DIRC: Increasing Indoor Wireless Capacity Using Directional Antennas

Xi Liu[†], Anmol Sheth[‡], Michael Kaminsky[§], Konstantina Papagiannaki[§],
Srinivasan Seshan[†], Peter Steenkiste[†]

[†]Carnegie Mellon University, [‡]Intel Research Seattle, [§]Intel Research Pittsburgh

ABSTRACT

The demand for wireless bandwidth in indoor environments such as homes and offices continues to increase rapidly. Although wireless technologies such as MIMO can reach link throughputs of 100s of Mbps (802.11n) for a single link, the question of how we can deliver high throughput to a large number of densely-packed devices remains an open problem. Directional antennas have been shown to be an effective way to increase spatial reuse, but past work has focused largely on outdoor environments where the interactions between wireless links can usually be ignored. This assumption is not acceptable in dense indoor wireless networks since indoor deployments need to deal with rich scattering and multipath effects. In this paper we introduce DIRC, a wireless network design whose access points use phased array antennas to achieve high throughput in dense, indoor environments. The core of DIRC is an algorithm that increases spatial reuse and maximizes overall network capacity by optimizing the orientations of a network of directional antennas. We implemented DIRC and evaluated it on a nine node network in an enterprise setting. Our results show that DIRC improves overall network capacity in indoor environments, while being flexible enough to adapt to node mobility and changing traffic workloads.

Categories and Subject Descriptors

C.2.2 [**Computer System Organization**]: Computer-Communication Networks

General Terms

Algorithm, Design, Performance

Keywords

Directional Antenna, Indoor Wireless Capacity

1. INTRODUCTION

The demand for wireless bandwidth in indoor environments continues to increase with the rapid integration of Wi-Fi

radios in every day consumer electronic devices such as laptops, cell phones, cameras, and audio/video equipment. This increased density of Wi-Fi transmitters has exacerbated contention for the wireless medium and reduced overall throughput. For example, in [17], the authors observed that more than 70% of the frames in a conference had at least one contender. One effective approach to improving throughput is to reduce interference among radios, thus allowing them to transmit concurrently.

Directional antennas have the potential to provide the necessary interference reduction by spatially confining transmissions. For example, commercial, off-the-shelf directional antennas can provide spatial isolation of up to 20 dB by confining the signal within a sector of 10°, i.e., the signal outside the sector is at least 20 dB weaker than the signal within the sector. This high degree of spatial isolation can support concurrent transmissions between pairs of radios even when they are located close to one another—provided that the antennas can be oriented correctly.

Traditionally, directional antennas have mostly been used in outdoor environments where there is a direct line-of-sight (LOS) between the two endpoints [19, 21]. In such applications, orienting directional antennas is relatively simple because the LOS antenna orientation usually maximizes the signal strength at the receiver and is the *only* configuration that works well. In an indoor environment, the presence of rich scattering and multipath effects results in non-LOS antenna orientations that provide comparable signal strength at the receiver. In fact, *conventional wisdom has been that directional antennas will not be effective indoors* because of this issue. Although the existence of these alternate good orientations complicates the configuration of the system, it also creates an opportunity for reducing interference between transmissions. For example, while the LOS orientation of a transmitter may cause interference at some location (e.g., S1's signal interfering with R2's reception in Figure 1(a)), an alternate configuration that uses a reflected path may steer the signal such that it reaches the intended recipient without causing interference (e.g., Figure 1(b)).

However, to achieve the gains offered by these alternate configurations requires significant, explicit coordination between the transmitters in the network. For example, coordination is necessary for transmitters to measure the interference created in different antenna configurations. In addition, since the presence of directional antennas exacerbates the presence of hidden and exposed terminals in the network [23], coordination is also needed to address MAC related problems.

In this paper, we present DIRC, a wireless network design that improves spatial reuse in indoor environments using directional antennas. We focus our exploration on a typi-

(a) Using max SNR/LOS directions

(b) Exploiting reflected paths
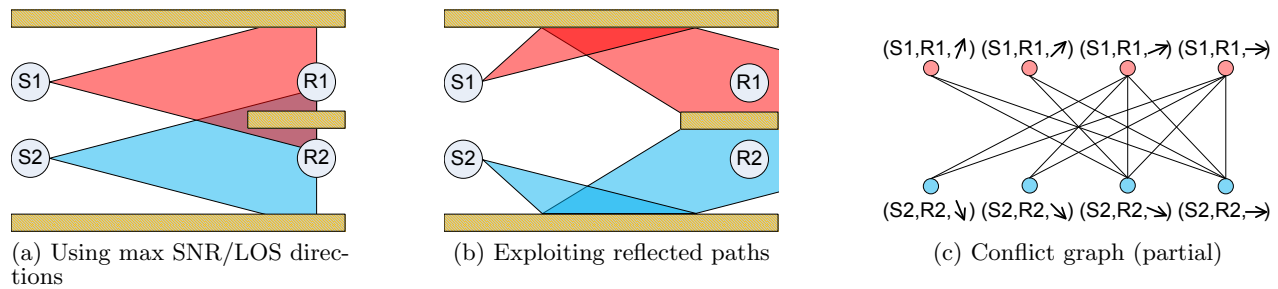
(c) Conflict graph (partial)

Figure 1: Example of exploiting multiple paths using directional antennas and conflict graph

cal enterprise environment in which: 1) the access points (APs) are centrally controlled and managed, 2) the APs are equipped with phased array (software steerable, directional) antennas, and 3) the clients have standard 802.11 hardware. The core of our solution is an algorithm that identifies close-to-optimal orientations for the directional antennas, maximizing system-wide capacity while ensuring that configuration overhead is low and scales linearly with the number of APs. DIRC also incorporates a new TDMA-based MAC protocol designed for indoor directional antennas. This design is much more efficient than CSMA/CA-based MAC protocols, which assume that the interference at the sender is similar to the interference at the receiver. This assumption breaks down dramatically with directional antennas.

This paper makes the following contributions. We design and implement DIRC, the first system that improves indoor wireless capacity through the use of directional antennas. DIRC's centralized algorithm can achieve close-to-optimal transmission scheduling and antenna orientations with very low computational overhead, as demonstrated on a testbed with 3 directional APs and 6 clients. We show that the end-to-end DIRC implementation works well in practice. In a testbed network, UDP performance improves by 65% over prior approaches and 100% over using only omni-directional antennas. Our experiments demonstrate that DIRC can handle node mobility and dynamic traffic patterns.

The rest of the paper is organized as follows. We first elaborate on the challenges associated with the indoor use of directional antennas in Section 2. We then present our solution, DIRC, in Sections 3 and 4. Section 5 evaluates DIRC in two real world deployments. In Section 6 we discuss related work and conclude in Section 7.

2. CHALLENGES

In this section, we first present the RF technology background that is necessary to understand the challenges and our proposed solutions. We then detail the two main challenges in using directional antennas to improve indoor RF spatial reuse: 1) determining effective antenna orientations and 2) determining which directional nodes should transmit concurrently. Note that these two challenges cannot be addressed independently. For example, the choice of antenna orientation may depend on the set of nodes that the MAC protocol determines should transmit concurrently. We consider such interactions below.

2.1 Background

The focus of our work is on the use of directional antennas to increase spatial reuse. Unlike omni-directional antennas that

have a uniform gain in each direction, directional antennas have a different antenna gain in each direction. As a result the signal level at a receiver can be increased or decreased simply by rotating the orientation of the directional antenna. In the directional mode, the antennas we use have the ability to increase or decrease the signal strength at a receiver node by up to 20 dB. Received signal strength indoors can be further affected by the presence of strong RF reflectors like metal cabinets, walls, and doors, resulting in multiple reflected paths that add up constructively or destructively at a receiver.

In addition to directionality, our design also relies on the property of many Wi-Fi radios, called the capture effect. If a card supports capture, it can successfully receive a packet transmission in the presence of concurrent transmissions, as long as the desired transmission has a sufficiently higher signal strength. The signal strength difference required is usually in the range of 20-25 dB for 54 Mbps. The majority of this difference, 10-20 dB, can be provided by appropriately orienting the antennas. The rest of the difference can be achieved from the fact that many receivers are closer to their senders than to the interferers. As a result, with careful configuration of directional antennas, we can often ensure that the intended receivers can "capture" their packets, despite the presence of interfering transmissions.

2.2 Antenna Orientation

Simple Heuristics Do Not Work Well. Directional antennas have primarily been used in outdoor deployments, where the LOS orientation of the antenna towards the receiver provides both the best performance and the best signal strength [19, 21]. Indoors, the LOS path may not exist because of obstructions between sender and receiver. As a result, existing indoor directional deployments have tried using the direction of maximum signal strength to determine the antenna orientation. We call this approach the *Max SNR* approach. However, this approach only works well *in isolation*. If multiple directional senders exist in an indoor space, and can potentially transmit simultaneously, then orienting the senders according to the max SNR direction will not necessarily lead to the maximum spatial reuse, or system-wide *capacity*.

Figures 1(a) & (b) illustrate why the Max SNR approach may not always maximize spatial reuse. Nodes $S1$ and $S2$ are two directional senders that wish to transmit data to omni-directional receivers $R1$ and $R2$ respectively. Given that there are no obstructions between senders and receivers, the max SNR direction is the same as the LOS direction (Figure 1(a)). Unfortunately, the LOS/max SNR directions lead to high
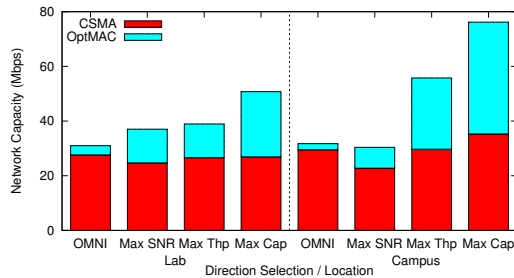
Figure 2: Median network capacity under different antenna orientation algorithms and MAC protocols

CSMA MAC protocol (red/lower bars) and using an optimal MAC protocol that is capable of coordinating transmissions across space while avoiding hidden and exposed terminals (OptMAC - red plus blue/lower plus upper bars).

Figure 2 shows that carefully configured directional antennas (Max Cap) with OptMAC can provide about twice the performance of using omni-directional antennas (also with OptMAC). Note that the maximum possible gain with three transmitters would be a factor of three. Unfortunately, simple local heuristics (Max SNR and Max Thp) are unable to identify configurations that perform close to optimal, even with OptMAC. Although it may prove difficult to find this optimal configuration, this result shows that there can be significant performance gains. The graph also highlights the importance of an effective MAC protocol; the CSMA protocol, for example, does a poor job using the capabilities of the directional antennas. Our work and contributions address both of these aspects: an optimal antenna orientation algorithm and a MAC protocol design. We will show later that the performance of our solution is very close to that of the Max Cap with OptMAC.

Enormous Search Space. The greatest drawback of the Max Cap approach is that it requires exploration of all possible orientations of every sender.[1] The size of this search space grows exponentially when potential interferers are also directional. Assuming that all n directional senders in the network can choose any of k directions, the search space to identify the optimal orientation for each one of the senders to their respective receivers is k^n. As directional antenna technology improves, beam widths are likely to become smaller [1], thus increasing k. This will render a brute force approach even more impractical.

Dynamicity. The problem of antenna orientation becomes even harder when one considers RF environment variability and user mobility. Any algorithm for optimal antenna orientation across directional senders will need to be able to collect, process and act on information very quickly. Current electronically steerable antennas can change their orientation on a per-packet basis. The challenge is to design an algorithm that provides the best tradeoff between optimality and stability/overhead.

2.3 MAC for Indoor Directional Transmissions

Choosing the correct MAC protocol is crucial to realizing the performance benefits described above. In this section, we present requirements of an indoor directional MAC protocol and discuss several candidates.

First, the most important task of any MAC protocol is to identify the set of non-interfering transmissions in an area and to coordinate the activities of the various senders. As we discussed above, the notion of non-interfering transmissions depends on the antenna orientations of the senders. Thus, an indoor directional MAC protocol must not only identify the set of possible concurrent transmissions but also determine their orientations.

Second, not all transmitters in the environment will be directional. For example, in Figure 1(a), the two senders are directional and the two receivers are omni-directional. If directional transmission $S1 \rightarrow R1$ and omni-directional trans-

interference at the receivers. For example, the transmission of $S2$ to $R2$ interferes with reception at $R1$, since $R1$ is within the transmission range of $S2$. In this configuration, the MAC protocol must ensure that the two senders never transmit at the same time. In contrast, if the two senders select the orientations shown in Figure 1(b), then both senders could transmit simultaneously. Interference would still exist at the receivers, but it would be weaker, leading to a higher SINR at $R1$ and $R2$, and potentially successful packet receptions. This example illustrates that the best antenna orientation is not just a function of the *receiver's location*, but also a function of the *location of interfering transmitters*.

This example also shows how the benefits of directional antennas differ from those of transmit power control (i.e., when the senders $S1$ and $S2$ reduce their transmission power levels to try and avoid interfering with one another). When senders reduce their transmission power, they do not only reduce the interference at the unintended receiver, but they also reduce (by the same amount) the signal strength at the intended receiver. Using directional antennas and reflected paths may make it possible for the transmissions to leverage the environment (e.g., the obstacle in this example) to reduce the interference level significantly while only slightly reducing the signal level.

Figure 2 illustrates the dramatic performance impact that correctly orienting directional antennas can have. These experimental results are for two testbeds, each using three directional transmitters. The testbed and experimental setup are described in Section 5 (this figure only shows median capacity, and mean capacity is similar). The OMNI bar in the graphs shows the performance observed when the APs transmit omni-directionally. Max SNR is the simple heuristic where each sender chooses the direction that has the maximum SNR, as observed at the receiver. An alternative heuristic is denoted as *Max Thp*, where each sender chooses the direction that maximizes its individual link throughput. Note that this second heuristic relies on an iterative process by which each sender picks the direction that maximizes its own throughput *given* the directions selected by the other directional senders. The final configuration of Max Thp is either one where no sender can further improve its throughput or the one reached after 30 algorithm iterations. Finally, the *Max Cap* bar in the graph shows the performance obtained by orienting the antennas in directions that maximize system-wide capacity. We determine this configuration through exhaustive search to illustrate the maximum capacity that can be achieved in each scenario. For each of these antenna orientation schemes, we operate the system using the 802.11

---

[1]Note that this complexity would be even higher in the presence of directional receivers. Throughout the paper we assume that receivers are omni-directional, since such devices are likely to be laptops and handhelds.
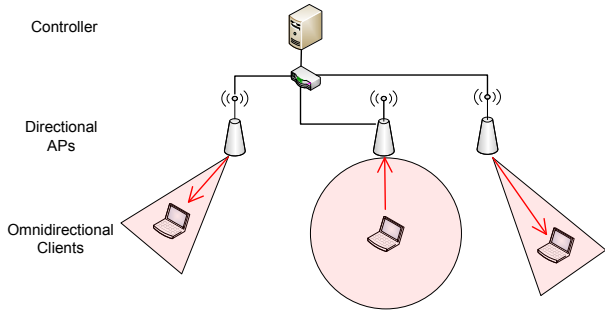
Figure 3: Network diagram of the system

mission $R2 \rightarrow S2$ happen simultaneously, the interference at $R1$ from $R2$ may be too strong, no matter what direction is used by $S1$. Thus, any MAC protocol must coordinate both the directional and omni-directional nodes in an area.

One possible choice for a MAC protocol is to use CSMA like in 802.11. However, as earlier research and our results above point out, while CSMA works well in networks with omni-directional transmitters, it has several problems in networks with directional transmissions and performs poorly.

Past research has proposed a wide range of MAC protocols for directional wireless networks [15, 10, 16, 26, 9, 20, 28, 5]. Much of this work uses RTS/CTS, and directional virtual carrier sensing (DVCS) or directional network allocation vector (DNAV). The basic idea of these solutions is that a direction will be reserved if RTS or CTS is received from that direction, and that direction is marked as unusable during the DVCS/DNAV. One way in which these designs fail to meet our needs is that they assume that the antenna orientation for any sender-receiver pair is fixed (it is a function only of the receiver). This is a reasonable assumption for outdoor settings, where there is a single reasonable orientation (the LOS direction) for any transmission.

Other research [19, 21] has proposed the use of TDMA-based MAC protocols in directional wireless networks. However, these past designs have assumed that there is either no or only controlled interference between directional transmissions in determining the TDMA schedule. Our tests show that this assumption is too aggressive in indoor settings. Even with narrow beamwidth (8°) antennas, our experience is that transmissions are not completely isolated from each other. Another weakness of past MAC designs is that they largely ignore packet capture, i.e., CSMA, DNAV and DVCA approaches tend to be too conservative in scheduling transmissions. Although this observation is not specific to directional networks or indoor environments, we found that taking advantage of capture can significantly improve spatial reuse.

3. DESIGN

In this section, we first provide an overview of the DIRC system and we then discuss its two key components, conflict graph generation and transmission scheduling, in detail.

3.1 System Overview

Within the larger umbrella of "indoor environments," there are a number of possible scenarios that one might consider. Rather than try to address all possible scenarios, we focus on designing and building a working wireless system for the fol-

lowing common scenario—an infrastructure wireless network where there are n APs and m clients. We assume that all of the APs have software steerable directional antennas with small reconfiguration times (e.g., phased array directional antennas), and all of the clients are omni-directional (e.g., laptops). Each directional AP has k directions, or, more precisely, antenna patterns. We use K to denote the set of directions. The APs are connected to each other through a separate, independent channel such as wired Ethernet and are coordinated by a central controller. Examples of such a scenario can be commonly found in enterprise wireless networks. The central controller uses a conflict graph to encode interfering links, and it schedules concurrent transmissions across the non-interfering links. Scheduling is done within the context of a TDMA MAC that creates upstream and downstream slots for scheduling purposes.

3.2 Building the Conflict Graph

A conflict graph is a concise way to encode interference in wireless networks. In a conflict graph, vertices represent transmissions and an edge between two vertices/transmissions means that those two transmissions cannot occur simultaneously. In an omni-directional antenna network, the construction of the conflict graph is based on direct measurements of pair-wise link throughput or frame receptions, either synchronously [4] or asynchronously [27], so the overhead is $O(n^2)$ for n transmissions.

In directional antenna networks, each vertex (i, j, k) represents the transmission from AP i to client j with direction k on the AP. Figure 1(c) shows an example of a conflict graph for the network of Figures 1(a) & (b). Since the full conflict graph for this network is very large, the figure only shows a small part of it. The lack of an edge between vertices $(S1, R1, \nearrow)$ and $(S2, R2, \searrow)$, for example, indicates that the two transmissions can occur simultaneously using direction $\nearrow$ and $\searrow$ respectively (See Figure 1(b)). The cost of building the conflict graph in directional antenna networks is higher than in omni-directional networks, because each AP can use one of k directions. As a result, a total of $O((nk)^2)$ measurements are required to build the conflict graph.

Although building the conflict graph through direct measurement is often practical in an omni-directional antenna network, the overhead is prohibitive in networks with directional antennas. For example, the Phocus Array antennas [2] have 16 factory preloaded directions; in a network with five active nodes, 2560 sets of measurements need to be collected. Each measurement takes 100 ms (we justify the figure 100 ms in Section 4.1), so an exhaustive search takes 4 minutes. Such a high overhead is especially intolerable in dynamic indoor environments, where reconfiguration may be necessary when conditions change. To make matters worse, the conflict graph is affected by the transmission rate used for the packets. More robust modulation and coding schemes are more likely to lead to a successful reception even in the presence of interfering transmissions. To reduce the overhead, DIRC instead generates the conflict graph based on a modified Signal to Interference and Noise Ratio (SINR) model that reduces the measurement overhead from $O((nk)^2)$ to $O(nk)$.

Building Conflict Graphs Using the SINR Model. The SINR model is widely used in wireless networks to predict whether a frame will be successfully received/decoded [13]. It states that a frame will be received if the ratio of signal to interference is higher than a threshold. Otherwise it will

not be decodable. The SINR model can be a good predictor given two conditions: 1) the interface hardware exhibits capture effects, and 2) there are no exposed terminals due to physical or virtual carrier sensing. The first condition has been confirmed by recent work [22] and our experimental results. If the second condition does not hold, the throughput of a link can be much lower than that predicted by the SINR model because carrier sensing prevents the senders from transmitting frames. We discuss how we ensure this condition in Section 3.2.

The SINR model incorporates several simplifying assumptions that we must adapt for our settings. First, it uses an SINR threshold that determines whether the receiver can decode either none or all the frames. In practice, one can observe intermediate loss rates. Thus, we extend the SINR model to support intermediate loss rates by using two SINR thresholds th_1 and th_2. When estimating the loss rate, we assume that any frame whose SINR is below th_1 will not be decoded, any frame whose SINR is above th_2 will always be decoded, and SINR values between the two thresholds will have an intermediate loss rate (line 12-14 of Algorithm 1). Second, the SINR model assumes that all nodes share the same SINR threshold. In our measurements, we have observed up to 3–4 dB difference in SINR thresholds in different locations. Thus, we calibrate the thresholds for each location. Note that ideally the SINR thresholds for different cards should also be calibrated by recording and comparing the SINR vs. loss rate curve for each client. However, our algorithms do not require the SINR model to predict the link throughput with high accuracy (we validate this claim experimentally in Section 5). Thus, in our experiments, we do not calibrate each individual card. Third, the SINR model assumes that the SINR thresholds do not depend on the interferers. We have observed that the SINR thresholds for captured frames are very different from that without interferers. Thus, we use two sets of SINR thresholds: one to predict the link throughput when all other APs are inactive, and one to predict the throughput otherwise.

To generate the edges of the conflict graph using the SINR model, we look at each pair of vertices. For example, a potential edge between (i_1, j_1, k_1) and (i_2, j_2, k_2) is based on the transmissions from AP i_1 to client j_1 using direction k_1 and from AP i_2 to client j_2 using direction k_2. Let $S(i, j, k)$ denote the received signal strength from AP i to client j using direction k. Then, the SINRs for both links are: $SINR_1 = S(i_1, j_1, k_1) - S(i_2, j_1, k_2)$ and $SINR_2 = S(i_2, j_2, k_2) - S(i_1, j_2, k_1)$. Finally, our modified SINR model can be applied on these SINR values to predict the reception rate for both links (note that each edge is tagged with two reception rates, one for each link).

Multi-rate Support. To support multiple data rates, we use multiple SINR thresholds, one for each data rate. We use $R = \{(r, th_1, th_2, thp)\}$ to denote the set of data rates with SINR thresholds, where r is the data rate, th_1 is the low threshold, th_2 is the high threshold, and thp is the full throughput for the data rate. Given these thresholds and the measured SINR on a link, DIRC can directly select the highest rate that can be supported on a link (line 11-18 of Algorithm 1). This is possible because the SINR measured on a link is independent of the data rate used on the interfering links.

The estimation process is shown in Algorithm 1. Note that the information needed to generate the complete conflict

---

Algorithm 1: Estimate capacity using the SINR model

Input: set of concurrent transmissions $T = \{(i, j, k)\}$, where i is the AP, j is the client, k is the direction; set of data rates with SINR thresholds $R = \{(r, th_1, th_2, thp)\}$

Output: capacity estimation $estcap$ for T

1 $estcap \leftarrow 0$
2 **foreach** $(i_1, j_1, k_1) \in T$ **do**
3     $signal \leftarrow S(i_1, j_1, k_1)$
4     $intf \leftarrow 0$
5   **foreach** $(i_2, j_2, k_2) \neq (i_1, j_1, k_1) \in T$ **do**
6   **if** $S(i_2, j_1, k_2) > intf$ **then**
7           $intf \leftarrow S(i_2, j_1, k_2)$
8   **end**
9     $sinr \leftarrow signal - intf$
10   $maxthp \leftarrow 0$
11   **foreach** $(r, th_1, th_2, thp) \in R$ **do**
12   **if** $sinr < th_1$ **then** $curthp \leftarrow 0$
13   **else if** $sinr > th_2$ **then** $curthp \leftarrow thp$
14   **else** $curthp \leftarrow thp * (sinr - th_1)/(th_2 - th_1)$
15   **if** $curthp > maxthp$ **then**
16         $maxthp \leftarrow curthp$
17   **end**
18   **end**
19   $estcap \leftarrow estcap + maxthp$
20 **end**
21 **return** $estcap$

---

graph is the set of $S(i, j, k)$ for any AP i, any client j and any direction k.

Collecting Measurements. We collect the RSSI information as follows. Each directional AP scans all its directions and transmits a burst of frames in each direction (the duration of the burst is determined empirically in Section 4). All of the clients record the RSSI readings for the frames received, and they calculate $S(i, j, k)$ as the mean RSSI at client j from AP i using direction k. When one directional AP is scanning, clients that are not associated with this AP also record the received RSSI as the interference level from that AP. Throughout this process, we assume that clients can be modified to collect the above measurements. To ensure that the measurement probes can be received at the client, during each measurement, all APs are silenced except the scanning AP, and each AP is instructed to scan in turn.

One important limitation of our measurement process is that we only consider decodable frames, but undecodable frames can also interfere; note that throughput tests do not suffer from this problem. To address this issue we proceed as follows. First, all probe frames use the minimum data rate (1 Mbps) to maximize the probability of decoding. Second, the clients report the RSSI readings of all frames even if they are erroneous. Then, if some $S(i, j, k)$ is still unavailable (i.e., no frame has been received), we use the following criteria to determine its value: 1) if client j can receive frames from AP i in other directions, we set $S(i, j, k)$ to the minimum decodable signal level, i.e., -95 dBm (we take a conservative approach), 2) otherwise, if no frames from AP i can be received by client j at all, we set $S(i, j, k)$ to NO_INTF, (we take an aggressive approach). Such a solution may still fail in scenarios where the interference level from a couple of directions could be right below the decodable threshold, and

may "hide" hidden terminal interference. In the next section, we discuss a feedback mechanism, aimed at dealing with these cases. Finally, we observed that the prediction of the link throughput $S(i, j, k)$ is usually very poor if only a few frames are available (received) for the calculation. Consequently, our algorithm treats measurements produced by fewer than 3 probes as if no frame was received.

3.3 Scheduling Using the Conflict Graph

Given the conflict graph, the second phase of the system operation is for the central controller to determine which transmissions can take place and, if they are directional, with what orientation. Such a decision is made for each timeslot and based on the actual traffic pattern across the network.

Full Scheduling Problem: The scheduling problem to optimize capacity for directional antennas is NP-hard. The size of the problem is $\binom{n}{n}k^n + \binom{n}{n-1}k^{n-1} + ... + \binom{n}{1}k$, where each $\binom{n}{i}$ is the number of ways the controller can pick i out of n transmissions, and k^i is the number of all possible antenna configurations for these i directional APs. (Even the simpler, special case of omni-directional antennas where $k = 1$, is NP-hard.)

Reducing the Search Space in Scheduling: We propose two heuristics to reduce the complexity. Algorithm 2 shows this process for one time slot. First, we order the transmissions according to the time their frames arrived at the APs (i.e., FIFO). The scheduler goes through each transmission, t, in order (line 2-24) and adds it to the schedule for the next time slot, T_{next}, if it improves the network capacity, $maxcap$, (line 21-23). This step reduces the complexity to $k^n + k^{n-1} + ... + k$. Any unscheduled transmissions will remain in the FIFO queue, and those transmissions that have been scheduled and have more frames to send will be appended at the tail. This way, starvation is prevented. In the worst case, a transmission will be scheduled after $n - 1$ time slots. At the same time, the channel will be well-utilized, since transmissions with fewer conflicts will be scheduled more often. Note that this approach of servicing requests will not lead to more unfairness than what would be expected in a normal 802.11 network; senders that experience more conflicts will have fewer opportunities to transmit.

Second, we apply a greedy algorithm to find the orientations. Instead of visiting all k^n states, the scheduler emulates an iterative process (line 6-19) where all APs start at antenna orientation 1 (line 4) and then taking turns (line 8-18) to maximize the network capacity by varying the orientation of its own antenna and by keeping the orientations of other APs fixed (line 11-17). This iterative process will converge because in each iteration, the capacity is improved (line 14). The number of rounds for the algorithm to converge depends on individual scenarios, and in the worst case, the complexity of the greedy algorithm can be $O(k^n)$. In our algorithm, we set an upper limit for 30 rounds, and, in practice, we find the algorithm converges after approximately 16.25 rounds on average. Note that we choose these two heuristics because they are simple and they perform well in our testbed. In a very large system, however, other heuristics may provide better performance. For example, algorithms like the one proposed in [7] for capacity maximization in omni-directional antenna networks, or simulated annealing may be better alternatives to meet the system goals and avoid local maxima. Indeed, we tested these two alternatives in our scheduler, but they did not provide further improvement.

Algorithm 2: Pick directions and assign timeslot

Data: FIFO queue of active transmissions $N_{all} = \{t\}$
Output: assignment of next time slot and direction $T_{next} = \{(t, k)\}$, where t is the transmission and k is the direction to be used; note that unscheduled transmissions will remain in the queue

1   $T_{next} \leftarrow \emptyset,\ maxcap \leftarrow 0,\ N_{cur} \leftarrow \emptyset$
2   **foreach** $t \in N_{all}$ **do** /* test if t can be added to T_{next} */
3      $N_{cur} \leftarrow N_{cur} \cup \{t\},\ T_{last} \leftarrow \emptyset$
4      $T_{cur} \leftarrow \{(t, k),$ where $t \in N_{cur}$ and $k \leftarrow 1\}$
5   /* each AP chooses orientation in turn, greedily increasing network capacity until convergence */
6   **while** $T_{cur} \neq T_{last}$ **do** /* test if T_{cur} changes since last iteration */
7         $T_{last} \leftarrow T_{cur}$
8   **foreach** $t' \in N_{cur}$ **do**
9            $localmaxcap \leftarrow 0,\ T_{tmp} \leftarrow T_{cur}$
10   /* find orientation of t' that maximizes capacity, with other orientations fixed */
11   **foreach** $k \in K$ **do**
12               $T_{tmp}[t'] \leftarrow (t', k)$ /* set the orientation of t' to k */
13               $curcap \leftarrow estimate\_capacity(T_{tmp})$
14   **if** $curcap > localmaxcap$ **then**
15                  $localmaxcap \leftarrow curcap$
16                  $T_{cur} \leftarrow T_{tmp}$
17   **end**
18   **end**
19   **end**
20   /* if including t can increase network capacity, then it should be scheduled for next timeslot */
21   **if** $maxcap < localmaxcap$ **then**
22         $maxcap \leftarrow localmaxcap$
23         $T_{next} \leftarrow T_{cur}$
24 **end**
25 $N_{all} \leftarrow N_{all} - T_{cur}$ /* remove scheduled transmissions */

Feedback. In some cases, the SINR model may fail to predict the link throughput, for example, when a client cannot decode frames from another interferer. Our system uses a feedback mechanism to deal with this problem. The idea of a feedback mechanism is that if a certain scheduling assignment fails to deliver the expected performance, the controller will pick the next best scheduling assignment. Note that it is possible that the prior assignment was indeed the optimal assignment but the actual measured throughput was lower than the estimated throughput. In our measurements, we observe that although there is some difference between the estimated and actual throughput, the estimation mostly preserves the ordering (i.e., schedules with higher expected throughput provide better throughput). Consequently, our algorithm samples at most three scheduling assignments and picks the best. In the greedy heuristic in Algorithm 2, we use a different starting state in each iteration, and could thus converge to a different final state.

3.4 Dynamics

A practical solution needs to account for the dynamics caused by people moving, doors opening and closing, and changes in the location of wireless devices (e.g., laptops). We identify

two types of dynamicity: short-term dynamics caused by multipath fading and other short-lived environmental changes, and long-term dynamics caused by more permanent changes.

Short-term dynamics cause the RSSI and link throughput measurements to fluctuate quickly. The APs should not respond to such changes, since the coherence time is too short for our algorithm to rescan and reconfigure. Using a moving average is a common mechanism to obtain more stable and statistically sound readings.

Long-term changes, on the other hand, should trigger the regeneration of the conflict graph based on new measurements. In some cases, changes can be readily observed (e.g., loss of throughput on a link or packet losses due to collisions) while in other cases they cannot (e.g., a previously inefficient transmission schedule is now attractive because a door was closed). As a result, pure failure-based re-scanning is not practical, and we need to periodically rebuild the conflict graph (Section 4). The challenge is to find a good trade-off between a rapid response to long-term dynamics while ignoring short-term effects. Note that periodic rescanning does not help with the SINR model mis-predictions discussed above, since these are caused by limitations of the node measurement capabilities.

4. DIRC IMPLEMENTATION, OPERATION

In this section we present the details of DIRC's protocol of operation. Our DIRC implementation is based on the Madwifi driver and Atheros 5212 wireless cards, present on both clients and APs. The controller is a Linux machine that can access all APs through the wired Ethernet.

4.1 Measurement Collection and Scanning

DIRC builds the conflict graph based on RSSI measurements that are collected across the network. To maintain an accurate conflict graph, it is important that the RSSI measurements adapt to short-term as well as long-term dynamics of the environment while minimizing the overhead on the system performance. Thus, DIRC periodically instructs all APs to scan for updated RSSI measurements, or a scan can also be explicitly requested by the APs when the measured throughput on any link drops below a set threshold.

The protocol between the APs and the central controller to initiate a scan is as follows. Scans are requested by the AP through a *request-to-scan* message. The controller initiates the scan by sending a *clear-to-scan* message to the AP; scans in response to AP requests are dampened and take place at most once a second. The controller gives higher priority to scan requests than to transmissions, since efficient transmissions require an accurate conflict graph. Each scan probe includes the AP identifier and the direction used for the probe frame. When the client receives a scan probe, it records the RSSI reading of the frame. The client will send this measurement back to the AP the next time it is allowed to access the medium. When the APs receive scan responses, they forward them to the controller. To reduce the probability that scan responses are lost, all APs that overhear the scan responses forward them to the controller.

There are four primary system parameters that determine the RSSI measurement collection: the interval and duration for periodic scanning, the interval over which link throughput is measured, and the threshold used to explicitly request a scan.

Link	Win. Size (ms)	5	10	100	1000
Least Bursty	RSSI (dB)	0.2	0.16	0.05	0.03
	Reception (%)	25	21	15	5
Most Bursty	RSSI (dB)	2.2	1.5	1.3	1
	Reception (%)	50	35	26	16

Table 1: Allan deviation of RSSI and frame reception rate for different intervals

Interval	500 ms	1 s	5 s	30 s	10 m
Mean Thp (Mbps)	13.9	19.3	19.5	18.3	17
Overhead (%)	48	24	4.8	0.8	0.04

Table 2: Effect of scanning intervals on link throughputs

Link Throughput Measurement Interval and Threshold: An AP can explicitly request a scan when it detects a significant drop in throughput on any link. Table 1 shows the Allan deviation in the reported RSSI and packet reception rate across different time intervals for the least bursty and most bursty link in our testbed. The antennas are oriented based on the *Max Cap* configuration (Section 2.2). We find that the links exhibit varying behavior. For example, for a 100 ms window, the least bursty link only has a 15% deviation in reception rate while the most bursty link deviates by as much as 26%. Thus, it is difficult to select a threshold that works for all links and maintaining per-link thresholds would lead to a significant overhead. This led us to an alternative approach, where the detection thresholds are set to a conservative value and we rely on fixed-cost periodic scanning to prevent the system from operating in a suboptimal state. We measure link throughput over 100 ms intervals (which span five 20 ms time slots) and set the detection threshold for explicit rescanning to be 50% of the reception rate during the first window after a scan.

Periodic Scan Interval and Duration: The periodic scanning should be frequent enough to maintain an updated conflict graph and should not significantly impact the network performance. Table 1 shows that the deviations of the RSSI measurements are stable even for small time intervals, e.g., even for the most bursty link it is only about 2.2 dB for 5 ms. The RSSI measurements are much more stable than reception rate because they are obtained when all other APs are silenced while throughput measurements are obtained when multiple APs are active. This means that a 4 ms scan duration will accommodate most short-lived changes in RSSI readings.

To determine the scanning interval, we evaluate the overhead of scanning on the link throughputs. Table 2 shows the mean link throughput and corresponding scanning overhead for different scanning interval values. We see that when the scanning interval is too small, e.g., 500 ms, the overhead of the protocol is too high, reducing the link throughputs; while if the scanning interval is too large, e.g., 10 min, the system is unable to escape suboptimal states. The table shows that 5 seconds is a reasonable tradeoff, incurring less than 5% overhead while providing up-to-date RSSI information.

This system design is optimized for nomadic clients, i.e., clients that may appear at multiple locations across the network (laptops) but are used in stationary positions. In Section 5 we show that DIRC can also adapt to scenarios

where users are walking with their devices.

4.2 Two-Phase TDMA Protocol

DIRC relies on the intelligent scheduling of transmissions across the network. A TDMA based MAC protocol is used to synchronize the downlink and uplink traffic. Additionally, DIRC's TDMA protocol also adapts to different traffic workloads instead of having a fixed time allocation for the uplink and downlink traffic.

All nodes in the network, including APs, clients, and the controller are time synchronized. On the clients and APs, clock synchronization is implemented in the Atheros 5212 card through beacon messages. Since the driver has access to the hardware clock, wireless nodes can transmit at the time slot boundaries with microsecond accuracy. Our TDMA-based MAC operates in two phases; the first phase is dedicated to directional transmissions, while the second phase accommodates omni-directional nodes, including clients and legacy APs. We term these two phases of our protocol `dirc-tx` and `omni-tx`.

During the `dirc-tx` phase, the centralized controller schedules AP-to-client transmissions, with carrier sensing disabled at the APs. This is done by raising both the receiver and transmitter CCA thresholds, allowing the APs to send frames even in the presence of interference. Furthermore, the interframe spacing (IFS) is set to the minimum possible value since it is not needed in a TDMA schedule. We also disable link-layer ACKs since they introduce mixed traffic from both directions. Similar to previous work with a TDMA-based MAC [19], our protocol uses block link-layer ACKs, i.e., a bitmap that encodes all frames received in the previous time slot, transmitted during the `omni-tx` phase.

Note that though individual channels are symmetric due to channel reciprocity, the SINR values are not, i.e., they do not apply if we switch the roles of all transmitters and receivers. For this reason, the `omni-tx` phase uses the traditional 802.11 protocol, with all nodes, including APs, using omni-directional antennas.

By default, our system operates on 16 ms `dirc-tx` phases and 4 ms `omni-tx` phases. There are two reasons for this choice: 1) the majority of the traffic in enterprise WLANs tends to be downlink, from the APs to the clients, and 2) 4 ms is the minimum slot time that can be scheduled reliably in our implementation. A fixed time allocation for directional and omni-directional transmissions, however, severely limits the types of workload that can be accommodated by the network. As a result, DIRC uses measurements to identify the right time division between the two phases. The controller periodically collects information about the queue length on clients and APs. It then calculates the client-AP backlog ratio and uses this to dynamically adjust the slot time sizes. For simplicity, our controller picks one of the three up-down ratios: 16-4 ms, 10-10 ms, and 4-16 ms. Starting from a split of 16-4 ms (down-up), the network will switch to 10-10 ms if the ratio drops below 1, and 4-16 ms if the ratio drops below 0.25. The same holds in the reverse direction. The slot time size are adjusted at most once every second to avoid thrashing, and to reduce the probability of inconsistencies among APs and clients.

4.3 Frame Transmission and Reception

The AP transmits frames destined to a particular client as follows. First, the AP sends a *request-to-send* to the centralized controller over the wired Ethernet connection. This request-to-send includes the AP and client identifiers, and the queue length on this AP; the AP discards additional requests containing the same AP and client identifiers.

The controller collects all of the requests-to-send and calculates a time slot assignment for the next `dirc-tx` time slot using Algorithm 2. The controller sends a *clear-to-send* message (over the wired Ethernet) to each AP that it has scheduled for the upcoming time slot. The clear-to-send message tells the AP which client it should transmit to next, what direction and what data rate to use when transmitting its frames. We chose to serve a single client during the entire time slot, since serving multiple clients would require a change in the antenna orientation. This may force clients to wait a time proportional to the number of clients associated with the same AP. We could reduce this time by (i) using finer granularity time slots (4 ms was the minimum time slot we could implement in our testbed), or (ii) allowing multiple clients to be served using the same antenna orientation (either because they are in the same sector, or happen to receive adequately strong signals under the same antenna configuration). Note that if no AP is active, and there is a change in the time slot split between directional and omni-directional traffic, the controller will assign the APs to send the new split information omni-directionally. This split information could be easily integrated in the Beacon frames.

When the `dirc-tx` phase arrives, each AP checks to see if it is clear-to-send. If not, the AP waits until the next time slot and checks again. If the AP is scheduled to send, it un-buffers the frames to that client, sets the direction, disables CCA, and transmits frames until the end of the `dirc-tx` phase. The AP appends a trailer to each frame that includes a frame identifier for a block ACK, and the split for the next time slot. Each AP buffers all unacknowledged frames. The AP estimates the transmission time for each frame that it transmits (or retransmits), so it can make sure that the total transmission time for the frames put on the hardware queue does not exceed the time slot duration.

During the `dirc-tx` phase, the clients receive frames from the APs and record all the frame identifiers in a bitmap. Then, in the `omni-tx` phase, each client first transmits a block ACK back to the AP. The use of block ACKs can result in out-of-order frames which significantly reduces the performance of TCP. To address this, frame reordering is implemented in the driver.

After the client transmits its block ACKs, it transmits any other frame. The queue length on that client is piggybacked in each frame, and forwarded to the controller by the AP. During this `omni-tx` phase, the APs all set their antennas to omni-directional mode, and start receiving frames from clients. When a block ACK is received at the AP, the corresponding frame is removed from the receive buffer. Note that in this phase, the wireless nodes operate using CSMA, e.g., link-layer ACKs are enabled and the default Madwifi rate adaptation algorithm is used.

4.4 Association

It is important for DIRC to quickly bootstrap a new client into the network and update the conflict graph while minimizing the overhead on the system. The number of probes needed to generate the conflict graph are $O(nk)$ and may result in a high overhead when the association and disassociation rate is high. Thus, in our protocol, we take the
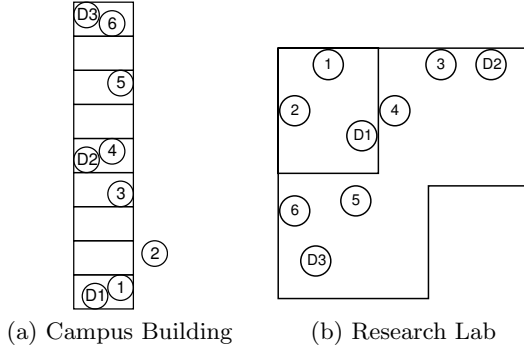
(a) Campus Building (b) Research Lab

Figure 4: Testbed layout

(a) Estimated vs. actual packet delivery rate

(b) DIRC's algorithm

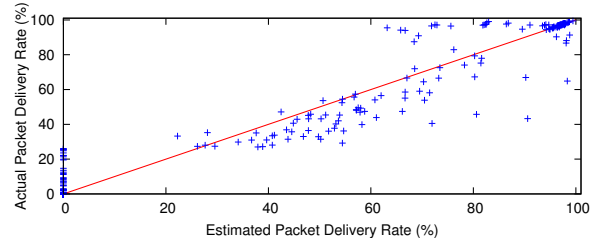Figure 5: Performance of the DIRC's scheduling and direction selection algorithm

lazy approach where measurements are delayed until the next scanning period. The lazy approach incurs minimum overhead for client association, but it has the downside that the conflict graph will be incomplete for this client. The controller will not schedule transmissions when the part of the conflict graph for that transmission is incomplete. Thus, when the client first associates with an AP, the AP is not able to initiate frame transmissions to the client until the next scanning period. In practice, we do not expect this delay to be a problem because the client typically associates with an AP upon boot time. The inherent delays in this process, e.g., the clock synchronization in wireless hardware can take several seconds, is likely to mask the controller delays. Certain data traffic that may follow the association, such as DHCP requests, need to be delayed until after synchronization and scanning is finished.

5. EVALUATION

In this section, we evaluate DIRC's centralized scheduling algorithm and overall performance using a working implementation of DIRC in two real-world settings. We present three key results: 1) DIRC's centralized algorithm can achieve close-to-optimal transmission scheduling and antenna orientations with very low overhead. 2) The end-to-end DIRC implementation works well in practice. In a network with three directional APs, UDP performance improves by 65% over the max SNR approach (100% over using only omni-directional antennas), and TCP performance improves by 40% over the max SNR approach (42% over omni-directional antennas). 3) DIRC can handle node mobility and dynamic traffic patterns.

5.1 Experimental Setup

Our experimental testbed consists of three directional antennas (Fidelity Comtech Phocus Array [2]) and six omni-directional nodes (Linux desktops and/or laptops). Although the number of directional antennas appears to be small, the tested configuration would be what one would see in an enterprise network in one of the multiple orthogonal frequencies, possibly reflecting performance in networks that may feature 3 times (in 2.4 GHz) or 11 times (in 5 GHz) more directional nodes. The phased array antennas have eight antenna elements and the antenna pattern has a beamwidth of 45°. The antenna can be electronically steered at angular displacement of 22.5°, providing a total of 16 directional antenna states
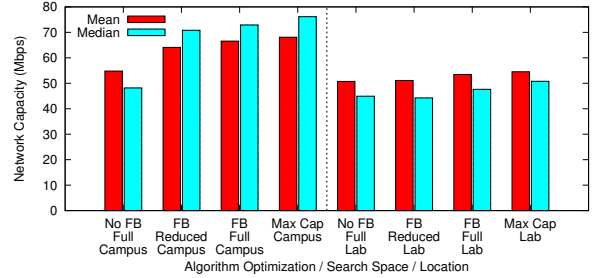
and one omni-directional antenna state. Since the goal of the protocol is to improve spatial reuse instead of extending communication range, the directional patterns we picked have similar antenna gain to that of the omni-directional pattern (only around 2 dB difference), and the side lobes of the directional patterns are very small.

We conducted our experiments in two physical testbeds: a campus and a research lab. The maps and the locations of the nodes are shown in Figure 4. Nodes $D1 - D3$ are the directional antennas, and the omni-directional nodes are labeled $1 - 6$. Since the experiments use the 2.4 GHz band, they share the channel with other 802.11 devices.

Our evaluation consists of two parts. In the first part, we use collected measurements to evaluate the accuracy of the SINR model and the emulated performance of DIRC's core algorithm. The second part reports on the working end-to-end implementation of DIRC, with all associated overheads.

5.2 Direction Selection and MAC Strategies

In this subsection, we first evaluate the effectiveness of the SINR based model to predict link throughput. We then evaluate various direction selection and MAC strategies using offline measurements, including the performance of DIRC's centralized algorithm for orienting the antennas and scheduling transmissions as presented in Section 3. In these experiments, we collect measurements first and then apply various algorithms on the collected data.

The data measurements are collected as follows: 1) The APs broadcast UDP traffic, and all clients record the RSSI and link throughputs from the APs. This approach helps to reduce the number of measurements that need to be collected. 2) To evaluate the effect of CSMA, we carry out one set of experiments where all three APs transmit using the default CCA threshold across all possible antenna orientations (i.e., $17^3 = 4913$ measurements). 3) To determine the maximum achievable performance, we run a set of experiments to

measure capacity under all possible schedules and antenna orientations (i.e., $\binom{3}{1}17 + \binom{3}{2}17^2 + \binom{3}{3}17^3 = 5831$ measurements). Here, we disable CCA on the APs to ensure that they all transmit simultaneously. Note that the maximum capacity results are drawn out of this particular experiment. 4) We run the above-mentioned experiments twice for each of two data rates: 36 and 54 Mbps. These data rates are chosen because the AP-client distance is relatively short in our testbed. In summary, in this offline experiment, we collect a total of $(5831 + 4913) * 2 = 21488$ measurements, and each individual measurement runs for 1 second. The whole experiment spans several hours, and the environment may change during the experiment. Since we apply the various algorithms on the same data set, all tested algorithms are consistently exposed to the same changes in the environment.

To evaluate the effectiveness of the SINR model, we calculate the correlation coefficient of the estimated throughput using the SINR model and the actual measured throughput. Figure 5(a) shows a scatter plot for a single client. The coefficient across all the nodes in the testbed is 0.98 which is sufficient for DIRC to pick the right schedules.

Client to AP association is currently not addressed by DIRC. To evaluate overall network capacity, however, we need to manually associate clients with APs. We do that by looking through all possible associations of 3 clients (out of the 6 client locations in our testbed) with the 3 directional APs. For each such configuration, we identify the configuration that leads to the highest overall capacity and use this as the most reasonable association pattern.

Before presenting the results, we should note that since we use broadcast frames to collect the measurements, our data does not incorporate the effect of rate adaptation. Thus, we cannot directly evaluate any interactions between the simple heuristics and rate adaptation. Therefore, in our evaluation, we emulate optimal rate adaptation for these heuristics, picking the best data rate for each individual link that achieves the max throughput. This provides an upper bound on the performance of these competing designs. Note that this does not apply to DIRC's algorithm since DIRC selects the data rate for each individual link as part of the algorithm.

Figure 5 compares the performance of three variants of DIRC's centralized algorithm to the maximum achievable capacity. *Reduced*, in Figure 5, refers to the scheduling heuristic for the reduction of the search space, while *Full* refers to the full scheduling algorithm. *FB* and *No FB* refers to the use of feedback to avoid problems with SINR-based throughput prediction. Max Cap (described in Section 2.2) is the maximum achievable capacity. DIRC's centralized algorithm is shown as FB/Reduced. The results show 1) the effectiveness of the SINR model in picking schedules and orientations, 2) the effectiveness of the feedback mechanism (FB), and 3) how the scheduling heuristics affect the network capacity and computation cost.

The results show that directly using the SINR model (No FB/Full) to pick the schedules often performs much worse than Max Cap. After applying the feedback mechanism that improves on the SINR model predictions (FB/Full), the performance is very close to that of Max Cap. DIRC's algorithm (FB/Reduced) leads to a slight reduction in network capacity, but its heuristics are able to reduce the number of configurations searched from 4912 down to 283 and 255 in two scenarios—a factor of 17 improvement.

5.3 End-to-End Performance

In this section, we evaluate the protocol and implementation of DIRC. We examine how much overhead is incurred, where the gain comes from, how well it interacts with transport protocols (especially TCP), and how well it can deal with environment dynamics and node movement. In this set of experiments, we deployed DIRC on the campus testbed. Since this set of experiments evaluates the end-to-end protocol performance that depends on the client responses, (e.g., block ACKs, scanning responses, TCP ACKs), we must pick a static AP-client association. For simplicity, we associate clients with the closest AP: $1, 2 \rightarrow D1; 3, 4 \rightarrow D2; 5, 6 \rightarrow D3$. We measure UDP and TCP throughput using the standard *iperf* utility. The packet generation rate for UDP is set to 30 Mbps (TCP manages the rate itself).

Figure 6 shows the UDP and TCP performance of DIRC. In these experiments, the directional APs run *iperf* to each of their clients for 10 seconds, and we repeat this end-to-end measurement for each of the AP's clients (total of eight experiments). We report mean and median capacity.

Although the UDP packet generation rate is 30 Mbps, the maximum actual throughput that DIRC can achieve is only 27 Mbps. This is because by default DIRC reserves 20% of the airtime for client transmissions, though it recovers some of that lost throughput by reducing the inter-frame spacing (IFS); as the maximum effective throughput for 54 Mbps is approximately 32 Mbps, the maximum throughput DIRC can achieve is approximately 27 Mbps. Consequently, the maximum network capacity for 3 directional transmitters is approximately 81 Mbps. Our results show that the median UDP capacity of DIRC is about 76% of this upper bound. Note that CSMA MAC does not have the 20% loss from airtime reserved for the client (nor the gain from reduced IFS), allowing it to achieve close to 30 Mbps (out of 90 Mbps). Figure 6(b) shows the breakdown of DIRC's gain and overhead, where the solid part shows the gain and the patterned part shows the overhead. It shows that much of the benefits come from directionality of the antennas, and the major overhead for UDP traffic is the reserved 4 ms omni-tx phase. The TCP performance of DIRC is approximately 45 Mbps, which is a 40% improvement over max SNR and 42% over the default omni-directional antennas. The improvement is not as high as UDP, which can be due to 1) the interactions between TDMA and TCP, and 2) the split between two phases is not flexible enough. Split-TCP can be used to solve the problem, which we will explore in future work.

Next, we study the performance of the protocol itself. We pick two links from the campus testbed: link 1 from $D1$ to client 2 and link 2 from $D2$ to client 3. We run *iperf* UDP tests with a packet rate of 30 Mbps for 10 seconds on both APs. We present the throughput of each link and the orientation of the directional APs in Figure 7(a). During $t = 0$ to 5, link 1 is the only transmitter and uses direction 9, which is the max SNR direction. At $t = 5$, scanning is carried out and link 1 suffers a drop in throughput due to the measurement collection. Since there are two transmissions now, the controller decides that two APs can transmit simultaneously, using directions 14 and 3. At $t = 10$, link 1 stops transmitting, and since link 2 is now the only transmitter, it can use max SNR direction (i.e., 2).

Figure 7(b) shows the protocol behavior when we use a laptop as the client and move from location 4 to location 2 in the campus setup, while associated with $D2$. Note
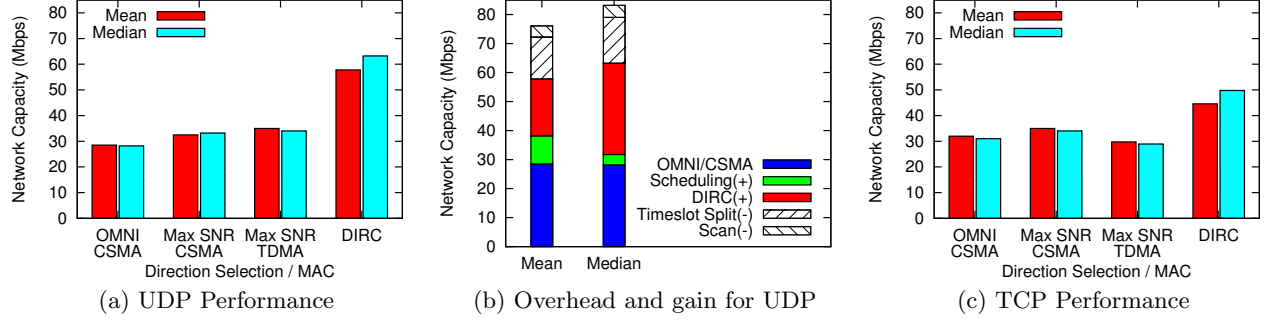
(a) UDP Performance   (b) Overhead and gain for UDP   (c) TCP Performance

Figure 6: UDP and TCP performance of DIRC

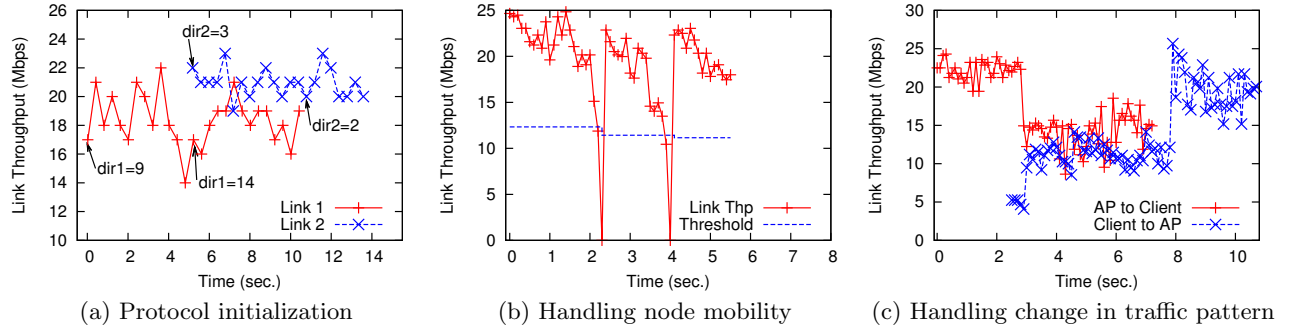(a) Protocol initialization   (b) Handling node mobility   (c) Handling change in traffic pattern

Figure 7: Protocol behavior during initialization, handling node mobility and change in traffic patterns.

that between any two scanning events, the movement of this laptop will only affect its own link throughput, but not the throughputs of other transmissions. Furthermore, for multiple mobile users, one rescanning will update the RSSI information for all such mobile devices. Thus, we only enable one transmission in this experiment to simplify our presentation. We run the UDP *iperf* test from $D2$ to the client at the 30 Mbps data transfer rate. The detection threshold for re-scanning is set to 50% of the link throughput of the first window after each scan. At time $t = 0$, the client is at location 4, as it moves towards location 2, the link throughput drops below the detection threshold and triggers a re-scanning at $t = 2.3$; this process is repeated and the re-scanning is triggered again at $t = 4$; finally it stabilizes at around $t = 4.5$. The drops in throughput at $t = 2.3$ and $t = 4.0$ are due to scanning.

Figure 7(c) shows how DIRC can respond to changes in traffic patterns by adjusting the amount of airtime allocated to APs versus clients. In this experiment, we let client 4 associate with AP $D2$ and run UDP *iperf* on both nodes to each other (bidirectional UDP), and we do not enable other transmissions. We let the second data transfer start several seconds later than the first data transfer. The figure shows the link throughputs for both directions over time. At time $t = 0$, only the AP is transmitting and the scheduler allocates most of the airtime to the AP. At $t = 2.3$, the client starts transmitting. The queue length on the AP reaches around 30, and on the client it reaches around 54; these lengths exceed DIRC's 1/1 threshold, causing the controller to split the airtime equally between the AP and client at

$t = 2.8$. At $t = 7.2$, the transfer from AP to client ends, and at $t = 7.8$, the controller re-allocates the airtime again, but favoring the client.

6. RELATED WORK

Existing work on directional antennas mostly assumes outdoor applications, where the dominating line-of-sight (LOS) path should be used for communication. This work includes MAC layer protocols for directional antennas [15, 10, 16, 26, 9, 20, 28, 5], using directional antennas in vehicular networks [18], and using directional antennas to form long distance Wi-Fi links [19, 21, 8]. In this paper, we show that because of the unique characteristics of rich scattering in indoors environments, these outdoor solutions do not perform well indoors. In particular, past schemes largely orient antennas by maximizing the signal strength (SNR), which does not apply indoors. One relevant effort [25] uses the capabilities of digital adaptive antennas to direct pattern nulls towards sources of interference. This is, in fact, similar to our idea of reducing interference to other transmissions.

Conflict graphs are widely used in wireless networks to encode interference information [27, 4]. However, existing approaches to generate conflict graphs based on omni-directional antennas are not suited for directional networks and incur a significant overhead. In [24], the authors discuss the definition and generation of directional conflict graphs, with additional consideration of side lobes, but the primary usage is still in an outdoor ad-hoc network with little or no multipath.

There is also extensive work on TDMA scheduling in wire-

less networks using conflict graphs or physical layer interference information [7, 14]. Some of this work is complementary to ours and might supplement or replace our basic scheduling heuristics in future versions of our system. Their integration, however, requires further study since our indoors scenario combined with directional antennas creates a much larger configuration state space.

In addition to spatial beamforming based on phased array antennas, MIMO beamforming can also be used to increase spatial reuse. Recent advances in multiuser-MIMO (MU-MIMO) techniques [12] enable a single AP to serve multiple clients simultaneously by leveraging independent multipath channels at each client. MU-MIMO techniques require channel state feedback across all the clients in the network to characterize the multipath channel [3] and use advanced signal processing techniques like dirty paper coding and block diagonalization [11] to simultaneously transmit to all or a subset of clients.

Recent work [6] has studied the performance of directional antennas in indoor environments, including directionality, spatial reuse, and localization. The authors observe that even though directionality is weaker (i.e., the resulting radiation pattern is less directional) in indoor environments, directional antennas can still offer enough isolation to improve performance over omni-directional antennas. They also observe that the LOS direction does not always have the strongest signal. The paper, however, does not consider how to make use of directional antennas indoors, even though their measurement results suggest that it is more challenging than in LOS environments. In short, their results provide excellent motivation for our work on DIRC.

7. CONCLUSION

Conventional wisdom has been that directional antennas are ineffective indoors, due to the complex scattering and multipath effects of the indoor RF environment. In this work, we demonstrate that this is not the case. DIRC is an indoor directional antenna system that is able to improve spatial reuse and system-wide network capacity in indoor spaces where wireless devices are densely deployed. In a nine node testbed, DIRC provides a 100% improvement for UDP and 42% improvement for TCP over omni-directional solutions, while being able to deal with changing workloads and environments. The core of DIRC is an algorithm that identifies close-to-optimal antenna orientations with low overhead, and a MAC protocol that coordinates transmissions to maximize transmission concurrency. We implemented this design in a network using phased array directional antennas, and our experiments show that DIRC works well in practice.

8. REFERENCES

[1] *SiBEAM: Wireless Beyond Boundaries (www.sibeam.com).*

[2] *Fidelity Comtech www.fidelity-comtech.com.*

[3] *IEEE 802.11n Draft.*

[4] N. Ahmed, U. Ismail, S. Keshav, and K. Papagiannaki. Online estimation of RF interference. In *CoNEXT*, 2008.

[5] L. Bao and J. J. Garcia-Luna-Aceves. Receiver-oriented multiple access in ad hoc networks with directional antennas. *Wirel. Netw.*, 11(1-2), 2005.

[6] M. Blanco, R. Kokku, K. Ramachandran, S. Rangarajan, and K. Sundaresan. On the effectiveness of switched beam antennas in indoor environments. In *PAM*, 2008.

[7] G. Brar, D. M. Blough, and P. Santi. Computationally efficient scheduling with the physical interference model for throughput improvement in wireless mesh networks. In *MobiCom*, 2006.

[8] K. Chebrolu and B. Raman. FRACTEL: a fresh perspective on (rural) mesh networks. In *NSDR*, 2007.

[9] R. R. Choudhury and N. H. Vaidya. Deafness: A MAC problem in ad hoc networks when using directional antennas. *ICNP*, 2004.

[10] R. R. Choudhury, X. Yang, R. Ramanathan, and N. H. Vaidya. Using directional antennas for medium access control in ad hoc networks. In *MobiCom*, 2002.

[11] T. Cover and J. Thomas. *Elements of Information Theory.* J Wiley and Sons Inc, 1991.

[12] D. Gesbert, M. Kountouris, R. Heath, C.-B. Chae, and T. Salzer. Shifting the MIMO paradigm. *Signal Processing Magazine, IEEE*, 2007.

[13] P. Gupta and P. R. Kumar. The Capacity of Wireless Networks. In *IEEE Trans. on Information Theory*, 2000.

[14] K. Jain, J. Padhye, V. N. Padmanabhan, and L. Qiu. Impact of interference on multi-hop wireless network performance. In *MobiCom*, 2003.

[15] Y.-B. Ko, V. Shankarkumar, and N. H. Vaidya. Medium access control protocols using directional antennas in adhoc networks. In *INFOCOM*, 2000.

[16] T. Korakis, G. Jakllari, and L. Tassiulas. A MAC protocol for full exploitation of directional antennas in ad-hoc wireless networks. In *MobiHoc*, 2003.

[17] R. Mahajan, M. Rodrig, D. Wetherall, and J. Zahorjan. Analyzing the MAC-level behavior of wireless networks in the wild. In *SIGCOMM*, 2006.

[18] V. Navda, A. P. Subramanian, K. Dhanasekaran, A. Timm-Giel, and S. Das. MobiSteer: using steerable beam directional antenna for vehicular network access. In *MobiSys*, 2007.

[19] R. K. Patra, S. Nedevschi, S. Surana, A. Sheth, L. Subramanian, and E. A. Brewer. WiLDNet: Design and implementation of high performance WiFi based long distance networks. In *NSDI*, 2007.

[20] A. Prabhu and S. Das. Addressing deafness and hidden terninal problem in directional antenna based wireless multi-hop networks. *COMSWARE*, Jan. 2007.

[21] B. Raman and K. Chebrolu. Design and evaluation of a new MAC protocol for long-distance 802.11 mesh networks. In *MobiCom*, 2005.

[22] N. K. Santhapuri, J. Manweiler, S. Sen, R. R. Choudhury, S. Nelakuditi, and K. Munagala. Message in Message MIM: A case for reordering transmissions in wireless networks. In *HotNets*, 2008.

[23] A. Sheth, S. Nedevschi, R. K. Patra, S. Surana, E. A. Brewer, and L. Subramanian. Packet loss characterization in WiFi-based long distance networks. In *INFOCOM*, 2007.

[24] K. Sundaresan and R. Sivakumar. A unified MAC layer framework for ad-hoc networks with smart antennas. *IEEE/ACM Trans. Netw.*, 15(3), 2007.

[25] K. Sundaresan, W. Wang, and S. Eidenbenz. Algorithmic aspects of communication in ad-hoc networks with smart antennas. In *MobiHoc*, 2006.

[26] M. Takai, J. Martin, R. Bagrodia, and A. Ren. Directional virtual carrier sensing for directional antennas in mobile ad hoc networks. In *MobiHoc*, 2002.

[27] M. Vutukuru, K. Jamieson, and H. Balakrishnan. Harnessing exposed terminals in wireless networks. In *NSDI*, 2008.

[28] Z. Zhang. DTRA: directional transmission and reception algorithms in WLANs with directional antennas for QoS support. *Network, IEEE*, 19(3), May-June 2005.