of localization inferences to obtain acceptable accuracy.

6.2 Comparison to Prior Work

In prior work [16], which used the Histo method in a client-side localization setting, 2 meter accuracy of 65% to 80% were reported in the same building. There are several factors that
cause my results to be not as accurate.

First, the results described in prior work [16] are per hallway numbers, whereas the results in Chapter 5 are cumulative error over four hallways. Knowing the target is on a particular hallway and using only the corresponding subset of the training points will certainly improve the localization accuracy.

Another factor to consider is that the previous work was engineered for a particular hardware and for limited time window of the day. In this work, my primary goal is robustness. The Golden-Eye system has the ability to work around the clock to locate a device using a different transmitter, using a different antenna configuration, and even transmitting at an unanticipated power level.

Furthermore, in the experiments described here, only five snoopers were available, and generally no more than four were used for localization at a given position. In prior work [16], however, up to nine APs were used for localization at any position. Additional sensors clearly help increase measurement accuracy.

6.3 Privacy Concerns

The results presented in this thesis emphasized the point that any machine broadcasting on a wireless network is advertising its location. This presents some privacy concerns, and perhaps some promising directions for future research. A rogue network operator or any other user on the wireless network could use a system like Golden-Eye to locate legitimate agents on a wireless network and associate individual transmissions with the transmitting agents. However, these risks are already present on wired networks. This work serves to demonstrate that anyone, malicious or otherwise, should not depend on anonymity being provided by wireless networks.

My results would tend to imply that a system can reduce its likelihood of being detected by changing how its antenna works and varying its transmission power. By replacing an omnidirectional antenna with a directional one, or simply by covering and uncovering the antenna with the operator's hand, an attacker can cause the localization system to exhibit
lower accuracy. However, improvements in location-sensing techniques will eventually negate such attempts. In order to protect legitimate users' right to privacy, new techniques beyond signal obfuscation must be developed. This is a field that calls for future research to provide demonstrable robust detection avoidance solutions.
Chapter 7

Conclusion

Traditional localization methods tend to have simple models of how nodes will behave. Malicious nodes can easily violate these assumptions by using different radio transceivers, changing transmission power level, or modulating their transmission power for each packet. This thesis presents a server-side location-sensing system, Golden-Eye, for locating mobile devices in an indoor environment, even when the nodes might attempt to hide their locations. Golden-Eye uses techniques that are robust against a variety of model errors, including malicious nodes, nodes with different hardware than it was trained against, and, to some extent, nodes located outside of the training area. This thesis demonstrates that a robust location-sensing system can be built in a WLAN with reasonable accuracy.
Bibliography


