If WiFi APs Could Move: A Measurement Study

Technical Report

Mahanth Gowda, Ashutosh Dhekne, Romit Roy Choudhury, and Srihari Nelakuditi

Abstract—This paper explores the possibility of injecting mobility into wireless network infrastructure. We envision WiFi access points on wheels that move to optimize user performance. Movements need not be all around the floor, neither do they have to operate on batteries. As a first step, WiFi APs at home could remain tethered to power and Ethernet outlets while moving in small areas (perhaps under the couch). If such systems prove successful, perhaps future buildings and cities could offer explicit support for network infrastructure mobility. This paper begins with a higher level discussion of robotic wireless networks – the opportunities and the hurdles – and then pivots by developing a smaller slice of the vision through a system called iMob. With iMob, a WiFi AP is mounted on a Roomba robot and made to periodically move within a $2\times2$ sqft region. The core research questions pertain to finding the best location to move to, such that the SNRs from its clients are strong, and the interferences from other APs are weak. Our measurements show that the richness of wireless multipath offers significant opportunities – even within a $2\times2$ sqft region, locations exist that are 1.7x better than the average location in terms of throughput. When multiple APs in a neighborhood coordinate, the gains can be even higher. In sum, although infrastructure mobility has been discussed in the context of Google Balloons, ad hoc networks, and delay tolerant networks, we believe that the possibility of moving our personal devices in homes and offices is relatively unexplored, and could open doors to new kinds of innovation.

Index Terms—Wireless, Robotic Networks, Infrastructure, Measurement

1 Motivation and Vision

The last 30 years have witnessed significant advancements in wireless networking, ranging from hardware improvements to breakthroughs in theory, algorithms, and protocols. In the recent years, however, there is growing agreement in the research community that gains from the lower layers (MAC and PHY) are saturating. Many are beginning to believe that the next “jump” in network performance will emerge from new ways of organizing networks [1]–[5]. In considering new network organizations, we explore the possibility of merging wireless networking with robotics. Specifically, we ask: what if network infrastructure of the future – WiFi APs, enterprise WLANs, cell towers – are empowered with the ability to move physically? In pursuit of this thought, we began surveying the current state of robotics, as well as the pros and cons of physically moving infrastructure (e.g., WiFi APs on wheels, or cell towers on drones). We make a few observations below.

(1) Infrastructure mobility may not be viewed as a one-size-fit-all solution, rather as a spectrum of opportunities illustrated in Figure 1. The opportunities range from centimeter scale antenna mobility to exploit multipath opportunities [6], to feet scale tethered mobility to evade wireless shadows and interferences, to full scale macro-mobility that minimize distance to clients. Network designers can choose to operate at different points on this spectrum, depending on user’s requirements, budget, applications, and psychological comfort.

(2) Mobility is expected to bring a new degree of freedom (DoF) to network design, but more importantly, this DoF compliments existing dimensions of wireless innovation. Techniques for power control, channel allocation, localization, topology control, can all benefit if APs have the ability to move, even in the scale of inches.

(3) The time scale of mobility can be regulated as necessary. Small scale mobility can be used to compensate for small changes in network conditions, while full scale mobility can be triggered occasionally when the system moves to a skewed state, or a strict QoS requirement is ordered. In cellular networks, for instance, quad-copters could occasionally fly out from cell towers and position themselves strategically to meet users’ demands – like a network cloudlet [2], [3]. Infrastructure mobility could evolve as an on-demand service, a cost-effective and scalable alternative to over-provisioning.

Of course, some basic questions arise.

(1) Is moving infrastructure really practical? Concerns on feasibility are valid, but could perhaps be alleviated by building the vision in small systematic steps. Advances in personal robotics, beginning from the popular Roomba [7] to the more recent quadcopters [8]–[11] are already mainstream. Hardware is rapidly becoming cheap and reliable – an Arduino based robot car chassis adequate for cradling WiFi APs is $16 today [12]. Based on where robotic technology stands today [13], it is certainly not the fundamental barrier to infrastructure mobility.

Questions on the architectural aspects are certainly more relevant, such as maintaining power/Internet connectivity to a mobile AP, tangling wires, awkward moving objects on the floor, etc. However, we do not envision an all-at-
once technology deployment, rather we intend to activate functionalities incrementally. As a first step in home settings, a mobile WiFi AP might just remain tethered to power and Ethernet, and only move in small spatial scales (say, under the couch or study table). In enterprises, airports, and hotels, the APs may also be tethered, but they could move in a coordinated manner (like a joint topology control problem) orchestrated by the cloud. Moreover, the AP movements need not be continuous; the time scales could slowly become more frequent as the system matures and gains social acceptance. Of course, facilities management and other logistical/policy questions will arise, but we believe they can be mitigated if the core performance gains are compelling.

(2) How compelling are the gains? While the answer obviously depends on numerous factors, the high level message is that the upper bound can reach 3x and more, compared to the static case. For example, in home environments, median throughput from 2 feet of mobility is 2x for single clients, with the possibility of reaching 4x in 20% of the cases. With multiple homes, if APs coordinate to avoid mutual interference and optimize client SNR, median gain in overall network throughput can be 1.77x or more.

It is crucial to recognize that the performance gains are not obtained by moving the AP close to one client – with multiple clients associated to an AP, moving close to one client will adversely affect others. The gains we observe actually arise from finding appropriate AP locations from which the SNRs to all its clients are strong. This is feasible due to rich spatial diversity in indoor environments, i.e., there exists certain nearby locations from which many clients experience strong channel conditions. In fact, the best AP locations could also experience lower interference from other APs and clients, enabling greater spatial reuse. On the other hand, blindly chosen AP locations can fail to leverage these benefits, resulting in far inferior performance.

iMob demonstrates the ability to improve throughput to 5+ clients simultaneously. If too many more clients are active simultaneously, iMob can choose the top-K demanding clients and optimize their performance without affecting the others. If no solution is feasible, i.e., no AP location is able to satisfy the requirements, iMob could reduce the value of K. In the worst case, iMob will degenerate to a “static” AP and behave exactly as today’s WiFi technology.

(3) Why move? Why not use MIMO, beamforming, or other software techniques? While these PHY layer techniques also leverage spatial diversity, mobility is still complimentary. Micro-shadowing scenarios are highly common in indoor environments [14], [15] – moving slightly can appreciably increase the rank of the channel matrix, resulting in higher MIMO gains. Our measurements confirm 3x:3 MIMO gains with today’s 802.11n interfaces. Further, interference at the MAC layer is a function of energy, implying that AP1 would need to move out of AP2’s carrier sensing range to enable spatial reuse. With beamforming/MIMO, AP1 will still sense AP2 and will defer communication. However, if AP1 could physically move out of AP2’s range, or if AP1 and AP2 could jointly move to become “independent”, system performance can improve further. Lastly, mobility and beamforming can be performed jointly to harness the best of both worlds.

The above is a high level vision (and qualitative arguments) aimed at motivating the overall research direction. We published a part of this vision in a workshop paper [16], along with toy measurements on USRPs using 1 MHz frequencies. This paper focusses on systematically characterizing the research landscape in real environments, and then builds a completely functional robotic AP system – iMob – using off the shelf 802.11n hardware. The key technical modules we develop are described next.

2 iMob: Robotic WiFi Access Points
As a first step of the broad vision, we focus on small scale mobility in homes, in a way that is minimally disruptive to the established notions of a WiFi network. The iMob system we develop will allow WiFi APs to move on wheels while being tethered to the same power and Ethernet cable, as is currently used in most homes. Ideally, the APs could be placed away from human movement, such as underneath a couch or a side-table, or at the corner of a room. In this setting, the iMob system will be tasked to offer performance gains to client devices. The main technical components we develop are as follows:

- We begin by measuring the upper bound on performance gain achievable through feet-length mobility of WiFi APs.

1. This is anyway the case in many homes, given that network devices and wires are typically hidden from eyesight.
These gains are measured using a testbed of 8 laptops mounted on Roomba robots – the laptops run 3x3 MIMO using Intel 5300 802.11n cards. Using one of the devices as a mobile AP and others as scattered clients, we find the optimal AP location from which system performance is maximized. Besides serving as an Oracle, these measurements also offer insights into the nature of the gains, ultimately guiding the design of a real-time robotic networking system.

- We cross-check the Intel card results with USRPs and Atheros cards and verify that the gains scale across heterogeneous hardware (and not a function of our hardware idiosyncrasies).

- We then develop a practical iMob system in which the AP observes channel conditions and moves in real-time to the best estimated location. The motion planning algorithm uses insights from channel measurements, properties of the robot, and results from optimal stopping theory, to balance the tradeoff between exploration and exploitation (i.e., whether the AP should continue to explore more locations or should stop and perform remaining transmissions from its current location). This tradeoff naturally arises because the channel changes over space and time, and the AP does not possess the Oracle’s view.

- We also build a coordinated iMob system in which the cloud moves multiple interfering APs (e.g., in neighboring apartments or houses) to optimize performance. This is essentially a topology control problem, with physical mobility as a degree of freedom. Both signals and the interferences can now be controlled to optimize desired performance metrics.

- We evaluate single AP iMob in faculty homes, student apartments, and in our lab. Coordinated iMob is evaluated with 4 APs deployed across 2 floors in our engineering building. Experiments are designed to evaluate a range of parameters and scenarios, including throughput and fairness, MIMO gains, impact of “leash length”, impact of increasing number of clients, client mobility, etc. The overall gains are promising, and achievable without accurate prediction of wireless multipath and spatiotemporal channel variations. The inherent statistical nature of the environment offers viable opportunities.

3 Measurements

To characterize performance upper bounds with mobility, we will exhaustively move APs in small spatial granularities and pick the best location that optimizes a given metric – we call this the Oracle. We will then focus on understanding the nature of the gains, and utilize the insights to guide the design of a practical, real-time robotic WiFi system.

3.1 Experiment Platform and Methodology

Figure 2(a) shows a iMob AP assembled using a Roomba iRobot 2.1, a webcam, and a laptop equipped with Intel 5300 802.11n cards. The laptop is mounted on the iRobot and connected to it over the serial interface; it is also connected to a Microsoft live cam (attached in front of the iRobot) to guide its motion. The laptop acts as the controller for the whole system, sending motion commands to the robot (via the OSI interface), while also controlling the network interface for transmission/reception. 8 laptop clients were uniformly scattered at various locations and programmed to communicate back to the iMob AP.

Fig. 2. (a) A laptop and a webcam mounted on a Roomba to emulate an iMob AP. (b) Raster scan in a box while communicating to scattered client(s).

The robot’s mobility is confined within a 2x2 feet square region, demarcated by colored duck tapes pasted on the floor. We term this 2x2 feet square region as a spot. If the robot drifts out of the square box, the camera detects the color of the duck tapes and triggers a change in heading direction. These square regions are selected from realistic areas in homes and apartments, i.e., near cable connection outlets. The AP performs “raster scans” within the square box (Figure 2(b)) at a speed of 10 cm/sec – during the scan, the AP continuously sends around 200 packets/second, equivalent to 60 packets per 5cms. Transmissions are performed on regular OFDM, 3x3 MIMO at both 2.4GHz and 5GHz bands. Clients record the per-packet channel state information (CSI) for offline analysis [17], [18].

The experiments were conducted in 4 different settings: (1) Student-office referred to as Office. (2) Various corridors opening into the atrium of the engineering building, called Lab. (3) Single bedroom graduate student apartment, called Apartment. (4) Large single family home with APs placed in different rooms, called Home. In all cases, people moved naturally during the experiments, and clients scattered at realistic locations. Total measurements exceed 100 hours, generating 5TB of data.

Metrics: We evaluate performance in terms of data rates, throughput, and fairness. While the Oracle selects the location with best data rate, our baseline scheme reflects today’s static systems where the AP is placed at an arbitrary location near cable connection outlets. In light of this, the median performance among all locations inside the spot is treated as the baseline. Thus, the upper bound gain, for throughput say, is defined as:

\[ Gain = \frac{\max_{i} \text{throughput}}{\text{median}_{i} \text{throughput}} \]

where \( i \) denotes location \( i \) inside a spot to which the AP can move to. Of course, when we design the real-time iMob system (later in Section 4), the median gain is not known to the AP since continuous raster scans are impractical. Still, the iMob AP should park itself at “good” locations from which the performance exceeds the median. We will discuss these later; for now, we focus on characterizing the system’s upper bounds.
3.2 Characterizing Upper Bounds: Software Radios

Recall that upon arriving at a spot, the robot AP moves through the pixels within the spot transmitting a few packets from each pixel. Eight randomly scattered clients record the SNRs from every pixel in a spot – for each client this results in a SNR heatmap. If the AP moves through N spots, each client records N heatmaps. Figure 4 shows 4 heatmaps from 4 arbitrarily picked clients when the robot moved within a randomly picked spot. Darker shades in the heatmap indicate stronger SNR and the vice versa. We make two crucial observations:

1. Dominantly light colored spots, indicating that client is far away from the AP, have several pixels that are dark. This suggests that it is possible for an AP to significantly improve SNR to its client with centimeter scale mobility.

2. Spots that are dominantly dark, indicating that the client is close to the robot, has several pixels that are light. This suggests that it is possible for a robot AP to move a little and avoid being interfered by nearby APs, enabling parallel transmissions.

Figures 5(a-e) report the statistics from all spots for each client in all settings (Lab, Office, Apartment, Home). Figure 5(a) shows the CDF of the difference between maximum and median SNR from each spot. On average, this difference is at around 8 dB, implying that on any given spot, antenna mobility should offer appreciable gains to a client. Figure 5(b) shows the CDF of the difference between median and minimum SNR from each spot. Observe that for around 20% of the cases, the interfering signal can be suppressed by around 10 dB, just by moving the robot antenna to the pixel with minimum SNR.

To reason about throughput gains, we convert SNR to throughput using Shannon’s equation (of course, this produces the upper bound and the protocol overheads will certainly diminish gains). Figure 5(c) shows that median throughput gain is around 70% in the absence of interferers – this is essentially the reward for moving the AP and picking the right pixel in a spot. Figure 5(d) shows that the median throughput gain is 200% under interference, i.e., the robot AP selects a pixel that simultaneously maximizes its client’s SNR and minimizes the interference. The distribution also has a long tail, implying that in certain topologies, even little AP mobility can be effective.

Satisfying Multiple Clients

Figure 5(d) plots the throughput gains for different client densities and different settings – in a large home, with interferences from 3 neighboring APs, the throughput gain can be up to 150% for 5 active clients. Observe that 5 simultaneous clients is reasonably high density, since in reality, not all clients are active at the same time. If they are, we could optimize for the throughput hungry clients and still achieve substantial spectrum savings.

While micro-mobility is within one spot, recall that with mini-mobility APs have a longer leash (i.e., it can move to adjacent spots). Figures 6(a) and 6(b) plot the throughput gains without and with interference for varying number of clients under Mini-Mobility. With 5 active clients and co-channel interference (from 3 surrounding interferers), the throughput can increase to around 2.5x.

Channel Non-Monotonicity

Intuitively, it might seem that the client needs to be brought very close to AP for a 3x gain. Mini-mobility seems to be suggesting that this is not necessarily true, instead, carefully searching for a good nearby pixel may be comparable to blindly moving close to the client. We believe this could be a valuable intuition. The core opportunity arises from the fact that the indoor wireless channel has non-monotonicity, that is, some far away locations can be strong and some nearby ones can be weak.

To quantify this, we perform the following experiment on our measurement data. We position the AP at a random pixel $P$ near the client – let’s say the SNR at the client from this AP is $S_P$ dB. We now scan all spots in the entire building and pick the maximum SNR pixel from them, say $X_i$, and check whether this maximum SNR is within 90% of $S_P$. If so, we draw a line joining $P$ and $X_i$. Longer the line, stronger is the evidence of this opportunity. Figure 6(c) visualizes the scenario, corroborating the intuition that carefully searching local pixels can be as effective as blindly moving close to the client.
Dense AP Deployment

AP mobility is complimentary to AP density. We believe robot APs can be very cheap, and hence, all installed APs can be mobile. Thus, we compare between two schemes: (1) $K$ static APs installed at realistic locations and scattered clients associating to the strongest AP at any given time; (2) the same $K$ APs but each AP capable of mini-mobility. Figure 6(d) shows that throughput gains from min-mobility can be up to $2x$ with interference (and less without). Injecting mobility to a high density system can still be useful, so long as the density is not extremely high.

3.3 Characterizing Upper Bounds: Real WiFi Card

The experiments are designed around 8 questions – the first 4 focussed on the amount of performance gain, and the next 4 on understanding the nature of the gains.

1) With centimeter scale mobility, an AP might appreciably improve data rate to a given client.
2) With centimeter scale mobility, an AP can relocate to minimize interference from nearby APs/clients (potentially improving spatial reuse).

Assuming that the iMob AP is able to magically relocate to the best position, what is the gain possible compared to a static AP? Figures 7(a) plots the CDF of “rate gain” plotted from 8 clients across 21 different boxes in which the AP moved. We compute the rate gain as the ratio of max data rate from each box. Evidently, an Oracle can easily double the data rate on average, and up to 4x in $\approx 20\%$ cases. Figures 7(b) now plots the CDF of “SNR reduction” to reflect how the mobile AP can move to avoid interference from nearby interferers. SNR reduction is computed as the difference between median and minimum SNR (note that interference is a function of energy and not the interferer’s data rate, and hence plotted in terms of SNR). The achieved average SNR reduction is about $4.5dB$, contributing to a modest improvement in spatial reuse and throughput. In summary, the potential gains seem substantial given that the AP moved within a box of side 2 feet.

(2) Does Gain Scale to Multiple Clients?

In most realistic settings, the AP must serve multiple clients. So the natural question is: is there any AP location from which the data rates can be simultaneously improved for all clients? For this, we sum the data rates of all clients for each AP location within a given box – let $S_i$ denote this sum for location $i$. Then we compute the average per-client data rate gain, $\beta$, defined as $\frac{\text{median}_i(S_i)}{\text{max}_i(S_i)}$. As before, the median represents the performance to be expected when the AP is placed statically at a random location.

Figure 8 plots the CDF of $\beta$ for increasing number of clients. The gains are obviously expected to diminish since...
the AP must satisfy a stricter condition, nonetheless, the gains are still upwards of $1.35x$ on average even with 7 clients, and up to $1.45x$ for 3 clients. Homes mostly fall within this regime, where greater than 3 simultaneously backlogged connections are rare. In enterprises and hotspots (e.g., coffee shops), perhaps iMob can serve the 7 most data-hungry clients or the 7 weakest clients, improving the overall performance of the entire network. This result confirms the richness in indoor multipath diversity, offering support for robotic AP mobility even for multiple clients.

### (3) How much Gain in Throughput?

Figure 9(a) plots the CDF of throughput experienced by each client due to AP mobility. If an Oracle were to pick the best AP location, the throughput gain (compared to a random location) is shown in Figure 9(b). Aligned with expectations, the throughput gains are proportional to the data rate gains, although slightly less due to wastage from backoff and DIFS/SIFS slots.

### (4) Does the Gain Scale across Environments?

Figure 10 reports the Oracle’s median data rate gains from each of 4 environments, namely Office, Lab, Apartment, and Home. The reported gains are computed using the same metrics as above (i.e., max/median), and the experiments executed at 4 to 8 different places/rooms in each environment. The environment was entirely uncontrolled with natural human and object/furniture movements. Improvements are consistent, especially in larger Office where the clients are relatively further away from the AP (i.e., lower SNR). This is because modest improvement in SNR here can translate to greater rate improvement due to their logarithmic relationship.

To verify portability across hardware platforms, we performed similar measurements on USRPs and Intel cards. Figure 12 summarizes the results — this is loose in the sense that experiment conditions differed and some parameters were not identical (e.g., packet aggregation, MIMO, etc.) The key message is that the gains are consistent over static (single client), precluding any misgivings on our hardware.

### (5) How Many High Gain Locations?

The existence of high gain locations is a necessary but not sufficient condition — if such locations are rare, the AP would have to spend a large time searching for it, affecting performance. Now, instead of targeting only the maxDataRate locations, we define high gain locations as those that achieve greater than 0.95 times the maximum data rate in that box. Figure 11(a) plots the CDF of the fraction of these high gain locations, computed across 64 boxes from all experiments (we define “locations” as a $3x3$ cm² area as will be clear soon). Evidently, ≈ 40 high gain locations are available on average in a box, with some boxes offering far more. This is a favorable indication.

### (6) How Scattered are High Gain Locations?

It is important to also characterize the scattering of the high gain locations within the box — if all the high gain locations are clustered in a small region, searching one of them can still be time consuming. Figure 11(b) shows one example of the scattering in one box — the white marks denote high gain locations and visually illustrate that they are “well scattered”. However, to quantify this, we compute the distance, $\delta$, that an AP must travel to encounter a high gain location. Figure 11(c) plots the CDF of $\delta$ with randomly chosen starting positions, and with mobility similar to a 2D raster scan within the box. Evidently, $\delta$ is quite small for a large fraction of the cases, suggesting that high gain locations can be encountered without searching for too long. This brings hope that the potential gains might actually be achievable.

### 3.4 Understanding the Nature of Gains

While the upper bounds on performance are valuable, the extent to which the bounds can be achieved is also important. The next 4 questions are focussed on achievability.
Of course, the above graph also suggests that in some cases, the AP needs to move a large distance to encounter a high gain location. However, this does not mean that for these cases, the performance will be poor. To capture this, we attempt to answer the following question: if the AP moves a pre-specified distance δ, what is the best performance that can be achieved? Specifically, for increasing values of δ, we record the best data rate encountered, and compare this data rate against a static AP (i.e., median data rate in the box) and the Oracle (i.e., the max data rate in the box). Figure 13(a) and (b) plot the two comparisons, respectively – δ is defined as a fraction of a full raster scan in the box. Figure 13(a) suggests that even when the AP travels a small distance (δ = 5% of the raster scan), the data rate gain over static AP is still 1.5x. Figure 13(b) suggests that this gain reaches close to the Oracle. Thus, the overall message is that strong locations are not elusive – even if the best location is unavailable, “good” ones can be found quite quickly.

(7) How Predictable are High Gain Locations?

In designing a practical system, it would be useful if the existence of a nearby high gain location is predictable. Such predictions may be possible if the locations surrounding the high gain location form a gradient, like a “hill”. On the other hand, if the surrounding locations exhibit significantly less correlation to the high gain locations, then predictions are difficult. To this end, we compute the CSI at a given location and measure how the correlation degrades as we move gradually away from it. If the correlation degrades gradually, it would indicate the “hill” we desire. Figure 14 shows the results of this experiment. Unfortunately, we observe that CSI correlations are strong until separations of 2.5 cm, but plummets drastically at separations of 3 cm and more. This implies that the coherence region of a signal is around 3 cm, and locations outside that region is a poor indicator of its neighborhood. We term this 3x3 cm² coherence region as a pixel – which now defines a “location” – and recognize that neighboring pixels will vary drastically in SNR or data rate. Thus, the data rate landscape is like a “jagged mountain range” in the granularity of 3 cm, making predictions difficult. These results and conclusions are consistent with multipath theory and independent measurements [19]–[21].

(8) How Persistent are High Gain Locations?

If small changes in environmental factors cause the channel to change drastically, then iMob may not be worthwhile, since the AP will need to move very frequently. We classify environmental factors in 3 categories, namely human mobility, object mobility (e.g., doors, furniture), and device mobility (e.g., a smartphone moving in the user’s hand). We then extensively investigate temporal stability by perturbing each of these factors – a human user typing on the keyboard, many people walking around, furniture moving, client laptops moving, etc. In the interest of space, we distill our key findings: (1) Client device mobility at the centimeter scale induces drastic change in the CSI, causing the channel to heavily fluctuate. iMob may not be beneficial to such devices (tablets, smartphones) when they are being held/carried in the hand. (2) For a static device (e.g., laptop, TV), human and object mobility impact the channel only when they block dominant signal components between the AP and the client. However, as shown in Figure 15(a) and (b), the channel revives once the human/objects have moved past. (3) Only when the human or object moves to a new position, and also blocks the dominant signals, the CSI (and data rate) changes persist. However, such changes occur in the time scale of minutes [19] and can be detected by tracking changes in the CSI (detailed later). Thus, the take away
message is that iMob could be effective even under dynamic environments, so long as the clients are static.

Fig. 15. Data rate fluctuates when (a) humans and (b) objects move close to client, dwells for 10s, and walks past; the rate revives quickly.

4 System Design

We take away 3 important messages from the measurements above: (1) The achievable performance improvement due to robotic AP mobility is substantial, available under realistic conditions (multiple clients and different indoor environments), and hence worth pursuing. (2) The high gain locations are challenging to model because they are randomly located, spatially small, and often juxtaposed next to poor SNR locations (making predictions difficult). (3) Although challenging, some opportunities offer hope – the high SNR locations are many, well scattered in a box, and stable for reasonable time scales even in real environments. This section is aimed at designing a practical AP motion planning algorithm that will suitably cope/leverage the above challenges and opportunities.

Some Design Guidelines

The core task of the algorithm is to search through different pixels (called exploration) and stop at a pixel that is expected to offer maximum performance gains (called exploitation). In the interest of space, we omit various trials and deliberations that led to our final design; instead, we briefly discuss the key design guidelines that emerged from them. We will then assemble these guidelines into a practical iMob AP.

(1) Since AP mobility is at far slower time scales than packet transmissions, the exploration process must be speedy. Otherwise, an AP would spend unnecessary time at suboptimal pixels, widening the gap with the Oracle.

(2) Robotic motion is not accurate due to skidding of wheels, noisy compass values, mechanical turns – thus a robot cannot go back on the exact path on which it has travelled. This implies stopping decisions need to be made on-the-spot based on the SNR at that pixel. Performing a search and then retracing back to the max pixel on that path is not an option.

(3) The need to stop immediately at a high SNR pixel limits the maximum speed of the AP. Specifically, the inertial displacement after applying the brakes should be no more than a pixel width – this will allow the AP to stay within the same pixel once it decides to stop.

(4) Stochastic hill climbing or simulated annealing algorithms are not an option. Simulated annealing either incurs excessive time, or the starting point of the algorithm must jump to different random locations, which is impractical for the physically moving AP. Also, as mentioned earlier, these algorithms assume that backward motion is possible, which in our case is difficult.

(5) When clients move, or the environment changes too much, the CSI at the AP exhibits substantial change. This can be a trigger for the AP to re-explore the best pixel, since the current one may have become sub-optimal. This is particularly necessary when this client is data hungry and optimizing its performance will boost the overall network performance.

Finally, and perhaps needless to say, the mobility heuristic must be lightweight to run on a simple robot in real time.

Optimal Stopping Theory

The crux of our heuristic is designed around a result from optimal stopping theory (OST) in applied statistics [22], [23]. The problem definition of OST is as follows. An employer intends to hire 1 individual out of n applicants (all of whom can be ranked based on quality). The applicants are interviewed one by one in a random order. However, unlike typical situations, in this case the interviewer must make a decision immediately after the interview; once rejected, an applicant cannot be recalled. Of course, during the interview, the interviewer can rank all candidates seen thus far, but is unaware of the quality of yet unseen candidates. OST asks: which candidate should be selected to maximize the probability of recruiting the best candidate. Selecting too early can leave many good candidates unseen; picking too late might mean that the best candidate is already rejected. The OST result dictates that the first \( \frac{n}{e} \) candidates should be rejected, and among the subsequent candidates, the first on that ranks better than all \( \frac{n}{e} \) candidates should be recruited.

OST bears a strong resemblance to our problem of selecting the best pixel, primarily because the pixels are scattered in an entirely random manner, with little spatial correlation (3cm/s) (Figure 14). As a result, there is hardly a notion of “gradient” that can be leveraged. Moreover, channel modeling or ray tracing seemed impractical since the iMob AP does not have any details of the environment (floorplan, furniture, etc.) that would influence the multipath signal components. A statistical approach seems inevitable. In fact, given that high SNR pixels are not rare and quite well scattered (recall Figure 11(b) and (c)), a statistical approach may be able to find such a pixel within a short time. The time to search can be reduced by moving the AP fast during the exploration phase, and slowing it down during exploitation (i.e., when its time to stop). With this background, we now describe the heuristic precisely.

4.1 Mobility Planning Heuristic

Figure 16 shows the flow-chart for iMob’s mobility planning heuristic. The AP is placed at a random location by the user. Once it observes a stream of packets from a client, it begins an exploration phase. In this phase, it performs a raster scan at its maximum permissible speed, \( V_{\text{max}} \), recording the channel state information (CSI) from each packet transmitted by client(s). Of course, the AP continues to communicate during exploration, moving through pixels of varying quality. The exploration continues until the AP...
has moved through \( \frac{N}{2} \) pixels, where \( N \) is the total number of pixels in the box. At this point, the AP computes the best pixel among these \( \frac{N}{2} \) pixels, where “best” is defined as an utility function of CSI:

\[
U_{\text{max}} = \max_{p \in \{1, \ldots, \frac{N}{2}\}} \left( \frac{\sum l \log(SNR_i)}{I_p} \right)
\]

where \( p \) denotes a pixel covered by the AP, \( i \) denotes the index of its own clients. \( I_p \) denotes the number of interfering APs and clients sensed at \( p \). The AP now enters the exploitation phase.

![Fig. 16. Core flow diagram of iMob’s heuristic](image)

During exploitation, the AP computes every pixel’s utility, and stops whenever a pixel’s utility is \( \geq U_{\text{max}} \). However, to brake and stop in the same pixel, the velocity of the AP must be reduced during exploitation. Otherwise, inertia and skidding of wheels will propel the AP forward, and returning back to this exact pixel will be time consuming. The reduced speed, \( V_{\text{min}} \), is designed such that inertial displacement (after the application of brakes) is less than a pixel length (\( 3\text{m/s} \)) (discussed earlier). Once stopped, the AP continues communication with the client(s), expectedly at a near optimal data rate.

The AP remains in this location until a new data hungry client joins, or if it observes a substantial change in the CSI of a client. Substantial CSI changes suggest mobility of the client or appreciable changes in the environment. Under both these conditions, the AP triggers the exploration phase again, and relocates to a new pixel.

A common perception might be that the exploration phase incurring a performance penalty because the AP is moving during this time and communicating from sub-optimal pixels. We observe that this sub-optimality is true with respect to the Oracle but not with respect to the static AP. Note that a mobile AP should statistically achieve the same performance as a static AP during exploration because the mobile AP will move through equal number of strong and weak pixels. Evaluation results confirm this (as discussed later in Figure 17(c)).

A natural question might be: what if the channel quality at other locations improve over time – a iMob AP will not be able to proactively exploit this opportunity. We observe that this is unlikely when CSI is used as the indicator function. If some other pixel has to improve substantially, then either the client must move to a new location, or the environment must change appreciably. Unlike SNR, both the effects will manifest in CSI variations.

### Improvements to the Heuristic

We discuss a few optimizations to the core heuristic above.

1. In some cases, the exploitation phase may not end quickly – the AP may not encounter a pixel offering \( U_{\text{max}} \) for a long distance. In such cases, the AP could be made to lower its expectations in proportion to the time spent in the exploitation phase. In other words, the AP starts with the hope to achieve \( U_{\text{max}} \), but progressively lowers the bar to some fraction of this value. The rational is stop soon at a pixel that offers reasonable utility, as opposed to paying the cost for finding the perfect pixel.

2. Data hungry clients, such as those that perform video streaming, are likely to be the highest beneficiaries of iMob. However, most video streaming clients buffer data, leaving bursts of time in which packet downloads are much less. The AP could exploit these gaps to explore – if new pixels are discovered with greater utility, it could relocate. Recall that the pixel at which the AP stopped moving is not guaranteed to be optimal – its only a statistical estimate using OST. Exploring more can still be beneficial.

### 4.2 Multi-AP Coordinated Motion Planning

We extend the above heuristic to multiple APs (e.g., in residential neighborhood) by engaging the cloud as a mobility coordinator. The goal of the coordinator/controller, in both home and enterprise settings, is to position the APs in a manner that maximizes the utility metric \( \sum U_i \).

We extend the previously defined utility metric for multiple APs as a function of other APs and clients as follows:

\[
U_i(\{AP_i \mid \forall l \in [1, K]\}, \{Client_{im} \mid \forall m \in [1, C]\}) = \frac{\sum_j \log(SNR_{ij})}{N_i},
\]

where \( K \) is the number of APs and \( C \) is the number of clients in the system. \( SNR_{ij} \) is the SNR of client \( j \) which is associated to AP \( i \), and \( N_i \) is the number of interferers (both other APs and their clients) audible to AP \( i \). The log function encodes some level of fairness, so that an AP is not incentivized to position itself too close to any client. Of course, this utility does not capture the variation in traffic, rather assumes that all APs/clients are backlogged. Our goal is to characterize the gains even under these simplifications.

The optimal solution to this problem obviously requires a joint optimization on mobility and the utility function. The search space is large because an individual AP could move quite a bit to optimize for itself; moreover, all the APs could jointly move to mutually benefit each other. Since any AP movement will alter both the numerator and the denominator of the AP’s utility function, the possible combinations quickly explode. We first prove that the problem is NP Hard and then present a simple heuristic.

### Proof of Hardness

In this section, we prove NP hardness on a simpler version of the problem. Consider \( K \) APs and \( n \) clients. APs have radii \( r \) such that an AP is said to cover a client if they are located within a distance of \( r \) from each other. Denote \( U_i = \{ C_j \mid C_j \text{ is covered by } AP_i \} \), where \( C_j \) represents clients \( \forall j \in \{1...n\} \). The objective function is \( \sum_{i=1}^{K} U_i \) i.e., the
cardinality of union of set of clients covered by the APs. The problem is to maximize this objective function.

**Algorithm 1 Reduction**

1. Input: \( \{S_1, S_2, S_3...S_m\} \): \( K \) APs
2. for all \( C_i \in C \) do
   3. draw a circle \( C_i \) on a plane such that it does not intersect with any other circle
4. end for
5. for all \( S_i \in \{S_1, S_2, S_3...S_m\} \) do
   6. Move circles representing elements in \( S_i \) so that a new region \( R_i \) is created, such that:
   7. \( \text{Overlap}(R_i) = S_i \)
8. end for

**PROOF.** The bottleneck steps in Algorithm 1 are 5,6 and 7. Steps 6,7 have a maximum complexity of \( O(n^2) \) where \( n \) is the cardinality of union of \( \{S_1, S_2, S_3...S_m\} \). The for loop in step 5 has \( m \) iterations. Therefore the complexity is \( O(mn) = O(n^2) \).

**LEMMA 2.** An optimal solution of the AP placement problem obtained from the reduction in Algorithm 1 generates a set \( O \) of client-subsets covered by APs. Then \( O \subset \{S_1, S_2, S_3...S_m\} \) and \( O \) is a valid solution to the maximum coverage problem.

**PROOF.** Algorithm 1 has a side effect. In step 6, it produces regions which are subsets of \( S_1 \). In other words, the reduction will not only produce \( S_i \) but also produces subsets of \( S_i \) where an AP can be potentially placed. Lemma 2 can be violated if a subset \( \text{Sub} \) of \( S_i \) is included in \( O \) such that \( \text{Sub} \notin \{S_1, S_2, S_3...S_m\} \). We will prove that replacing \( \text{Sub} \) by \( S_i \) will create another optimal solution \( O \) such that \( O \subset \{S_1, S_2, S_3...S_m\} \).

\[ O' = (O - \text{Sub}) \cup S_i \]

Given that \( |\text{Sub}| < |S_i| \), we have \( O' \geq |O| \).

Hence \( O' \) is a solution that satisfies lemma 2.
LEMMA 3. An optimal solution to the maximum coverage problem is O. Then O is a valid assignment of clients to APs in the AP placement problem.

PROOF. Follows directly from the reduction (Algorithm 1). Every subset picked in the maximum coverage problem, corresponds to a valid region such that the clients corresponding to the subset are served by placing an AP there. K (or few) such subsets correspond to K (or few) valid AP placements. If fewer than K subsets turn out to be optimal, the remaining APs can be ignored.

LEMMA 4. An optimal solution of the AP placement problem obtained from the reduction in Algorithm 1 generates a set O of clients covered by APs. Then O \subset \{S_1, S_2, S_3 ... S_m\} and O is an optimal solution to the maximum coverage problem.

PROOF. Suppose the optimal solution O to the AP placement problem is sub-optimal to the maximum coverage problem (from lemma 2, O is a valid solution to the maximum coverage problem). Suppose O’ is the optimal solution to the maximum coverage problem. Then |O’| > |O|. It follows from lemma 3 that O is a valid AP-client assignment in the AP placement problem. However |O’| > |O|. This is a contradiction because O was the optimal solution to the AP placement problem. Hence lemma 4 must be true.

With lemma 4, we prove that the simpler version of the AP placement problem is as hard as the maximum coverage problem. We now present a simple heuristic for the broader version of the AP placement problem.

Heuristic Design

Algorithm 2 describes our heuristic for the enterprise (a small modification makes it applicable to homes). While the focus of the paper is to demonstrate the promise of feet scale micro-mobility, the heuristic covers a broader class of macro-mobility where the AP can move across multiple spots, covering several meters. In future smart homes, perhaps tracks installed on false ceilings can help realize macro-mobility. Yet, we restrict real testbed experiments to micromobility but evaluate both macromobility (for enterprises) and micromobility (for homes) under simulations (detailed later). Assuming K APs available, the core intuition is that clients can first be clustered into K groups (K means clustering), and each AP assigned to a cluster. Assuming client locations are roughly known, the APs can be initially placed at the center of mass (CoM) of their respective clusters. For ease of explanation, let us number the APs from 1 to K. The first AP is moved within a radius r such that \(U_1(\text{Client}_m | \forall m \in [1,C])\) is optimized (i.e., ignoring other APs); the second AP is moved within a radius r from its CoM, such that it relocates to a location that maximizes \(U_2(AP_1, \text{Client}_m)\). With AP1 and AP2 fixed in their positions, AP3 is now moved within radius r from its CoM, such that it relocates to a location that maximizes \(U_3(AP_1, AP_2, \text{Client}_m)\). This continues for all APs, and at the end of this pass, the controller computes \(\sum_{i\in[1,K]} U_i\). The controller executes multiple passes of the same operation but placing the APs in a different order each time. The maximum value of \(\sum_{i\in[1,K]} U_i\) from all these passes is selected, and the corresponding AP configuration prescribed. APs move to the prescribed locations on the grid-tracks, and performs micro-mobility (explained later).

Algorithm 2 Coordinated AP mobility heuristic

1: Input: P: Set of K APs
2: Create cluster of clients \(C_i \forall i \in [1,K]\)
3: Assign cluster \(C_i\) to \(A_i \forall i \in [1,K]\)
4: for \(p \leftarrow 1\) to MAX_PASS do
5: for all \(A_i, i \leftarrow \text{Random Ordering}(1\ to\ K)\) do
6: Place \(A_i\) @ location that maximizes \(U_{ip}\{AP|\forall l \in [1,i-1]\}, \{\text{Client}_m|\forall m \in [1,C]\}\)
7: Update \(U_{ip}\) and send to CONTROLLER
8: end for
9: end for
10: CONTROLLER: Select AP positions to maximize \(\sum_{i\in[1,K]} p[e\in[1,MAX_PASS]} U_{ip}\)

The heuristic for Home is identical, except for two factors. (1) APs are not initialized at the center of mass (CoMs), but at their installed locations (near the wall). (2) They only perform micro-mobility step using the same utility function, which is outlined next.

In the next phase, the controller co-ordinates micro/miNI-movement of APs to further optimize the same utility metric by taking advantage of multipath diversity. The APs physically move and explore/exploit a spot (2x2 feet area) around their controller assigned positions similar to the single AP placements (similar to Section 4.1). Each AP performs this step, one after the other in a sequential order, by considering only the previous APs who have settled down. The APs that are yet to move are ignored in exactly the same fashion of controlled assigned positions in Algorithm 2. The APs physically measure the utility metric and settle at their respective local optimum positions.

5 Evaluation

We evaluate a completely functional single and multi-AP iMob system and focus on (1) the throughput and fairness comparison with today’s static APs, (2) gap from the Oracle, and (3) impact of various parameters, such as client density, traffic sessions, mobility area, etc. We begin with a brief description of our experiment methodology.

5.1 Implementation and Methodology

The evaluation platform is similar to the measurement platform, with few key differences. The iMob exploration/exploitation heuristic has been implemented in the Linux kernel to completely operate in real time (e.g., pixel search, utility computation, Roomba speed control, braking). Performance is measured on the wireless link only – the wired Internet connections at residences are the bottleneck, so connecting to the Internet would not reflect the actual wireless gains. We perform both single AP and multi-AP experiments. In the multi-AP case, a central server controls 4 APs – deployed across 2 floors of our university building – to extract holistic SINR and topological gains. Clients associate to our AP and upload/download packets over UDP/TCP while the AP moves to optimize performance. To compare against the Oracle, we performed experiments
with continuous mobility and used the CSI data to precisely infer data rates [18] and throughput of each scheme. For realistic backlogged traffic, we record and use packet traces from YouTube, Google Hangout, and casual browsing sessions, captured from Wireshark. Across all experiments, the AP and clients were placed at realistic locations (to the extent possible). The environment was completely uncontrolled with people naturally moving, working, etc.

As a final point, Figure 20 plots the inertial displacement of our Roomba robot from the time of braking, for increasing AP speeds. Given pixels width of 3 cm/s, the maximum AP velocity prescribed by this graph should be less than 20 cm/s – we conservatively use 5 cm/s since the braking may happen half-way into the pixel.

_fig20_ Roomba’s inertial displacement after braking.

5.2 Real-time Single AP Experiments

Figure 17(a) plots the throughput comparison between iMob and a Static AP for various sessions, using 4 static and fully backlogged clients. Average throughput improvement is 44%. One of the cases shows Static performing slightly better, perhaps because it was fortunately located at a strong SNR pixel. This is statistically a rare event, but possible.

Figure 17(b) compares the throughput achieved during the time the iMob AP was moving – this confirms that AP mobility does not impose a performance penalty. The throughput achieved by Static and Mobile are comparable since, statistically, the Mobile AP moves through both strong and weak quality pixels. However, once the AP stops at a strong SNR pixel, the performance exceeds Static thereafter, translating to net gain. Figure 17(c) zooms into the data rates observed during the exploration and the exploitation phase, showing how iMob’s performance improves after stopping. Note that even while stationary, an AP (both Static and Mobile) still experience rate variations by around a notch due to temporal fluctuations (as seen in - Figure 3).

5.2.1 Coping with Environmental Dynamism

Observe that environmental dynamism will alter the optimal AP position, hence the iMob AP will need to trigger a new exploration phase. iMob uses a CSI based classification method that correlates the newly observed CSIs with recent CSIs, using techniques similar to [19]. If the correlation drops greater than a threshold, the AP triggers a relocation. For this, one of the clients was mounted on a Roomba and programmed to move periodically in our experiments – Figure 21(a) plots example timings of the client mobility and the Mobile AP’s relocation trigger. The detection accuracy is robust and not affected by other humans moving in the environment. Figure 21(b) plots the detection accuracy across all experiment sessions, as a function of the distance the client moved from its prior position. In some additional cases, the AP also triggered mobility because of CSI changed (even though the client did not move), but we are unable to verify if it was a valid trigger. This is because we do not know the ground truth on whether the environment truly changed or not, hence false positives cannot be computed in such cases. To shed more light, Figure 21(c) shows the CDF of throughput variation between two cases: (1) a human is typing and working with the client laptop, and (2) the client laptop without the human user. The similarity in deviation suggests that the channel does not vary due to the human working, obviating the need for iMob APs to move in such realistic cases.

5.2.2 Fairness and Leash Length

Figure 22(a) shows that throughput improvements with iMob is not obtained at the cost of fairness. Using Jain’s Fairness Index, we find comparable performance as Static. Moreover, if desired, iMob can explicitly optimize for fairness, or even a combination of throughput and fairness.

Figure 22(b) plots the variation of throughput with decreasing coverage area of the mobile AP. The performance does not degrade too much, indicating that the diversity is truly rich. This bodes well for iMob – even where the AP has less than a feet to move around, the single AP throughput gains can still be 40%.

5.2.3 Comparison with Oracle

Figure 23 compares iMob’s performance against Oracle and Static AP, for single client scenarios. The experiment sessions are derived from wireshark traces of YouTube, Hangout, and a casual browsing session. For example, for YouTube, active time windows were concatenated, while intermediate gaps (typical for buffered playback) were not considered. Evident from the graphs, increasing session lengths improve throughput because the sub-optimality during the exploration phase gets amortized over longer session lengths, and the performance at the best pixel begins to play a more dominant role. Figure 23(a) shows that iMob remains reasonably close to the optimal, around 0.9. Against Static AP, iMob continues to achieve around 40% gain on average, but exceeds 80% or more in some cases with longer traffic sessions.

Figure 24 shows the variation of iMob’s throughput against the Oracle and Static for increasing number of clients. iMob outperforms Static consistently and stays close to the upper bound. This suggests the efficacy of the optimal stopping heuristic to find a high quality pixel, even within 2 feet mobility.
Figure 21. (a) AP detects when client moves and trigger relocation. (b) Detection accuracy for increasing client displacement. (b) Variation of data rates when human typing on a laptop versus the absence of humans.

Figure 25. (a) iMob testbed deployed in 2nd and 4th floors. (b) Downlink throughput comparison. (c) Uplink throughput comparison. (d) Gain due to spatial reuse only, caused by sidestepping mutual interference from the other APs.

that the neighboring APs are at the edge of each other’s interference range. Transmit powers were assigned at 8dBm to all the nodes; clients remain static for all the sessions. The topology mimics an EWLAN network of access points where the APs in the same channel are placed far from each other. A central server connects to each AP over WiFi and coordinates their movements to configure an effective topology that offers strong SNR to the AP’s clients but avoids interference (to the extent possible) from other APs.

Figures 25(b) and (c) report the downlink and uplink UDP throughput comparison between the Mobile and Static AP. Gains are higher – 65% for downlink and 90% for uplink on average – implying that interference avoidance and better client SNR together contribute to net benefits. Fairness remains greater than the static case (not shown here). Figures 25(d) zooms into this break-up and shows the improvements due to spatial reuse. The “Gain %” on the Y axis shows how much extra opportunity was created by evading interferers in comparison with the static AP case. The average gain was about 12%, considerably less than client throughput gains. This is because of the binary nature of the carrier sensing threshold (APs need to find positions where the interferer is outside the sensing range). Nevertheless, the gains are still worthwhile because it combines multiplicatively with data rate gains resulting in net amplification in throughput.

6 Large Scale Simulations

We conduct NS3 simulations using measured real channel data for scalability testing. For enterprises (EWLAN), the setup is modeled after the floorplan of our 54 x 36 square-meter lab 12 rooms, with around 1 AP for every two rooms, and 4 active clients for each AP. For residences (Home), we consider 6 neighboring houses, each with 1 AP and 4 active clients. As a comparison baseline, APs are placed arbitrarily near the walls.
Performance Results

Data Rate

Figure 26(a) plots CDF of data rate improvement in the Enterprise and Home, against the respective baseline of static APs. The median improvement in Enterprises is around 10 Mbps, demonstrating the value of optimizing client SNR alone (note that avoiding interference improves spatial reuse, but not data rate). Median improvement in Home is around 7 Mbps, lower than enterprises due to the shorter “leash” in mini-mobility.

Figure 26(b) compares the Enterprise heuristic with “Extensive”. The latter is essentially the same heuristic, except that it is not restricted to move within \( r \) radius from the CoMs; the APs can relocate to any location in the building. Evidently, searching extensively offers appreciable data rate improvements, albeit some increase in the search time. Since macro-mobility can be entirely simulated in the cloud with coarse pathloss models, perhaps “Extensive” can indeed be achieved in practice.

![Fig. 26. Data rate gains in enterprise and homes.](image)

Throughput and Fairness

Figure 27 plots the median throughput improvement for UDP, TCP, uplink, and downlink traffic. These are averaged across 50 random client topologies. Collisions cause TCP to backoff aggressively, resulting in lower gain than UDP.

![Fig. 27. TCP and UDP throughput gains.](image)

Energy

The increase in SNR and data rates reduce the air time of packets, both for a given AP-client link, as well as for (overheard) interfering transmissions. Avoiding interference further reduces overhearing, all together contributing to substantial energy reduction. Figure 30 shows the percentage energy savings over the static baseline, normalized by the number of packets – 30 to 40% gains are feasible. Energy specific optimizations can offer additional gains – the APs could move to ensure that some smartphone’s remaining battery life is optimized.

![Fig. 30. Energy savings in Enterprise and Home.](image)

Tradeoff between Mobility and Performance

We explore the variation of performance gain with “moved distance”. Figure 31(a) shows rise in throughput in the Enterprise when the AP is allowed a longer “leash” (expressed in terms of spots scanned). Evidently, moderate amount of mobility can offer most of the gains – searching 20 spots attain almost 65% of Extensive. Home trends are similar, except that the gains saturate quicker due to the shorter “leash”. We measured the CDF of overall AP mobility and find that the median is 4m (and max 8m), indicating that moderate mobility brings most of the gains.

7 Limitations and Opportunities

This is an early attempt to characterize and exploit the landscape of robotic wireless networks. Much remains to be done.
MIMO interface – we believe that feet-scale mobility can be an alternative to these. Our results show that moving within a 2 feet box can yield higher data rates even with a 3x3 MIMO interface – we believe that feet-scale mobility can offer higher ranked channel matrices. From the robotics side, authors in [30], [31] have researched how robots cooperate to achieve a common wireless communication goal. In one instance, robots plan their motion paths to constructively beamform towards a specified receiver. Authors in [32] have envisioned robots forming a “chain route” to maintain connectivity to first responders (e.g., fire fighters) moving into a catastrophe stricken building. Delay tolerant networks have also considered node mobility [33], even in under water [34] and mobile sensor networks [35]. We believe this paper is still different in the sense that it brings feet-scale controlled mobility to existing network infrastructure that are conventionally viewed as static.

In [16], authors envisioned robotic wireless networks and presented initial upper bounds on USRPs. This paper builds significantly on top of [16], including (1) full scale testbed with Intel 5300 802.11n 3x3 MIMO turned on, (2) insights into the nature of mobility gains, (3) a practical heuristic using optimal stopping theory, (4) a real time system running with single and multiple coordinated APs constructed out of Roomba robots.

9 Conclusion

This paper envisions WiFi APs-on-wheels that move in controlled ways to minimize APs power control and channel allocation can now be performed jointly with mobility, and adapted to changing traffic conditions.

Localization and Security: Micro-moving APs may be able to mitigate the impacts of multipath, converging to a reasonably accurate pathloss index for their observed channel. Moreover, they could move macro distances to “look” at clients from different vantage points, ultimately improving the various techniques in triangulation and trilateration. Security benefits can also emerge from constantly moving the device, thereby changing the channel properties that are used as the “secret key” between the transmitter and receiver.

8 Related Work

The work closest to this proposal is probably MoMiMo [6], where the receiver adjusts its antenna in centimeter scales to perform interference alignment. While MoMiMo is a specific optimization for interference, this paper attempts to create a broader theme of robotic wireless networks, and presents a case for the regime of feet scale full-device mobility. Perhaps a further step in this direction is “software defined mobility” where the cloud controls the mobility of network infrastructure. Finally, MoMiMo is complimentary to iMob – a future WiFi AP can implement both. Google’s project Loon [24] provides Internet access to remote areas via ad hoc network-style balloons drifting above the stratosphere. DARPA envisioned the use of self-autonomous network of LANDroid robots [25] to provide connectivity in urban warfare areas. Our broad proposal certainly bears similarities, but concentrates on injecting controlled mobility to today’s established infrastructure.

Spatial diversity has been exploited in MIMO, beamforming [26], [27], and through other opportunistic ideas [28], [29]. Infrastructure mobility is by no means an alternative to these. Our results show that moving within a 2 feet box can yield higher data rates even with a 3x3 MIMO interface – we believe that feet-scale mobility can offer higher ranked channel matrices. From the robotics side, authors in [30], [31] have researched how robots cooperate to achieve a common wireless communication goal. In one instance, robots plan their motion paths to constructively beamform towards a specified receiver. Authors in [32] have envisioned robots forming a “chain route” to maintain connectivity to first responders (e.g., fire fighters) moving into a catastrophe stricken building. Delay tolerant networks have also considered node mobility [33], even in under water [34] and mobile sensor networks [35]. We believe this paper is still different in the sense that it brings feet-scale controlled mobility to existing network infrastructure that are conventionally viewed as static.

In [16], authors envisioned robotic wireless networks and presented initial upper bounds on USRPs. This paper builds significantly on top of [16], including (1) full scale testbed with Intel 5300 802.11n 3x3 MIMO turned on, (2) insights into the nature of mobility gains, (3) a practical heuristic using optimal stopping theory, (4) a real time system running with single and multiple coordinated APs constructed out of Roomba robots.

9 Conclusion

This paper envisions WiFi APs-on-wheels that move in controlled ways to optimize desired performance metrics. Early results are promising, although a deeper treatment is needed to fully characterize the interplay of many parameters underlying the success of such technology. Nonetheless, mobility is a valuable degree of freedom missing in today’s network infrastructure, and extending research attention to it, we believe, is entirely worthwhile.

References