Accurate Distance Tracking using WiFi

Martin Schüssel
Institute of Communications Engineering
University Ulm
Email: martin.schuessel@uni-ulm.de

Abstract—This Paper presents an approach to indoor distance tracking using commercial off-the-shelf (COTS) WiFi hardware. We show that it is possible to exploit the channel state information (CSI), exposed by WiFi cards, to track the distance to other clients with cm accuracy. The proposed method does not require any hardware modifications, and highlights and solves some of the problems often encountered when using the CSI for indoor positioning applications.

I. INTRODUCTION

Indoor Positioning using existing infrastructure is often favoured, since installing new hardware for positioning is expensive. Localization with WiFi has received much research attention in the past years. The first systems used the signal strength toward several Access Points (APs) to achieve positioning[1]. While this approach has been refined and improved in recent years, the methods using RSS still only achieve accuracies of at best 1m, while also requiring an extensive measuring phase to learn the signal strength before they can be used.

For more accurate positioning the channel state information can be used. Every WiFi packet is transmitted with a preamble, that is used to estimate the channel between the receiver and transmitter. This information is exposed as the CSI, by some modification to the driver [2],[3]. The CSI contains the phase and amplitude of the channel on a subset of the frequencies WiFi uses. Usually some preprocessing schemes are used before applying the CSI for indoor positioning. This is needed since the measured CSI will be influenced by the packet detection delay (PDD), and the frequency offset between the receiver and transmitter. These preprocessing methods usually loose the real phase information of the channel. Figure 1 shows the preprocessing scheme used in [4]. As long as this preprocessing is consistent one can still use this distorted CSI.

This is reasonable when using the CSI in pattern matching algorithms, or when using the phase relationship between different antennas on the receiver.

We use the preprocessing method proposed in [6] to extract real phase information from CSI measurements. We then use it to track the distance between two wireless devices. In contrast to [6] our approach does not require channel hopping, and thus is potentially less disturbing to WiFi networks.

II. RELATED WORK

The channel state estimation has been used in [5] to achieve meter level accuracy with a pattern matching algorithm. To our knowledge they were the first to use a preprocessing scheme to use the CSI for pattern matching.

SpotFi [4] achieves decimeter level accuracy. While they also use a similar preprocessing scheme they afterwards apply a modified version of the MUSIC algorithm to calculate the distance and the angle of the transmitter. They use several APs in their system, and require multiple packets to select the correct peak in the spectrum that is calculated by MUSIC. This is problematic if the target moves during a measurement. Closest to our work is [6]. They introduced the preprocessing methods we use to extract the real channel from the faulty CSI measurement. We expand on their work for the channel estimation. For positioning they use the nonuniform Fourier transform to calculate the distances between the receive and transmit antennas. This requires using all available WiFi channels. Neither the transmitter or the receiver will be able to perform their normal duties (e.g. serving other clients as an AP, or receiving data as a client), while measuring over all the available WiFi channels. The key difference in our approach is, that we do not require measurements from multiple channels, which means that our method should lead to less interference with normal network operation. On the other hand we are only able to track the distance, not actually estimate the absolute distance between two WiFi devices.

III. CHANNEL STATE INFORMATION

The Channel State Information (CSI) is determined in the preamble of every WLAN packet. It is used to determine the phase and amplitude variations the channel introduces.
to the signal. This enables quadrature amplitude modulation for the actual data transmission. Although normal WLAN drivers do not make the CSI available, there are modified drivers for Intel [2], and Atheros [3] WLAN cards that provide these measurements. The channel is assumed to be static for the duration of a packet. More formally the CSI consists of distorted measurements of the wireless channel.

\[ h_n = ae^{-j2\pi f_n \tau} \]  

(1)

Where \( h_n \) is the wireless channel for the \( n \)-th subcarrier, \( a \) is the attenuation, \( f_n \) is the frequency at subcarrier \( n \) and \( \tau \) is the time of flight. We are especially interested in the phase of the channel:

\[ \angle h_n = -2\pi f_n \tau \mod 2\pi \]  

(2)

This equation means that we can easily calculate \( \tau \) which corresponds to the distance between the receiver and the transmitter. Adding or subtracting multiples of \( 2\pi \) will not change the phase, so we have to deal with ambiguities. Because we need to eliminate the packet detection delay (see [II-A], we only have the center subcarrier and its phase available. This means that we can not calculate the real distance, but we can follow changes of it.

The packet detection delay (PDD) and carrier frequency offset (CFO) introduce errors into the channel estimation. In the next sections we will explain how the PDD and CFO affect the CSI and how one can overcome their effects.

A. Packet Detection Delay

The Packet Detection Delay is introduced since the WLAN card detects the presence of a packet at slightly different times. A packet is detected if the first few samples of a signal cross a certain energy threshold. The number of samples required, the distance to the receiver and noise can influence this delay. In practice the PDD influences the CSI by introducing an additional delay that looks similar to the Time of Flight. In [6] it is shown that the PDD can be about 8 times higher than the actual time of flight and varies significantly between packets. They also show that the center subcarrier does not experience this delay. Since this subcarrier can not be measured accurately and is not reported by the WiFi card, we estimate it by using the subcarriers closest to it.

B. Frequency Offset

The clocks of the receiver and transmitter differ by a small amount. The phase measurement done by the receiver will include the phase change introduced by the clock differences between his and the transmitters clock. The difference between the frequencies will usually be several kHz (the Wifi standard allows up to 200 kHz). This phase change adds up quickly and leads to large errors in the phase of the estimated wireless channel. [6] noted that this change in phase is reversed if we send a packet from the receiver to the transmitter. We follow their approach at removing the CFO.

Assuming two Wifi clients, a packet sent from client 1 to 2 (and measured at 2), will experience a certain phase shift introduced by the CFO. A packet sent in the reverse direction will experience the same phase shift, but with reversed direction. If we focus on one subcarrier we can formulate the two channels measured in the following way:

\[ \hat{h}_{12} = h e^{j(f_1-f_2)\tau} \]  

(3)

\[ \hat{h}_{21} = h e^{j(f_1-f_2)\tau} \]  

(4)

where \( \hat{h}_{12} \) and \( \hat{h}_{21} \) are the measured wireless channels from 1 to 2 and reversed, \( h \) is the true wireless channel, and \( f_1 \) and \( f_2 \) are the center frequencies at client 1 and 2. Because of the reciprocity of the wireless channel the true channel is the same no matter which client receives or transmits. The measured channels will also be affected by additional phase errors because of delays in the hardware (e.g. cables from the antennas to the Wifi card). We accumulate these changes into a new variable \( k \). Assuming we have access to both of the measurements at the exact same time, we can multiply the channels leading to:

\[ \hat{h}_{12}\hat{h}_{21}k = h^2 \]  

(5)

So we can recover the squared wireless channel from the two measurements. The phase of this channel will change twice as fast as we would expect from Eq 4 but we can still use it to track the distance.

In practice it is impossible to send two packets at the same time. We implemented a program that replies immediately when a packet is received. Since there will be a delay between the two measured CSIs, we expect a small drift of the squared channel over time, since the frequency offset will not be completely eliminated.

IV. Distance Tracking

Now we can write the following equation to correlate the distance to the measured phase:

\[ d = -\frac{\angle h^2}{4\pi f_c} \]  

(6)

where \( d \) is the distance, \( c \) is the speed of light, \( \angle h^2 \) is the phase of the squared channel, given by the phase of Eq 5 and \( f_c \) is the center frequency. Using the squared channel means that the ambiguities happen twice as often, since its phase changes twice as fast. For the frequencies used in WiFi this means that we have a valid solution every 6 cm in the 2.4 GHz band and about every 3 cm in the 5 GHz band.

The channel model so far assumed only one path from the receiver to the transmitter. This assumption is not true in indoor environments. Usually the signal arrives at the receiver from multiple paths at once. The received signal is a mixture of all signals received from all paths. The multipath will change even for small movements and affect the phase and the amplitude of the received signal. Since we currently do not account for multipath, we try to minimize its influence on the results by experimenting with strong line of sight between the receiver and transmitter. Although this will not remove...
the multiple paths, it should minimize their impact, since the strong line of sight signal will be the dominant part of all the paths.

V. EXPERIMENTAL RESULTS

We used an off the shelf WiFi Card (Intel 5300) for our experiments. A normal desktop PCs was used as an AP. For the mobile client we used a notebook. Ubuntu 14.04 LTS was installed on both systems, with the driver modifications from [2]. The Wifi NICs were installed into PCI boards with 3 antennas, but we only used one transmit and one receive antenna. To make sure we had good line of sight between them we placed them at the same height, without any obstructions between the antennas.

A. Stability of CSI measurements

The first experiment revolves around the stability of the channel measurements. We sent 10000 packets over 100 seconds between the AP and the client. The center frequency was 5.32GHz (WiFi channel 64). Then we calculated the squared channel by estimating the center subcarrier, and multiplying the CSI from both directions. We performed these experiments in a room, with no person inside during the measurements. We repeated the experiment 10 times at different positions in the room. The phase of the squared channel of 2 representative measurements can be seen in Fig. 2.

At first one can notice the results drift slightly over time. This is to be expected, since we are not able to remove the frequency offset completely. Fig. 2b was the worst drifting we measured over 100 seconds. If one uses the mean over 100 CSI measurements to counter the noisy measurements this corresponds to a drift in distance of about 4cm. Although we can not guarantee that the drift will always be smaller than this, and more measurements are needed, it gives a good indication of what kind of drift to expect in most situations.

There are also several clusters of phase measurements. Fig. 3 shows the first 1000 measurements of 2b and displays the clustering behaviour. The clusters are more problematic than noise, because if we mean the measurements, e.g. over ten points, several measurements in the clusters outside the true phase will have a big impact on the result. Clusters like these have been observed in [5]. They note that these clusters are location dependent and that not every location experiences this clustering behaviour in the CSI. To the best of our knowledge they they do not provide a reason why these clusters appear in the first place. From indoor channel measurements with high quality equipment [7] it is known, that multipath components often arise in clusters. So one explanation might be that these multipath clusters are not stable over time. On the other hand the Intel 5300 WiFi cards only have a prototype implementation of the multiple antenna CSI measurements according to the developers of the modified drivers. Under certain conditions it is known that they do not provide good measurements (e.g. in the 2.4 GHz band). So we are not able to rule out the hardware as a root cause of these clusters. Lastly we did not perform the experiments in a shielded room, but in an office environment with active WiFi devices. These could also interfere and lead to additional noise in the measurements. More experiments with different hardware are needed to determine the cause for the clustering behaviour with certainty.

Independent of the reason we choose to average the channel over 100 measurements (achieved by 200 packets), to minimize the impact of these clusters and to counter noise.

B. 1-D Distance tracking

For the second experiment we moved the laptop in a straight line toward the AP, while leaving the latter static. We injected 12000 packets over the course of 12 seconds, so 10000 packets per seconds. We started at a distance of 70cm and after two seconds started to reduce the distance to 60cm. We averaged over 100 packets, resulting in 10 distance measurements per second. We repeated the experiment 4 times. We used the first 10 measurements to calibrate the distance to 0. Here we show the relative distance calculated by Eq. 6.
As one can see from Fig. 4, the relative distance follows the expectation closely. The clients were moved manually, which explains the slightly different start and end points of the runs. We performed a similar experiment under worse conditions. We set the starting distance to 4m. We increased the distance by 50cm, waited for roughly 2 seconds and then returned to the starting position. The results can be seen in Fig. 5. While we are still able to follow the movement, one can see that the distance is not as accurate as in Fig. 4, and that the ending distance is about 7cm too close to the AP. One of the reasons for the worse performance is the increased distance between the two WiFi devices. This will lead to a weaker signal and more noise in the measurements. Also multipath propagation should have a bigger impact when the client and AP are further apart.

VI. CONCLUSION AND FUTURE WORK

We presented a method to track the distance between two devices equipped with off the shelf WiFi hardware. With only software modifications we showed that it is possible to track the distance with cm accuracy. More experiments in different environments are needed to give a better estimation of the achievable accuracy. Especially experiments with several simultaneous distance measurements, leading to 2 or 3-D position tracking are interesting. Currently our approach requires an extensive amount of packets and some calibration. A faster implementation of our software might be able to only require calibration once for a new setup. The amount of packets could be greatly reduced by using multiple antennas or sensorfusion with an inertial measurement unit. Multipath mitigation might also be possible by using multiple antennas and applying the MUSIC algorithm with time of flight and angle of arrival estimation.

REFERENCES
