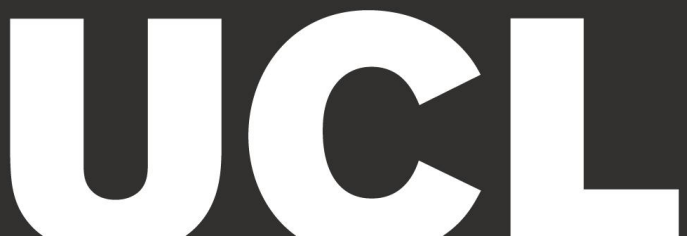
UCL DEPARTMENT OF
COMPUTER SCIENCE



Research Note

RN/14/04

ArrayPhaser: Enabling Signal Processing on WiFi Access Points

March 12, 2014

Jon Gjengset Graeme McPhillips Kyle Jamieson

Abstract

Signal processing on antenna arrays has received much recent attention in the mobile and wireless networking research communities, with array signal processing approaches addressing the problems of human movement detection, indoor mobile device localization, and wireless network security. However, there are three important challenges inherent in the design of these systems that must be overcome if they are to be of practical use on commodity hardware. First, phase differences between the radio oscillators behind each antenna can make readings unusable, and so must be corrected in order for most techniques to yield high-fidelity results. Second, in many deployments, access points are elevated to maximize coverage, introducing height differences between the mobile device and antenna array (usually the access point) that can skew results. Third, while the number of antennas on commodity access points is usually limited, most array processing increases in fidelity with more antennas. Worse still, these issues work in synergistic opposition to array processing: without phase offset correction, no phase-difference array processing is possible, and unknown height differences between mobiles and access points make automatic correction of these phase offsets even more challenging. Furthermore, limited numbers of antennas result in poor fidelity, complicating both problems further. We present ArrayPhaser, a system that solves these intertwined problems to make phased array signal processing truly practical on the many WiFi access points deployed in the real world. Our experimental results on three- and five-antenna 802.11-based hardware show that 802.11 NICs can be calibrated and synchronized to a tolerance of 15° median phase error, enabling inexpensive deployment of numerous phase-difference based spectral analysis techniques previously only available on costly, special-purpose hardware.

This material is based on work supported by the European Research Council under Grant No. 279976.

1. INTRODUCTION

In recent years, there has been renewed interest within the mobile and wireless networking communities in addressing problems related to the sensing of signals using multi-antenna (MIMO) access points. Recent examples include systems that localize RFID tags [28] and WiFi devices [23, 31], and passive radar systems that pinpoint human movements [1, 2, 17].

These systems, and others, share an important common thread: they all rely on phased array signal processing; a set of techniques that makes various comparisons between the radio signals received from each of the antennas of the access point (AP). Phased array processing has been applied in weather and military radar, seismology, and astronomy to great benefit, but its application to indoor and outdoor wireless and mobile communications presents new challenges and dictates novel system designs, as the above work demonstrates.

While the aforementioned work makes phased array signal processing practical in mobile wireless local-area network designs, three important challenges remain that must be overcome if it is to be widely deployed on off-the-shelf hardware:

1. On a single commodity wireless NIC, the RF oscillators are frequency locked, preventing their relative phases from drifting over time. However, these oscillators still have an unknown, absolute phase offset relative to one another. This offset must be compensated for in order to obtain readings from an AP that have meaning in physical space.
2. Today's cutting-edge APs are equipped with multiple unsynchronized network interface cards (NICs), each with multiple antennas [32], but phased-array signal processing benefits most from phase differences between all pairs of antennas, whether they share a radio card or not. The complete lack of synchronization across NICs in both time and radio frequency stymies attempts to run the above systems across multiple NICs at the same AP.
3. All the above systems benefit from a translation between the readings at the antenna array and physical space, but close to the AP, elevation differences between the AP and the mobile client derail the relationship between the two, degrading the accuracy of measurements.

In this paper, we present *ArrayPhaser*; a system that enables phased array signal processing for commodity WiFi APs. We also propose minor, cost-effective hardware modifications that can further improve the phase-data fidelity on such APs. WiFi today is ubiquitous, with APs deployed everywhere people congregate. ArrayPhaser's vision is to convert every one of these APs into a miniature phased array receiver, so that when integrated with ArrayPhaser, the above systems can truly run pervasively—everywhere WiFi APs are deployed.

To realize this vision, ArrayPhaser addresses the above three challenges with three corresponding wireless system components that we implement at a backend server connected to the APs over Ethernet backhaul:

First, novel software-based digital signal processing on the general-purpose CPU of the backend server works alongside an inexpensive custom antenna design at the AP to enable multiple commodity NICs to be synchronized in time and frequency, and form a larger and more useful phased array.

Second, ArrayPhaser introduces a novel elevation compensation algorithm to correct for differences in elevation between mobile clients and APs. We observe a synergy between elevation compensation and phased-array signal processing: even applications that do not require a user's elevation benefit by estimating that user's elevation, because that estimate can increase phase-data accuracy in the horizontal plane.

Finally, ArrayPhaser introduces a novel phase *autocalibration* algorithm that corrects the phase offsets between the different radio oscillators at an AP. Our autocalibration algorithm operates in the presence of noise and indoor multipath reflections, and leverages the elevation compensation algorithm above to further improve calibration results.[1]

We observe that the system's components require only the addition of software-based digital signal processing on a general-purpose CPU, and for multicard operation, the addition of custom-designed, yet simple and inexpensive hardware: RF switches, splitters and antennas with known geometries. Most of the computational load is also performed by a server connected to the Ethernet backhaul, further reducing the changes needed on the APs. We therefore position ArrayPhaser as a likely design pattern for future 802.11 AP hardware designs that is compatible with the MIMO designs in 802.11n and upcoming MU-MIMO designs in 802.11ac.

Contributions. To summarize, this paper makes the following three research contributions:

1. We introduce a multicard synchronization algorithm and design that allows multiple NICs to take phase-difference readings as if they were one large antenna array.
2. We propose an autocalibration algorithm so APs may calibrate their antennas automatically—even across multiple NICs—based on transmissions from other clients or APs.
3. We integrate the above two ideas with an elevation compensation algorithm into a system called Elco, which corrects for phase errors introduced by height difference between a client and an AP.

Paper roadmap. The rest of this paper is organized as follows: we begin with a detailed description of ArrayPhaser's design (§2) followed by details of our implementation on commodity Intel 5300 NIC hardware, combined with a custom, low-cost switching and antenna hardware circuit design (§3). Our experimental evaluation (§4) measures the quality of the ArrayPhaser AP improvements both in anechoic chamber isolation and in end-to-end operation in an office environment. Results show that ArrayPhaser can resolve multipath reflections using multiple NICs to a $\pm 2^\circ$ accuracy, and automatically calibrate APs in a busy office environment to an average accuracy of 15–20° phase error, yielding qualitatively similar angle-of-arrival information to that of expensive, specialized hardware built for phased array processing. We also

[1] N.B.: this phase calibration is conceptually distinct from the "calibration" or "war-driving" that some indoor location systems require, involving a human taking readings within an indoor space.

report improvements in calibration, and a significant (20%) improvement in median localization accuracy due to elevation compensation alone. We cover related work in §5 and conclude in §6.

2. DESIGN

In this section, we discuss the three main components of ArrayPhaser that allow accurate phase-difference readings to be extracted from commodity RF hardware. We first describe the theory of multicard operation in §2.1. In §2.2, we tackle the problem of automatic detection of radio-chain phase offsets, and in §2.3, we describe how to correct for elevation differences between a transmitter and an array.

2.1 Multicard operation

On a single NIC, phase data from each antenna can be compared after compensating for the phase offsets introduced by each of the radio chain oscillators as described in §2.2. However, across radios, many other factors distort the OFDM signal, and each of these varies slightly between the two NICs:

1. The sampling frequency f_s, with which the RF front end samples the incoming signal in the time domain.
2. The carrier frequency f_c to which the RF front end is tuned. f_c and f_s both vary over time because the two NICs' RF oscillators are not frequency-locked.
3. The RF sample clock time offset τ, measured in time samples relative to the beginning of an OFDM symbol. This varies because the two NICs detect and acquire the incoming frame at slightly different points in time.
4. RF oscillator phase offset ϕ. As noted above, this varies between antennas on one NIC because it is locked to a random but constant value at power-up. It varies between antennas on different NICs because it is constantly changing, due to slightly different f_c on each.

To further complicate matters, all the above quantities vary on a frame-by-frame basis, so any correction algorithm must be able to work at line rate as frames arrive from the NIC. In this section, we present a hardware design and a synchronization algorithm that lets us compensate for these differences, and thereby enable phase-difference signal processing across antennas on multiple NICs as if they were phase-locked.

2.1.1 Hardware design

To achieve higher antenna counts, ArrayPhaser augments an AP's NIC with one or more receive-only NICs, operating on the same channel. ArrayPhaser splits the incoming signal from one antenna and routes it to one radio chain on both NICs, as shown in Figure 1. This allows one radio chain on each NIC to receive a signal that we know is equal (as viewed at the antenna) to that which a radio chain on the primary NIC received. While the result is a five-antenna AP, the raw CSI[2] readings from the two unsynchronized NICs cannot be combined for phased array processing without compensating for inter-NIC irregularities.

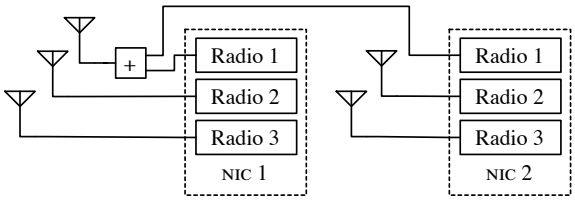


Figure 1: ArrayPhaser's multi-radio hardware design for the case of two NICs. The design generalizes to any number of NICs, each sharing one antenna from their first radio.

2.1.2 Software synchronization algorithm

Upon receiving CSI from each NIC, we first isolate the readings from the shared antenna introduced above. We then arbitrarily, but consistently across packets, select one NIC as the *primary*. For the purposes of discussion, we refer to the frequency-domain CSI of the primary and secondary NIC as $H_1[k]$ and $H_2[k]$, respectively. We now seek a transformation of $H_2[k]$ so that readings for the shared antenna on that NIC matches the readings taken on the primary. We know such a transformation must exist since the same signal arrived at the antennas We know such a transformation must exist since the two NICs observed the same signal at the same antenna.

Synchronizing carrier and sampling frequency. We first reconcile f_c and f_s between the two NICs. A difference in f_c between an OFDM sender and receiver manifests in an added phase in H_2 whose magnitude is constant across subcarriers [26]. Suppose the difference in f_c between the sender and NIC 1 is $\Delta f_{c,1}$, and the difference in f_c between sender and NIC 2 is $\Delta f_{c,2}$. The two NICs will experience added phases whose y-intercepts are as shown in Figure 2.

Furthermore, if the sender transmits with a sampling frequency of f_s', then the fractional difference in sampling frequency between sender and NIC 1 is $\zeta_1 = \frac{f_{s,1}}{f_s'} - 1$, with an analogous definition for NIC 2. This manifests as an added phase whose magnitude varies linearly across subcarriers [26] with slopes $2\pi\zeta_1$ and $2\pi\zeta_2$ for NICs 1 and 2, respectively, as shown in Figure 2.

We correct for both carrier and sampling frequency differences by setting the angle of each complex sample from the second card to the angle of the corresponding sample from the primary card while leaving the magnitudes untouched. In the frequency domain, this rotates $H_2[k]$ by the difference between the two curves shown in Figure 2, creating a new CSI for NIC 2 with a phase shift of $\gamma = 2\pi\left(\Delta f_{c,2} - \Delta f_{c,1}\right) T_s$ and a linear phase shift in frequency of slope $\alpha = 2\pi\left(\zeta_2 - \zeta_1\right)$:

$$H_2'[k] = H_2[k] e^{j(\gamma + \alpha k)}. \quad (1)$$

Synchronizing sample clocks. Due to multipath effects, two NICs also acquire an incoming transmission at slightly different points in time, causing a shift τ in the time-domain

[2]WiFi standards refer to the channel readings that arrive with each received frame as *channel state information* or CSI. CSI readings are primarily used for channel estimation, but also contain the phase information that phased-array signal processing techniques require. We refer to phase information and CSI interchangeably throughout this paper.

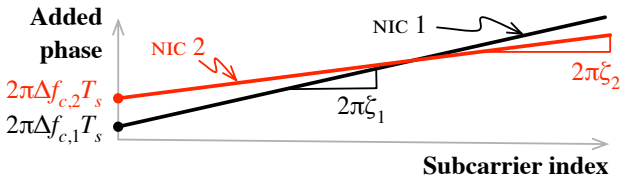


Figure 2: Phase added to a CSI measurement of an OFDM reception due to transmitter-receiver carrier frequency differences $\Delta f_{c,1}$, $\Delta f_{c,2}$ and the respective sampling frequency differences ζ_1, ζ_2 between the transmitter and each of the receiving NICs.

samples. To see the effect this has on the CSI, let us examine the Discrete Fourier Transform of NIC 2's CSI:

$$H_2[k] = \sum_{n=0}^{N-1} h_2[n] e^{-j2\pi kn/N}. \quad (2)$$

If we circularly shift $h_2[n]$ in time by a whole number of samples, it can be shown that the effect in the frequency domain is again a phase shift linear in frequency:

$$H_2[k] e^{2\pi j \tau_0 k/N} = \sum_{n=0}^{N-1} h_2\left[(n-\tau_0)_N\right] e^{-j2\pi kn/N}. \quad (3)$$

Comparing the right-hand-side of Equation 1 with the left-hand-side of Equation 3 we see that the carrier and sampling frequency synchronization step also corrects such time shifts.

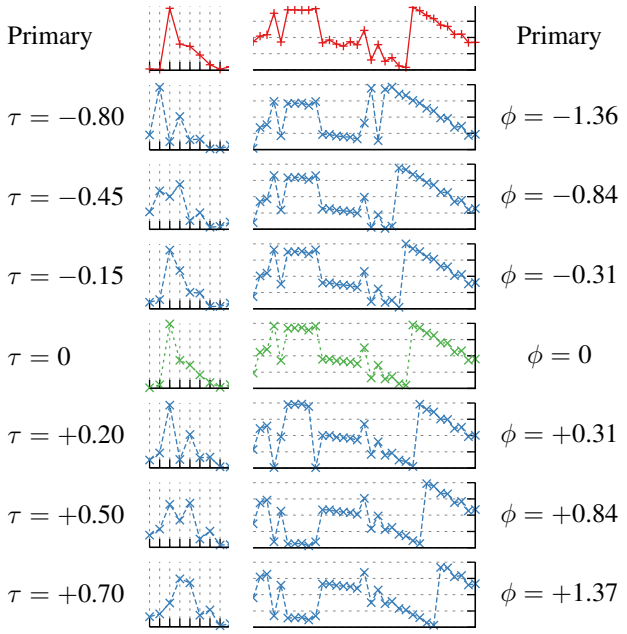
This shift will in practice be *less* than one sample period in magnitude. Nonetheless, it can also be shown that a linear phase shift in frequency whose slope is fractional approximates a filter that shifts the sampled signal $h_2[n]$ by reconstructing the underlying continuous signal and resampling that fractional number of samples later in time [12]. The upshot of this is that our carrier and sampling frequency synchronization step also synchronizes the two NICs in time even down to a fraction of a time sample.

Figure 3(a) shows the shared antenna signal from the primary NIC (uppermost plot) and from another NIC, with varying fractional delays applied (remaining plots) after f_c and f_s have been compensated for. We see that the frequency-domain transformation has temporally aligned the two signals. Note that for the longer fractional delay, the peak falls between two samples, and thus becomes obscured.

Correcting oscillator phase offset. The absolute phase measured by the radio chains connected to the shared antenna on each card varies, as their oscillators have different phases. More importantly, the two vary continuously and independently of each other, and can thus not simply be measured once like the inter-antenna phase offsets on a single NIC with frequency-locked oscillators discussed in §2.2. In order to give the appearance that the oscillators from the two NICs are all phase-locked[3], our synchronization algorithm must therefore also cancel out this unpredictable phase offset.

This phase offset manifests as a phase rotation of every time-domain sample. Again, however, consider the right-hand-side of Equation 1. By adjusting γ to equalize phase in

[3]If two oscillators are phase-locked, their absolute phase difference is constant over time.



(a) Clock delay **(b)** Oscillator phase offset

Figure 3: Magnitude (a) and phase (b) for consecutive time-domain samples from NIC 1 and NIC 2 with different ArrayPhaser search parameters after compensating for f_c and f_s.

the frequency domain, we do the same in the time domain, and so align the phase of the primary NIC's signal with the secondary's, as shown in Figure 3(b).

To complete the synchronization, we apply the correction $e^{j(\gamma+\alpha k)}$ from Equation 1 to the samples from all the other antennas on the non-primary NIC. We also multiply the magnitude of the time-domain samples taken from the shared antenna by the number of cards present in order to counter loss introduced by the splitter. This gives us a set of readings that can be treated as phase-locked, despite being from different cards. We close with a note that this synchronization algorithm also compensates for any differences in phase caused by slightly different cable lengths to the shared antenna for the two cards, or by phase unbalance in the splitter, as these are subsumed by the constant γ phase shift of Equation 1.

2.2 Autocalibration

Since each radio chain connects to a different RF oscillator, their constant phases differ despite being frequency-locked. In order to obtain useful phase-difference data, array processing systems must eliminate these phase differences. While these could be measured by splitting a reference signal along carefully-measured cables [31], or by transmitting a known signal from one antenna to the other antennas on the same AP [25], these approaches interrupt normal communication and become disruptive if multiple devices have to be calibrated, or if devices restart and require recalibration.

In this section, we propose a technique for automatically calibrating an AP $\mathcal{A}$ that has just been powered-up. We present our method in two stages: first, at the lower level,

Metric (η-score):

1. Find a normalizing constant k such that $\int k\mathcal{P}(\theta)\,\mathrm{d}\theta$ is one, and set $\mathcal{P}' = k\mathcal{P}$.
2. Construct a Gaussian *mask* $g_\alpha(\theta)$ with an expected value α and a variance according to the desired level of error tolerance. Set $\bar{g}_\alpha(\theta) = 1 - g_\alpha(\theta)$.
3. Calculate $\eta(\mathcal{P},\alpha) = \dfrac{\int g_\alpha(\theta)\mathcal{P}'(\theta)\,\mathrm{d}\theta}{\int \bar{g}_\alpha(\theta)\mathcal{P}'(\theta)\,\mathrm{d}\theta}$.

Figure 4: Scoring a pseudospectrum $\mathcal{P}$ arising from a trial combination of phase offsets against a bearing to the AP α. The output of this calculation, the η-score, feeds into the overall autocalibration algorithm.

we present a metric, *η-scoring*, for evaluating the degree of correspondence between a pseudospectra and a given bearing. Then we present the full algorithm, which uses η-scoring, cross-packet calibration and a "reset" mechanism to achieve resilience against multipath RF propagation (§2.2.2).

Requirements. The only information autocalibration needs is the locations of the APs participating in the system; these are assumed to be entered by the sysadmin who installs the AP into ArrayPhaser's database. η-scoring uses frames transmitted either by other APs, or by other mobile clients whose location is determined by a localization system.

2.2.1 *Scoring individual frames*

When a frame arrives at the autocalibrating AP $\mathcal{A}$, we use spectral techniques to construct an angle-of-arrival *pseudospectrum* [21]; an estimate of the power of the incoming signal versus bearing to the AP. Since we know the location of the transmitter and the receiver, we have a relatively accurate estimate of the direct-path bearing $\hat{\alpha}$, and thus we know in which direction we are likely to sense energy along the direct path. Since the pseudospectrum changes with different oscillator phase offsets, we use ArrayPhaser's η-scoring metric to find a set of phase offsets that produce a pseudospectrum with energy directed toward $\hat{\alpha}$. The careful reader will note that when an obstruction blocks the direct path to $\mathcal{A}$, this approach in isolation will miss the direct path and score a reflected path highly. We rely on the "outer loop" of the autocalibration algorithm (§2.2.2) to overcome this challenge.

To determine the extent to which a pseudospectrum $\mathcal{P}$ matches $\hat{\alpha}$, we seek a metric that increases in the presence of peaks towards bearing $\hat{\alpha}$, and decreases in the presence of peaks at bearings away from $\hat{\alpha}$. Since $\hat{\alpha}$ is an estimate, we also want the metric to be continuous with respect to bearing, so that $\mathcal{P}$'s peak does not have to *exactly* coincide with $\hat{\alpha}$.

By comparing the area under $\mathcal{P}$ in the vicinity of AP bearing α with the area under $\mathcal{P}$ toward all other bearings, the *η-score* $\eta(\mathcal{P},\alpha)$ shown in Figure 4 fulfills the above goals. Peaks in directions away from α or a high noise floor increase the denominator in Step 3, thus decreasing $\eta(\mathcal{P},\alpha)$. Figure 5 shows η for six different combinations of phase offsets for a linear three-antenna array. As desired, the η-score is highest

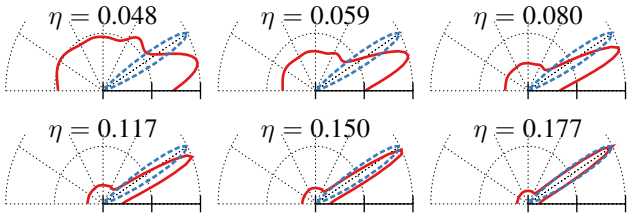


Figure 5: η-scoring different phase combinations that yield various pseudospectra (solid red curves) with energy directed towards and away from the true bearing (broken blue curve).

when a peak is present in the direction of $\hat{\alpha}$, and lower when there are peaks elsewhere.

The Gaussian ensures that pseudospectra with peaks close to α also get high η-scores, thus tolerating inaccuracies in the estimation of α. The width of the Gaussian should be chosen such that it matches the width of a peak in a pseudospectrum with most power directed toward the signal's true bearing. With increasing number of antennas, peaks get sharper, and thus the variance of the smoothing function should be decreased. For arrays with relatively few antennas, an experimental sensitivity analysis (§4.3.2) shows that a Gaussian with variance $\sigma^2 = 0.1$ works well.

2.2.2 *Autocalibration algorithm*

Upon receiving CSI from a frame[4] at $\mathcal{A}$, an N-antenna AP, we explore the $N-1$-dimensional phase offset space. An exhaustive search of all possible phase offset combinations on each received transmission would be very expensive for large N. To make this high-dimensional search tractable, we observe that an N-antenna array can be viewed as a number of smaller and overlapping L-antenna sub-arrays.

We begin with the algorithm for each L-antenna subarray $\mathcal{A}_i$. This maintains a *current best set* of phase offsets $\mathcal{B}_i$, which starts out empty. Each candidate E in $\mathcal{B}_i$ consists of a set of $L-1$ phase offsets $\phi_1^E,\ldots,\phi_{L-1}^E$ along with the η-score of the pseudospectrum resulting from applying those phase offsets to the frame's received CSI. We now consider every combination of the $L-1$ possible phase offsets with a *phase granularity* $\Delta\phi$ a candidate for insertion into $\mathcal{B}_i$. Given a candidate C, we now need to decide whether C should be inserted into $\mathcal{B}_i$, and if so, which element to evict from $\mathcal{B}_i$ once it grows beyond a certain *candidate population size* S.

We aim for each $\mathcal{B}_i$ to contain phase offset combinations that yield good η-scores, while still maintaining diversity so that if the ground-truth correct phase offset has a lower η-score, it will not be completely disappear from $\mathcal{B}$. To achieve this, we compare C with a randomly-selected $E \in \mathcal{B}$ and probabilistically replace E by C based on two ratios. The first, $\Delta\eta = \eta_C/\eta_E$, measures how much better C's η-score is than E's. The second ratio captures the impact of replacing E by C on diversity. We compute

$$\sigma^2_{\mathcal{B}_i} = \sum_{k=1}^{L-1}\sum_{E\in\mathcal{B}_i}\left(\phi_k^E - \bar{\phi}_k\right)^2 \tag{4}$$

[4]Possibly from multiple cards and combined as explained in §2.1.

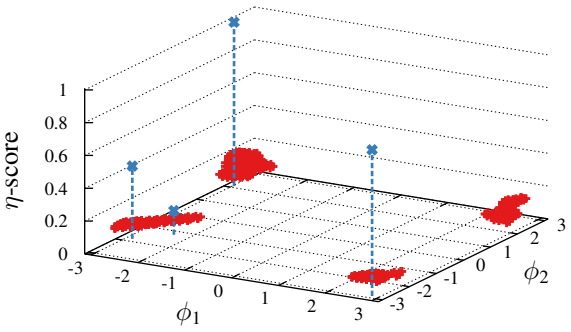


Figure 6: ArrayPhaser's cross-packet calibration clustering algorithm operating with packets from four different APs.

where $\bar{\phi}_k$ is the average phase offset for antenna pair k in $\mathcal{B}_i$. Then we compare $\sigma^2_{\mathcal{B}_i}$ with $\sigma^2_{\mathcal{B}_i \setminus \{E\} \cup \{C\}}$ by the ratio

$$\Delta\sigma^2_{CE} = \sigma^2_{\mathcal{B}_i \setminus \{E\} \cup \{C\}} / \sigma^2_{\mathcal{B}_i}. \quad (5)$$

The two ratios $\Delta\eta$ and $\Delta\sigma^2_{CE}$ give us information about how much better or worse C is than E. To determine the probability of replacement, $p_r(C,E)$, we first observe that, as the population stabilizes, $\Delta\eta$ and $\Delta\sigma^2_{CE}$ will generally be very close to one. We therefore want to increase $p_r(C,E)$'s sensitivity when the ratios are close to one, and cap it such that we discard obviously poor ratios and keep obviously good ratios. This can be achieved by applying a shifted exponential function to each ratio as shown in Equation 6:

$$p_r(C,E) = a\left(b^{\Delta\eta-1}/2 + b^{\Delta\sigma^2_{CE}-1}/2 - c\right) + 1/2. \quad (6)$$

By choosing a and b, we can adjust the sensitivity of the probability to small changes in $\Delta\eta$ or $\Delta\sigma^2$ when the ratios are close to one. We pick c to center the distribution at zero, meaning c is uniquely determined by the choice of a and b. By adding $1/2$ to the result, we normalize $p_r(C,E) = 0.5$ when C and E are equally good candidates.

The choice of a and b will depend upon the expected variation in $\Delta\eta$ and $\Delta\sigma^2$. While the expected variation in $\Delta\sigma^2$ is unpredictable across transmissions, and thus cannot be tuned for specifically, the variation in $\Delta\eta$ will vary depending on the array geometry, because different antenna geometries may produce different distributions of η scores as combinations of ϕ are searched. We have found that for a five-antenna uniformly linear array, $a = 10$, $b = 2.718$ and $c = 1$ yields good diversity in the population while still maintaining high η-scores across several deployments.

After performing this selection process, we have $N/(L-1)$ populations, each with the best candidates for one sub-array. We can now generate candidates for $\mathcal{B}$ by picking one candidate from each population, $c_i \in \mathcal{B}_i$ and subtracting the phase offset of the overlapping antenna in c_0 from all phase offsets in each candidate. The resulting candidate is probabilistically inserted into $\mathcal{B}$ in the same way as for each $\mathcal{B}_i$. We explore the effect of varying S and $\Delta\phi$ in §4.3.

Multi-packet operation. After this process completes, $\mathcal{B}$ contains a diverse set of phase offset combinations[5] that all yield high η-scores. We reduce this to a single combination with the observation that across frames from different transmitters, combinations close to the ground-truth phase offsets E^* appear in almost every population $\mathcal{B}$, and phase offsets due to multipath reflections vary randomly across populations from different transmitters. To exploit this, we introduce a *cross-packet population*, Ψ, into which all elements of each $\mathcal{B}^k$ (the final candidate phase offset set for received frame k) are placed such that $\Psi = \mathcal{B}^1 \cup ... \cup \mathcal{B}^K$ across K received packets. Then we apply standard clustering algorithms to Ψ, where the distance between two samples is determined by the unwrapped Euclidean distance of the phase offsets and the sample's η-score. Figure 6 shows the result of performing this clustering on a three-antenna array across 50 packets received from three transmitters.

To estimate the correct phase offset combination, we now find the cluster with the highest product of number of samples and average η-score, and the parameters of its centroid are used as the true phase offset combination. This metric dictates the height of the dotted line going through each identified centroid in the figure. This ordering rewards phase combinations that consistently give pseudospectra that have good correlation with the true AoA.

2.2.3 *Practical considerations for autocalibration*

AP placement estimation error. Autocalibration only has an estimate of the bearings of incoming signals. Because of this, the η-score will no longer reach a maximum for the true combination of phase offsets as the peak of the pseudospectrum for the true phase offsets will no longer align with the bearing that autocalibration is attempting to maximize against.

To counter this problem, we add another dimension to the search in each L-antenna sub-array. For each phase offset combination, we calculate the η-score for several smaller rotations of the normalized pseudospectrum $\mathcal{P}'$. Since we assume that the system knows the rotation and location fairly accurately, only a very narrow range of rotations ($\pm 5°$) need to be searched. Due to the Gaussian smoothing, we can also search quite coarsely, as even an approximate match will receive a higher η-score. When performing probabilistic insertion, we now also apply the rotation that was used in conjunction with the phase offsets we are iterating on top of. Since a rotation of $\mathcal{P}'$ is computationally trivial, recalculating the η-score for a small number of rotations is a sufficiently inexpensive operation that it does not significantly impact the running time of the autocalibration algorithm.

Calibration reset. Once autocalibration has terminated, the AP participates in a larger, multi-AP system along with other nearby APs. Suppose this system is a location tracking system and estimates a client's location as $\mathbf{x}$. The calibrated AP $\mathcal{A}$

[5] Several phase combinations can yield peaks toward a given bearing depending on the multipath environment and the antenna geometry.

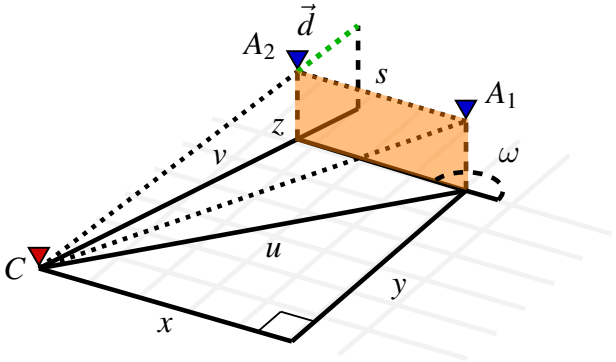


Figure 7: A transmitter C's transmission to a linear array with antennas A_1 and A_2.

measures the degree to which it agrees with that location by evaluating its pseudospectrum at the bearing corresponding to $\mathbf{x}$, $\theta_{\mathbf{x}}$. If across many clients, $\mathcal{A}$ consistently disagrees with the location system's estimate, it is likely that $\mathcal{A}$ has been miscalibrated. ArrayPhaser thus resets the calibration and reruns autocalibration. This alleviates the event that a burst of packets from a far-away AP or client miscalibrates $\mathcal{A}$.

2.3 Elco: Elevation compensation

Many array processing methods are based on detecting the additional distance traveled by an incoming signal between the different antennas of the array, as this added distance varies with the transmitter's bearing to the array.

For a transmitter C broadcasting to a two-antenna array, we have the situation in Figure 7. In two dimensions, if $s \ll u$, then u and v can be considered to be parallel. Consequently, the additional distance traveled by the signal to reach A_1 can be approximated as $d = u - v$. This additional distance d introduces a phase offset in the received signal which varies as $2\pi d/\lambda$, where λ is the wavelength of the incoming signal.

When there is an elevation difference between transmitter and AP, both the azimuth and elevation cause a phase shift of the incoming signal. In three dimensions, the added distance can be represented by a vector, $\vec{d}$ whose length $|\vec{d}|$ is the additional distance traveled by the signal due to azimuth and elevation. For the purposes of two-dimensional AoA estimation, the phase added by the additional vertical distance that the signal travels introduces an error term, because the transmitter's azimuthal bearing to the AP no longer solely determines the phase difference. The phase error introduced by a 1.5 meter height difference for different azimuths and radial distances is shown in Figure 8.

Autocalibration, as described in §2.2, uses pseudospectra to determine the phase offsets for the radio chains of an array. The more accurate the pseudospectra are, the better the estimates of the phase offset will be. Without compensating for elevation error, autocalibration could compute phase offsets with an error on the order of tens of degrees if transmitters were relatively close to the AP. When the AP and transmitter are known to be located at different heights, we therefore want to counter the phase introduced by the elevation before doing autocalibration, so that the pseudospectrum better

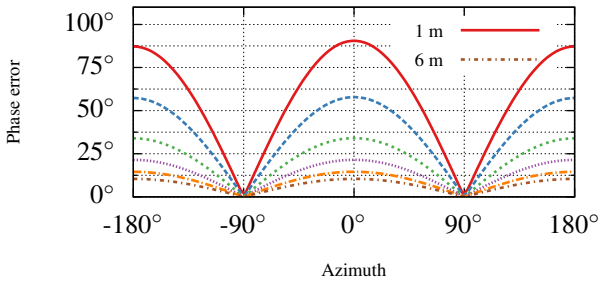


Figure 8: Phase error induced by an AP-client elevation difference of 1.5 meters as the radial distance and bearing to the client varies. Distances are shown in 1 m increments.

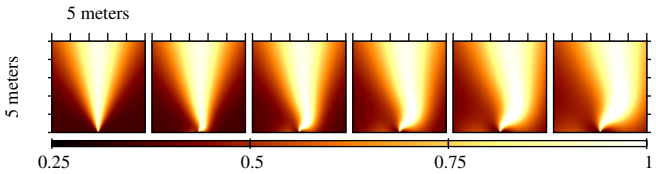


Figure 9: Pseudospectrum map with elevation compensation for height differences of 0–2.5 m in 50 cm increments.

reflects the chosen phase offsets.

It can be shown that the part of $|\vec{d}|$ introduced by elevation is $\psi = \left(\sqrt{(u + s/u \cdot (x\cos\omega + y\sin\omega))^2 + z^2} - \sqrt{u^2 + z^2}\right) \times \left(1 - u/\sqrt{u^2 + z^2}\right)$. This error can be negative, causing a phase derotation. This happens when the transmitter is closer to A_1 than A_2, and the signal travels *farther* to reach A_2.

Based on the error term $p = 2\pi\psi/\lambda$, we construct a *height compensation steering vector* for a candidate position $\mathbf{c}$ relative to an N-antenna array:

$$\mathbf{q}(\mathbf{c}) = \begin{bmatrix} 1 & e^{jp} & e^{j2p} & \cdots & e^{j(N-1)p} \end{bmatrix}^{\mathsf{T}}.$$

We multiply $\mathbf{q}(\mathbf{c})$ with the received signal $\mathbf{y}$ to get a new, compensated signal for the array, $\mathbf{y}'$. We can then use array processing algorithms as if the client were level with the AP.

We note that the error depends on all of x, y, *and* z, or alternately, both azimuth and elevation angle, which makes the resulting pseudospectra three-dimensional. While it is possible to adapt the η-scoring metric to handle multidimensional pseudospectra, this becomes prohibitively computationally expensive. Instead, we consider the η-score of such a 3D pseudospectra to be the η-score of the slice corresponding to the vertical angle between transmitter and the AP. This is quickly computed by finding the height compensation vector as above for the known location of the transmitter, applying it to the incoming signal to produce $\mathbf{y}'$, and then using standard AoA methods on $\mathbf{y}'$ to produce a full 2D pseudospectrum.

To illustrate the impact elevation compensation has on an AP's pseudospectrum, consider Figure 9. The first figure is a pseudospectrum as produced without elevation compensation, projected onto a two-dimensional map. As expected, the probability is equal for all points with the same bearing to the array. The following plots show the location probabilities for each location for the same signal, but compensating for in-

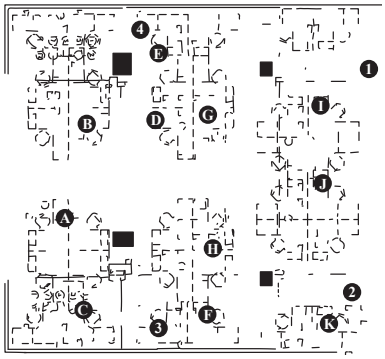


Figure 10: Map of the office environment used for the following experimental evaluation. The office measures 2,055 square feet and is populated with numerous partition walls and concrete pillars (denoted by black rectangles). We denote AP locations by circled numbers, while circled letters denote mobile test locations.

creasing height differences. A significant distortion is clearly visible at locations near the AP as the height increases.

3. IMPLEMENTATION

We have implemented ArrayPhaser on Intel Next Unit of Computing devices with Intel Wireless Link 5300 802.11n MIMO NICs. Using this commodity hardware we have access to 802.11 CSI readings [9], which enables us to extract per-subcarrier phase information. Due to firmware limitations, we were only able to get reliable phase readings at 5 GHz.

System data flow. The APs are set up in promiscuous wireless monitoring mode, and forward all 802.11 headers and CSI readings back to a central server over Ethernet backhaul. The server then arranges these into groups based on timing and header information to determine which readings belong to the same transmission. If the server detects that it has received multiple CSI readings from different cards on the same AP, it performs multicard merging as described in §2.1 and replaces them with the merged reading. Upon receiving a CSI reading from an uncalibrated AP, the server initiates the single-packet autocalibration algorithm from §2.2.2, while compensating for known elevation differences (§2.3). After a sufficient number of such packets have been processed for an AP, the system performs the multipacket clustering described in §2.2.2, after which the AP becomes operational.

4. EVALUATION

Methodology. We experiment in both a busy, uncontrolled 2,055-square foot open-plan office shown in Figure 10 and the controlled anechoic chamber environment of Figure 11. The office is populated with concrete pillars as shown, as well as numerous wooden partition walls throughout. Co-channel interference and overlapping channel interference are also present in the office. To evaluate Elco, our office APs are mounted on the ceiling, and so the environment is a partial line-of-sight environment: non-line-of-sight when a partition or concrete pillar blocks the mobile's path to an

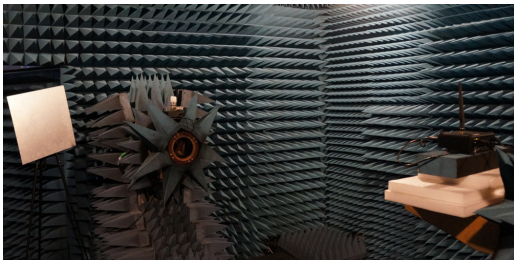


Figure 11: AP (far) and client (near) setup in anechoic chamber with multipath reflector.

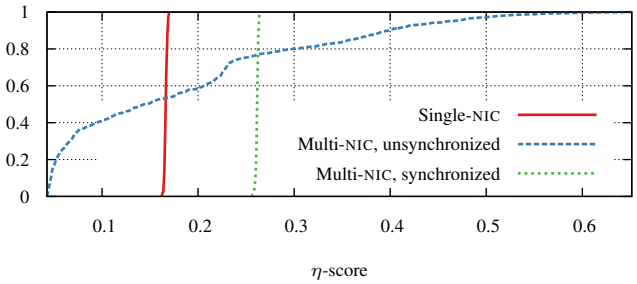


Figure 12: η-scores of a transmitter at 210° with and without multi-NIC operation in an anechoic chamber.

AP. As a result of the firmware limitations mentioned in §3, ArrayPhaser on the Intel 5300 NICs can only operate at 5 GHz. Since higher-frequency signals are subject to worse multipath reflections from walls and nearby objects, the carrier frequency we chose is in the most challenging band in the WiFi spectrum.

Evaluation roadmap. We begin with measurements of the accuracy of multicard operation (§4.1) and Elco (§4.2) in isolation and in the anechoic chamber. We then evaluate autocalibration (§4.3) and report end-to-end results from a localization system built atop ArrayPhaser (§4.4), both in the office.

4.1 Multicard operation

In order to test multicard operation in isolation, we perform manual cable calibration on an AP (§2.2) and place the AP with a client in the interference-free anechoic chamber where the multipath environment can be completely controlled. By placing the AP and the client at the same height, we also remove any error that elevation differences could introduce. Our test setup is shown in Figure 11: the metal plate is used to introduce a multipath reflection in later experiments. We average over 1,000 frame transmissions in each experiment.

Line-of-sight. We present data from a single, three-antenna NIC, two three-antenna unsynchronized NICs, and two three-antenna NICs with ArrayPhaser's inter-NIC synchronization (§2.1). Figure 12 shows η-score distributions across frames from each of these three experiments with the client at 210° to the AP's broadside. Since the experiment took place in a controlled environment, we expect very little variation in the score over time. While this holds for both single-NIC and synchronized multi-NIC operation, we see that for un-

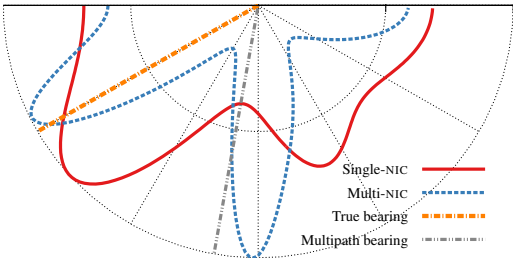


Figure 13: Pseudospectra with a transmitter at 210° and a multipath reflector at 100° from a single-NIC, three-antenna array and a multi-NIC, five-antenna array.

synchronized multi-NIC operation, the η-score varies widely across frames as the oscillators of the two cards drift relative to each other. We also note that the η-score is higher for multi-NIC operation than single-NIC. This is due to sharper pseudospectra produced as number of antennas increases.

Multipath. Another advantage of a larger array is the ability to resolve multiple arriving signals. Figure 13 shows a similar experiment, but with an additional signal arriving at 100° to the array's broadside. This skews the three-antenna pseudospectrum, whereas the five-antenna pseudospectrum clearly distinguishes the two incoming signals.

4.2 Elco: Elevation compensation

To demonstrate the need for Elco, we introduce a known elevation difference of 26 cm between the client and the NIC AP in the same setup as above without the multipath signal. At a distance of two meters, this elevation introduces a very small, but noticeable, error into the pseudospectrum; where the pseudospectra for the level transmissions had their peak 1.2° away from the true bearing, the peak for the uncompensated, elevated transmissions are off by 3.7°.

We now apply Elco to compensate for the elevation difference. To produce a corrected pseudospectrum, we evaluate the pseudospectrum map on the points of a circle centered at the AP and with a radius equal to the distance to the transmitter. After this compensation, the peak of the pseudospectrum is located 2.8° from the true bearing. While this may seem negligible, a 1° difference in bearing at a distance of 2 m is equivalent to a localization error of 3.6 cm. As the height difference increases, this error term grows quickly and Elco becomes increasingly important. We explore the larger-scale implications of elevation compensation in the end-to-end evaluation in §4.4.

4.3 Autocalibration

To evaluate how well autocalibration functions in an uncontrolled office environment, we place four multi-NIC ArrayPhaser APs in the ceiling of our office environment. The true phase offsets were determined by applying manual calibration using an RF signal splitter as described in §2.2. Except where noted, the APs were all positioned at the same height so that Elco could be disabled. We also present calibration error separately for single-NIC and multi-NIC operation to show the performance of the calibration process in isolation

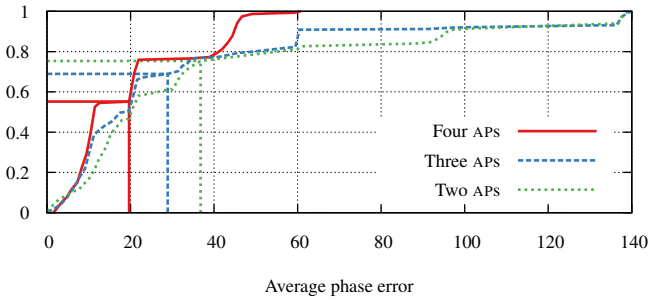


Figure 14: Average single-NIC phase error as the number of calibration transmitters varies. Vertical lines indicate mean.

without relying on the correctness of multi-NIC synchronization. The interplay between autocalibration and multi-NIC synchronization is discussed further below.

4.3.1 *Number of APs*

One of the most important factors for autocalibration is the number of distinct signal sources, as increased path diversity helps eliminate incorrect phase offset combinations caused by persistent multipath fading or interference. In Figure 14, we use the same four-AP setup, running autocalibration with differently-sized subsets of APs enabled. Accuracy decreases with transmitter diversity because the clustering at the end of the autocalibration process only works if the true phase offset combination is the only one with samples present in every population. As the number of distinct transmitters decreases, incorrect phase offset combinations are more likely to be present in a sufficient number of populations that they are considered to be more likely candidates than the true phase offset combination. We obtain good calibration accuracy with four, and acceptable accuracy with three active APs.

4.3.2 *Sensitivity analysis: Search parameters*

In §2.2, we also mention a number of parameters that can affect the performance of autocalibration; principally the number of transmitters, the number of transmissions, and the population size S. In addition, the granularity at which the phase offsets are searched when generating candidate phase combinations, $\Delta\phi$, clearly affects the accuracy achievable through calibration. Figure 15 on p. 9 shows how modifying each parameter affects the accuracy of autocalibration.

Population size. Recall from §2.2 that population size S determines the number of phase combinations that survive each iteration. As S increases, we retain more combinations from each packet, giving the final step of autocalibration and clustering more samples to work with. Results show that increasing the population size S past 64 increases the number of samples for the true combination. When we then cluster at the end of the autocalibration process, the cluster near the true combination will have more points, and the end result becomes more accurate.

Step size. The step size (measured in degrees) that we try for each phase offset, $\Delta\phi$ affects the precision autocalibration achieves. As the steps become coarser, the phase difference

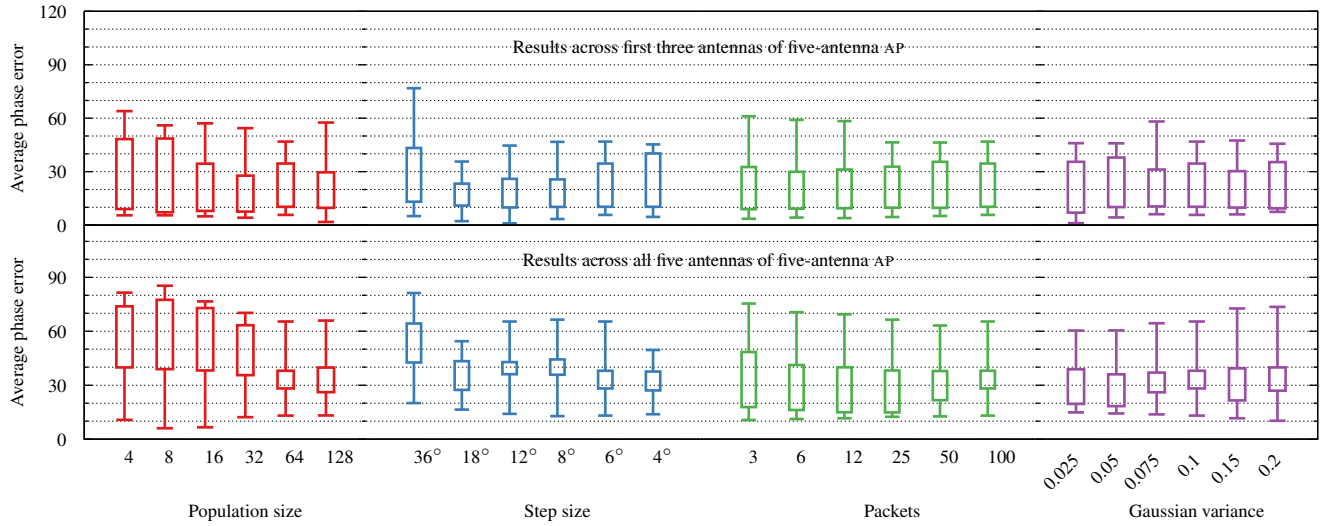


Figure 15: Average phase error as autocalibration parameters are varied. Top row shows single-NIC error, bottom row shows multi-NIC error. The non-variable parameters are held at 64 survivors, 100 packets, 60 steps and $\sigma^2 = 0.1$. Boxes denote the 25th and 75th order statistics, while whiskers denote the 5th and 95th percentile order statistics.

between the true offsets and their closest step will increase, and so will the overall error. However, below a step size of 18°, noise in the calibration process will surpass the step error and we will reach a point of diminishing returns, for both single-NIC and multi-NIC measurements. This correlates well with the lowest observed phase errors across all runs.

Number of frames. We measure the effect of changing the number of frames that the autocalibration algorithm waits for before performing clustering and picking the final phase offset combination. As this number is lowered, we would expect to see a similar degradation to what is observed as the number of transmitters decreases; fewer samples means incorrect phase offset combinations have a greater chance of creating winning clusters. We observe this trend in Figure 15—as we sample more transmissions, the true phase combinations become more numerous in the final population, and their cluster is more likely to dominate.

Gaussian variance of η-score. Finally, we measure the effect of changing the width of the Gaussian curve that the η-score metric correlates against, as described in §2.2.1. We observe low sensitivity to the width of the Gaussian and good performance at our chosen value ($\sigma^2 = 0.1$).

4.3.3 Impact of synchronization on autocalibration

From the lower half of Figure 15, autocalibration accuracy lessens for antennas on a non-primary NIC. By plotting the distribution of absolute phase error for the phase offsets for antennas on the primary NIC and on the secondary NIC separately for a five-antenna, dual-NIC array, we explore the error multi-NIC synchronization introduces.

Figure 16 shows the distribution of phase errors across 300 calibration runs on the four APs in our testbed. The mean phase error increases by almost 20° for the antennas on the secondary NIC. While this is a noticeable error, the fidelity gain from more antennas often outweighs the noise

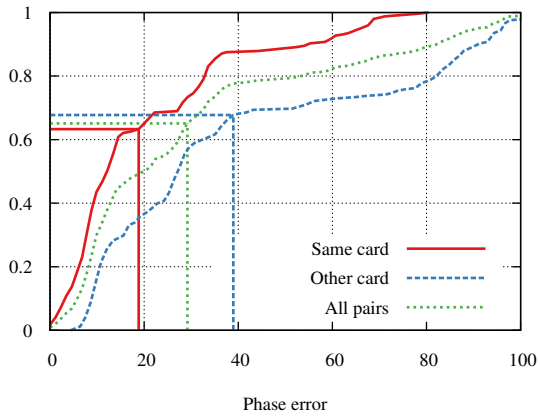


Figure 16: Phase error distribution across antenna pairs for antennas pairs on the same card versus different cards

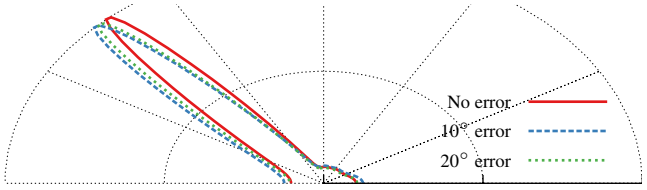


Figure 17: Pseudospectra for varying average phase errors

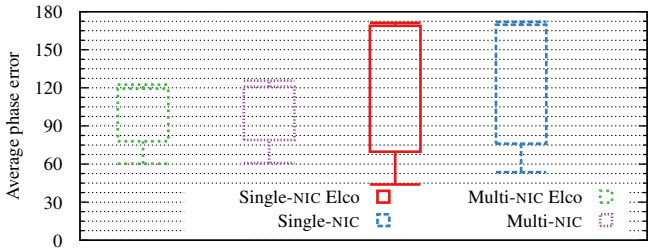


Figure 18: Autocalibration phase error for elevated AP with/without Elco with two calibration sources. Box/whisker meanings are as in Figure 15.

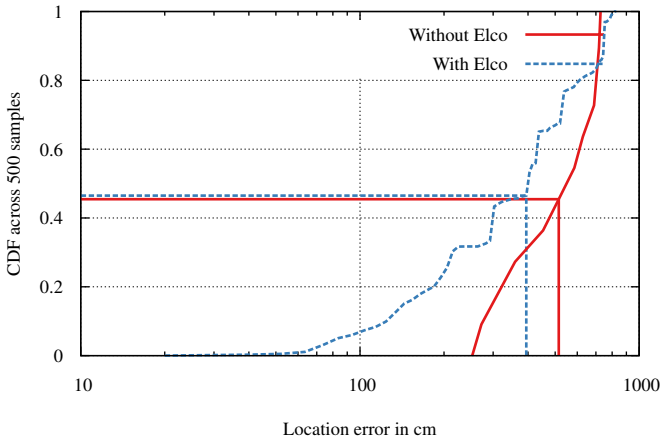


Figure 19: End-to-end localization system error with four multi-NIC five-antenna APs, with and without Elco. Vertical and horizontal lines denote the mean and the order statistic of the mean respectively.

synchronization introduces. To give an idea of the impact of such a phase error, consider Figure 17 where we show a pseudospectrum produced with the correct phase offsets, as well as pseudospectra where each phase offset is perturbed by a random error of $\pm 10 \pm 5$ or $\pm 20 \pm 5$. While the error is noticeable, the overall shape of the pseudospectrum is maintained even for an average error of 20°.

4.3.4 Impact of elevation on autocalibration

As we point out in §2.3, elevation differences between a client and an AP introduce a phase error in the received signal, which may skew the phase offsets produce by the autocalibration algorithm. To determine the efficacy of Elco for the purposes of calibration, we introduce another AP in our testbed that is located 140 cm below the other APs with distances to two other APs of 1.5 m and 6.3 m. Figure 18 shows autocalibration phase error with and without Elco enabled for this one AP. We observe that Elco decreases the 25% percentile error for single-NIC results, but unfortunately the improvements are lost in the noise for multi-NIC results.

4.4 Application: Localization

In order to determine whether ArrayPhaser truly enables practical phased-array signal processing on commodity hardware, we implement a angle-of-arrival based location tracking system similar to ArrayTrack [31] on top of ArrayPhaser. In this context, the server constructs pseudospectra from phase-difference information obtained from each AP's reading in a transmission group and uses this information to triangulate the transmitter's location. This is done by finding the location on the map that maximizes $\prod P_i(\theta_i)$; the combined pseudospectrum probability across APs for the candidate location's bearing to each AP. All APs in the deployment are first calibrated using autocalibration. Figure 19 shows the end-to-end localization error of this system when deployed on ArrayPhaser with four five-antenna APs in our office environment. Test locations are marked with letters in the map (Figure 10). We replay the same trace with and without Elco enabled to evaluate the degree to which it improves horizontal localization accuracy by compensating for differences in client elevation[6]. Results show that ArrayPhaser enables practical indoor location estimation on off-the-shelf hardware with little configuration and no training. Elco further improves mean accuracy by about one meter, and best-case accuracy from two meters to less than a meter. The median accuracy also improves by one meter, representing a 20% improvement on the median location accuracy due to elevation compensation alone.

5. RELATED WORK

In this section we survey prior work in the three types of radio systems where ArrayPhaser is useful: indoor location systems, radio imaging systems, and MU-MIMO systems that employ transmit beamforming from AP to mobiles.

5.1 Indoor location systems

ArrayPhaser's functionality is most directly applicable to indoor location systems, and there are many different such systems.

Ultrasound and light-based methods. Active Badge [29], Bat [30, 10], and Cricket [15] systems are among the pioneers in the ultrasound and ultrasound/RF area, but do require specialized hardware and per-room infrastructure. ArrayPhaser's elevation compensation is inspired by, and begins with, similar three-dimensional geometrical reasoning as the Cricket Compass [16] the Lighthouse location system [19], and Pharos [11] use, but advances the state-of-the-art by compensating for and estimating elevation using purely two-dimensional signal readings that typical AP radio antenna arrangements generate.

RF-based methods. The RADAR system [3, 4] pioneered this popular approach based on AP radio signal strength maps, with further work refining the approach probabilistically [35], through perturbation [34], and by exploring the use of cross-technology interference [8]. To build the radio map without human involvement, follow-on work uses crowdsourcing,

[6]For this experiment, the client was held at a constant, known height, and the localization system was told what height difference it would need to compensate for.

step counting, smartphone inertial sensors [18, 27, 33], and GPS [6]. Other approaches leverage ray tracing [7] or human intervention (either by manual location input [14], or a user's spinning movement [22]) to obviate the need for manual radio map construction. These RF-based methods, however, focus almost exclusively on two-dimensional location, and can benefit from part of ArrayPhaser's elevation estimation techniques.

RF angle-of-arrival based methods measure the two-dimensional bearing at which a transmission arrives at the AP [13, 31]. Further work refines these methods to use the smartphone's inertial sensors to estimate the line-of-sight path to the AP [23]. ArrayPhaser's autocalibration can augment these systems to make them even more practical, and ArrayPhaser's elevation compensation and multicard operation components make them even more functional.

RF time-of-flight based methods such as PinPoint [36] measure the time it takes an RF signal to travel between mobile and AP. These techniques also benefit from the elevation compensation and estimation algorithms of ArrayPhaser.

Spot localization. PinLoc [24] places the mobile in one of several pre-defined, geographically separated spots by statistical clustering and classification of 802.11n CSI information. ArrayPhaser's multicard operation can offer these systems the ability to classify based on five or more antennas instead of three, potentially improving system performance.

5.2 Radio imaging

Also known as device-free localization and passive radar, this work uses RF measurements to localize humans and objects moving about in space, without the need for them to be carrying RF transmitters. WiSee [17] uses Doppler shift analysis combined with MIMO spatial signatures to analyze human motion, and so ArrayPhaser's multi-antenna processing design can complement it. Wi-Vi [2] uses inverse synthetic aperture radar combined with angle-of-arrival measurements using the MUSIC algorithm [21] to track users through walls, and so all three parts of ArrayPhaser can add direct benefit.

Other such systems use signal strength and/or Doppler shift measurements only. Ichnaea [20] uses statistical processing on received signal strength (RSS) measurements, and MonoStream uses image processing on the magnitude information of 802.11n CSI data to recognize trained patterns when users stand at various locations. Chetty *et al.* [5] describe a single-antenna passive bistatic radar system that can track targets through a wall. Here ArrayPhaser is of no immediate benefit, but in general, coherent combination of data from multiple antennas boosts fidelity, and so with this addition, ArrayPhaser can offer complementary benefits to these systems.

5.3 Transmit beamforming

Muti-user MIMO transmit beamforming systems like Argos [25] also need to perform phase calibration. To do so, Argos sends from one antenna on the WARP FPGA-based AP while receiving on the others. But this approach is not directly applicable to current commodity NICs, as they usually do not support transmitting on one antenna while receiving on the other antennas simultaneously. Nevertheless, a similar calibration technique can be applied on APs with multiple NICs; the cards can take turns transmitting and receiving for a brief period after the AP first comes online in order to gather information similar to that obtained in Argos' calibration phase. However, this calibration technique has a number of drawbacks that make it problematic. First, it requires multiple NICs to be present for it to work. As the current trend is for the number of antennas per NIC to increase, the need for multiple NICs is declining, and having a hard requirement on additional hardware for the sole purpose of calibration is undesireable. Second, using an APs own transmissions for calibration means additional shielding is required between the antennas and the antenna ports on the NICs. Any signal leaking between the two will severely skew calibration results. Finally, with dense antenna geometries, and particularly for linear arrays, the antennas themselves will add multipath reflections and obstruct line-of-sight signal propagation, introducing additional errors into the calibration results. ArrayPhaser's autocalibration mechanism is not subject to these problems, but does require cooperation with other APs or mobile clients to function, which is not the case for Argos-like calibration.

6. CONCLUSION

We have described ArrayPhaser, a system that allows commodity WiFi APs with or without minor cost-effective hardware modifications to become phased array signal processing platforms capable of high-fidelity array signal processing. ArrayPhaser contributes novel design techniques for (1) multi-NIC operation, (2) AP-client elevation difference compensation, and (3) autocalibration. We have experimentally measured the fidelity of ArrayPhaser's phase measurements, and in an application study we have demonstrated improved end-to-end indoor location accuracy Finally, we have outlined three distinct kinds of wireless systems that benefit from ArrayPhaser (§5).

7. REFERENCES

[1] H. Abdel-Nasser, R. Samir, I. Sabek, and M. Youssef. MonoStream: A minimal-hardware high accuracy device-free WLAN localization system. In WCNC, 2013.

[2] F. Adib and D. Katabi. See through walls with WiFi! In SIGCOMM, 2013.

[3] P. Bahl and V. Padmanabhan. RADAR: An in-building RF-based user location and tracking system. In INFOCOM, 2000.

[4] P. Bahl, V. Padmanabhan, and A. Balachandran. Enhancements to the RADAR user location and tracking system. Technical Report MSR-TR-2000-12, 2000.

[5] K. Chetty, G. Smith, and K. Woodbridge. Through-the-wall sensing of personnel using passive bistatic radar at standoff distances. IEEE *Trans. on Geo. and Remote Sensing*, 50(4):1218–26.

[6] K. Chintalapudi, A. Iyer, and V. Padmanabhan. Indoor localization without the pain. In *MobiCom*, 2010.

[7] A. Eleryan, M. Elsabagh, and M. Youssef. AROMA: Automatic generation of radio maps for localization systems. In WINTECH, 2011.

[8] Y. Gao, J. Niu, R. Zhou, and G. Xing. ZiFind: Exploiting cross-technology interference signatures for energy-efficient indoor localization. In INFOCOM, 2013.

[9] D. Halperin, W. Hu, A. Sheth, and D. Wetherall. Tool release: Gathering 802.11n traces with channel state information. SIGCOMM CCR, 41(1), 2011.

[10] R. Harle and A. Hopper. Deploying and evaluating a location-aware system. In *MobiSys*, 2005.

[11] P. Hu, L. Li, C. Peng, G. Shen, and F. Zhao. Pharos: Enable physical analytics through visible light based indoor localization. In *HotNets*, 2013.

[12] T. Laakso, V. Valimaki, M. Karjalainen, and U. Laine. Splitting the unit delay. IEEE *Sig. Proc. Mag.*, 13(1):30–60, 1996.

[13] D. Niculescu and B. Nath. VOR base stations for indoor 802.11 positioning. In ACM *MobiCom*, 2004.

[14] J. Park, B. Charrow, D. Curtis, J. Battat, E. Minkov, J. Hicks, S. Teller, and J. Ledlie. Growing an organic indoor location system. In *MobiSys*, 2010.

[15] N. Priyantha, A. Chakraborty, and H. Balakrishnan. The Cricket location-support system. In *MobiCom*, 2000.

[16] N. Priyantha, A. Miu, H. Balakrishnan, and S. Teller. The cricket compass for context-aware mobile applications. In *MobiCom*, 2001.

[17] Q. Pu, S. Gupta, S. Gollakota, and S. Patel. Whole-home gesture recognition using wireless signals. In *MobiCom*, 2013.

[18] A. Rai, K. Chintalapudi, V. Padmanabhan, and R. Sen. Zee: Zero-effort crowdsourcing for indoor localization. In *MobiCom*, 2012.

[19] K. Römer. The Lighthouse location system for smart dust. In *MobiSys*, 2003.

[20] A. Saeed, A. Kosba, and M. Youssef. Ichnaea: A low-overhead robust WLAN device-free passive localization system. IEEE *J. on Sel. Topics in Sig. Proc.*, 8(1), 2014.

[21] R. Schmidt. Multiple emitter location and signal parameter estimation. IEEE *Trans. on Antennas and Propagation*, AP-34(3):276–80, 1986.

[22] S. Sen, R. Choudhury, and S. Nelakuditi. SpinLoc: Spin once to know your location. In *HotMobile*, 2012.

[23] S. Sen, J. Lee, K. Kim, and P. Congdon. Avoiding multipath to revive inbuilding WiFi localization. In *MobiSys*, 2013.

[24] S. Sen, B. Radunovic, R. Choudhury, and T. Minka. Spot localization using PHY layer information. In *MobiSys*, 2012.

[25] C. Shepard, H. Yu, N. Anand, L. Li, T. Marzetta, R. Yang, and L. Zhong. Argos: Practical many-antenna base stations. In *MobiCom*, 2012.

[26] J. K. Tan. An adaptive orthogonal frequency division multiplexing baseband modem for wideband wireless channels. Master's thesis, Massachusetts Institute of Technology, June 2006. pp. 43-47.

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

[28] J. Wang and D. Katabi. Dude, where's my card? RFID positioning that works with multipath and non-line of sight. In SIGCOMM, 2013.

[29] R. Want, A. Hopper, V. Falcao, and J. Gibbons. The active badge location system. *ACM Trans. on Information Systems*, 10(1):91–102, Jan. 1992.

[30] A. Ward, A. Jones, and A. Hopper. A new location technique for the active office. *IEEE Personal Communications*, 4(5):42–47, Oct. 1997.

[31] J. Xiong and K. Jamieson. ArrayTrack: A fine-grained indoor location system. In NSDI, 2013.

[32] Y. Yang, B. Chen, K. Srinivasan, and N. Shroff. Characterizing the achievable throughput in wireless networks with two active rf chains. In INFOCOM, 2014.

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

[34] M. Youssef and A. Agrawala. Small-scale compensation for WLAN location determination systems. In WCNC, 2003.

[35] M. Youssef and A. Agrawala. The Horus WLAN location determination system. In *MobiSys*, 2005.

[36] M. Youssef, A. Youssef, C. Rieger, U. Shankar, and A. Agrawala. PinPoint: An asynchronous time-based location determination system. In *MobiSys*, 2006.