

Adaptive Antenna Adjustment for 3D Urban Wireless Mesh Networks

Guoqing Yu, Wei Wang, James Yong, Ben Leong, and Wei Tsang Ooi
Department of Computer Science, National University of Singapore

Abstract—We design and evaluate a new type of wireless mesh nodes called *Dyntenna* nodes that are equipped with steerable omnidirectional antenna. Designed for 3D wireless mesh networks, these nodes adaptively adjust the antenna orientation to increase throughput by improving the Received Signal Strength Indicator (RSSI) reading between nodes. We demonstrate the importance of being able to programmatically orient the antenna, by presenting the measurement results from our 3D urban mesh testbed. We propose a simple antenna adjustment algorithm that can improve the throughput for 26% of one-hop paths and 35% of multi-hop paths by a median value of 31% and 46%, respectively. Our algorithm converges quickly and typically probes less than 10% of all possible antenna orientations on average.

I. INTRODUCTION

In a dense urban environment with many tall buildings, it is often not practical to deploy an 802.11 wireless mesh network in such a way that the nodes are located on a 2D plane (i.e. on the roof [16] or on poles/trees [4]). Instead, it is often necessary to place the nodes on tall buildings at different heights, forming a *3D mesh network*, where the antennas may not have direct line-of-sight to one another.

One problem of such 3D mesh networks is that the *default vertically-upright orientation is not always optimal*, because the popular omnidirectional antennas are omnidirectional only in a plane and have limited vertical beamwidth [3, 18]. While it is possible to manually calibrate the antenna orientation to improve the connectivity during deployment, manual calibration would be extremely time-consuming for a large mesh network. Moreover, because the optimal orientation also depends on the environmental conditions, which may be transient (e.g., rain and natural fluctuations in wireless connectivity), it is impractical to re-calibrate the antenna orientation whenever the optimal orientation changes. In addition, the optimal orientation could also change with different traffic patterns.

To avoid the frequent manual re-calibration of antenna orientation, we designed mesh network nodes with mechanically-steerable 2D omnidirectional antenna for our 3D mesh testbed. We call these nodes *Dyntenna* nodes. A *Dyntenna* node can programmatically orient its antenna to one of 121 possible orientations. While there is a large body of work on steerable directional antennas [10, 11, 12, 13, 17, 22], their focus is mainly on the simple single-hop wireless LAN scenario. To the best of our knowledge, we are the first to dynamically adjust the orientation of 2D omnidirectional antennas in a 3D mesh network, and it was not immediately clear what would be the optimal antenna orientation in our context, or how to efficiently find the optimal orientation. Besides, most of

the above-mentioned available work use commercial phased array antenna [7], which are expensive. Our proposed *Dyntenna* nodes are much cheaper, making wide-scale deployment feasible.

Our antenna adjustment algorithm is based on the following key insights from our initial measurement study: (i) the Received Signal Strength Indicator (RSSI) between nodes changes smoothly as the antenna orientation is gradually adjusted; (ii) for each link, there exists an RSSI threshold above which the link becomes reliable; and (iii) while RSSI values (wireless connectivity) do change over time, they change on a timescale that is slow enough (i.e., in the order of hours on average) that makes automated antenna adjustment practical.

The antenna adjustment algorithm incorporates a sampling technique that allows us to interpolate the RSSI over a large number of orientations by probing only a small number of orientations. Also, the prediction of whether a link would have good connectivity is based on the inferred relationship between RSSI and packet delivery ratio (PDR) from the sampling. The algorithm finally adjusts the antenna to a new orientation that achieves maximum total RSSI between a *Dyntenna* node with all its neighbors, subject to the constraint that none of the links fall below a minimum quality threshold.

In this paper, we make the following contributions. First, we designed and built a 3D mesh testbed with prototype *Dyntenna* nodes equipped with steerable 2D omnidirectional antenna. Second, we conducted a measurement study on the physical characteristics of such a 3D mesh network, focusing on understanding the effect of antenna orientation on signal strength and the variations of the optimal antenna orientation over time. Finally, we designed, implemented, and evaluated an efficient antenna adjustment algorithm, demonstrating that by exploiting steerable omnidirectional antennas, we can improve the throughput for 26% of one-hop paths and 35% of multi-hop paths by a median value of 31% and 46%, respectively.

We show that steerable omnidirectional antennas can improve the performance of existing 3D mesh networks by adding an important dimension to the design space. We believe that our work lays the foundation for a new class of 3D mesh networks with steerable omnidirectional antenna, and for more in-depth study into the integration of such antennas with mechanisms like rate adaptation and routing.

The rest of this paper is organized as follows: in Section II, we introduce our 3D mesh testbed and the hardware of the *Dyntenna* node. In Section III, we present the measurement study on our testbed along with the main insights gained from

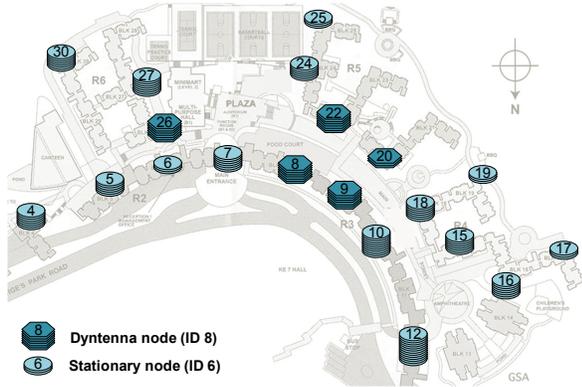


Fig. 1. Overview of 3D wireless mesh testbed. The height of a shape is proportional to the height of the corresponding actual deployment.

the testbed. In Section IV, we describe our antenna adjustment algorithm and in Section V, we present a performance evaluation of the system on a real 20-node mesh testbed. In Section VI, we contrast our work with similar previous work in the literature. Finally, we conclude in Section VII.

II. DYN TENNA TESTBED

Our 3D mesh testbed is deployed in a student residential complex at a university, and the nodes are installed at different levels of the various apartment blocks. There are 20 nodes in the mesh, of which 5 are Dyntenna nodes (blocks 8, 9, 20, 22, and 26) and the remaining are traditional static nodes with antennas in the default vertically-upright orientation. The physical layout of the testbed is shown in Fig. 1, where we also represent the height of each node graphically.

Each node in the testbed consists of a PC Engines ALIX system board with a 500MHz x86 CPU and Atheros-based 802.11 a/b/g wireless cards, running *OpenWrt*. The antenna is a simple rubber duck antenna [1], with 360° horizontal beamwidth and 90° vertical beamwidth. For a non-Dyntenna node, its antenna is mounted outdoors and fixed in the vertically-upright orientation, and is connected via a coaxial cable to the system board that is placed indoors.

The antenna of a Dyntenna node is mounted on a physical moving base, which is attached to one end of a metal bar (see Fig. 2(a)). The other end of the metal bar is mounted on a wall outside of a building. Through a USB cable, the base is controlled by the system board, which is connected to our lab using the campus LAN. The base (see Fig. 2(b)) is the only movable portion of the node and it has two degrees of freedom: along the X and the Y axes. The movement is controlled by two motors that can be activated simultaneously so that the antenna can move diagonally in a single step. The base contains a sensor that can measure the tilt of the antenna and the sweep angle on each axis is 90° (from -45° to $+45^\circ$), with a movement precision of $\pm 2^\circ$. The default antenna orientation is vertically upright. Each Dyntenna prototype currently costs about USD100 to fabricate, but we believe that this cost can be significantly reduced for mass production.

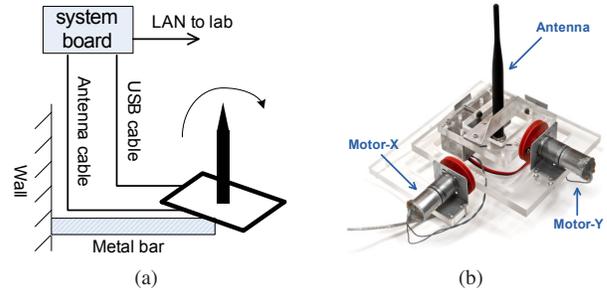


Fig. 2. The diagram on the left shows how a Dyntenna node is deployed. The moving base of a Dyntenna node is shown on the right. The two motors control the rotation along X-axis and Y-axis respectively.

To simplify the implementation, we chose a step size of 9° for each motor, resulting in 11 steps along each axis and 121 total possible antenna orientations. The selection of the step size was based on three considerations: (i) it should be large enough so that changes in the RSSI can be observed between each step; (ii) it should be small enough so that the collected RSSI samples at different steps are able to capture the true variations in the RSSI; and (iii) it should be an odd number so that the default upright orientation of antenna is exactly in the middle of all the steps.

The antenna adjustment algorithm was implemented as a Click [6] module that operates between the MAC and routing layers. This design allows our algorithm to access low-level information such as RSSI, while remaining transparent and compatible with any routing algorithm.

III. MEASUREMENT STUDY

To understand the characteristics of a 3D mesh network with Dyntenna nodes, we conducted an extensive measurement study to answer the following questions: (i) how does the RSSI between two nodes change with antenna orientation? (ii) how does the packet delivery ratio (PDR) change with RSSI and antenna orientation? (iii) how does the relationship between RSSI and antenna orientation change over time? In this section, we present the findings from our measurement study.

A. RSSI Maps

To understand how the antenna orientation affects the signal strength of the reception between the nodes, we ran a systematic experiment on our testbed to measure the RSSI of each link between the Dyntenna nodes and their neighbors, at different antenna orientations.

The experiment is conducted as follows: we picked a Dyntenna node, moved its antenna to each of its 121 possible orientations, and sampled the RSSI readings from the neighbors within range. While this Dyntenna node's antenna is moving, the antenna for all its neighbors are kept in the default vertically-upright orientation. Once the node has finished sampling at all orientations, its antenna is reset to the default orientation and the process is repeated with the next Dyntenna node until samples from all 5 Dyntenna nodes are collected.

For each link (i.e., each Dyntenna-neighbor pair) l , the measured values were recorded in a 11×11 matrix R_l . Each element (i, j) in R_l represents the average RSSI value measured at the antenna orientation (i, j) at 6 Mbps for a duration of 10 s. The packet sending rate is 10 packets per second. We refer to the matrix R_l as the *RSSI map* of the link l . Multiple sets of RSSI maps were measured at different times and we collected 361 sets of RSSI maps, where each set contained results from a single run of the experiment, over a period of one month. In total, we collected 3,487 RSSI maps.

We classified the RSSI maps into three broad categories depending on the distribution of the RSSI values over the space of possible antenna orientations. We noticed that whenever the RSSI of a link was above 9 dB at 6 Mbps, the connectivity is almost guaranteed to be good (see Section III-B). We thus use 9 dB as the threshold to help us categorize the RSSI maps. The categories are as follows:

- *Category A*: RSSI values are all below 9 dB. Links with RSSI maps in this category are deemed unusable. Of the 3,487 matrices collected, 1,224 (35%) fall into this first category.
- *Category B*: RSSI values are all at least 9 dB. Links with RSSI maps in this category are good links for all antenna orientations. 1,029 (30%) matrices fall into this category.
- *Category C*: Some RSSI values in the matrix are greater than or equal to 9 dB, but others are below 9 dB. The remaining 1,234 (35%) matrices fall into this category.

In Fig. 3, we present some sample RSSI maps for links from each of the three categories. In Fig. 3(d), we see an RSSI map with multiple local maxima. Since 35% of the RSSI maps are in Category C, it is likely that the antenna orientation can significantly affect the connectivity of the links in many cases and that *it is indeed necessary in a 3D mesh network to adjust the antenna to obtain good RSSI and connectivity*.

We found that some 80% of the peak RSSI values in our RSSI maps were at least a Chebyshev distance¹ of 3 steps or more away from the center position. This suggests that for the majority of the links in our testbed, the default vertically-upright orientation is not optimal. Another important observation from these measurements is that, as the antenna orientation changes, the RSSI readings vary smoothly. This observation suggests that we can estimate an RSSI map by probing a small number of positions and *interpolating* among them.

B. Relationship between RSSI and PDR

To understand the relationship between RSSI and PDR, we conducted another measurement study on our testbed, where each Dyntenna node moved its antenna to each of the 121 orientations and sent data to each of its neighbors at various

¹The Chebyshev distance between two points (x_1, y_1) and (x_2, y_2) is defined as $\max\{|x_1 - x_2|, |y_1 - y_2|\}$, which corresponds to the number of steps our antenna needs to take to move from one orientation to another.

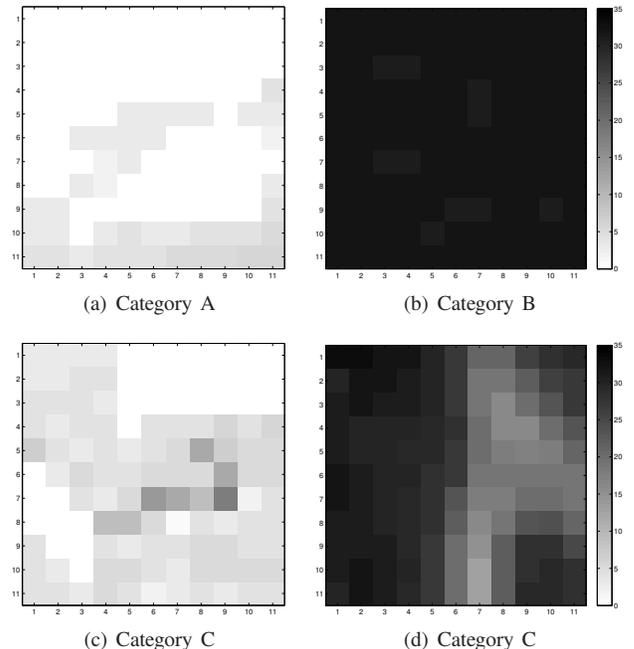


Fig. 3. RSSI maps in different categories. A cell with darker color indicates higher RSSI value. Two RSSI maps for Category C are shown, (c) has a unique global maximum and (d) has multiple local maxima.

data rates. Fig. 4 shows how PDR changes with RSSI on two illustrative links. Each data point in each figure shows the RSSI value and the corresponding PDR for a measurement done for a given link, at one antenna orientation and at the specified link data rate. We observed that there is a sharp increase in PDR values over a small window of RSSI values in all cases. This suggests that a sharp RSSI threshold exists, above which the link becomes reliable. Furthermore, the threshold increases as the link data rate increases, and the threshold for one link can be different from that of a different link even at the same data rate (e.g., at 12 Mbps, the threshold is about 6 dB in Fig. 4(a) and is about 10 dB in Fig. 4(b)). We use this threshold, in combination with an RSSI map, to determine a good antenna orientation in Section IV.

While it has generally been reported in the literature that the RSSI/PDR curve has a much gentler slope [2, 21], the curves in previous work were typically *aggregated* RSSI/PDR curves with values taken over multiple links, each likely with a different threshold. Reis et al. also plotted RSSI/PDR curves by varying the transmission power for a link and observed a sharp threshold [15]. With Dyntenna nodes, we achieved a similar effect by varying the antenna orientation and obtained 121 RSSI/PDR samples for each link.

C. Temporal Variations in RSSI

To understand how RSSI values vary over time, we probed the RSSI values at all 121 orientations continuously for three of the Dyntenna nodes (nodes 8, 9, and 26). At each orientation, a Dyntenna node sends packets to a neighbor at 10 packets per

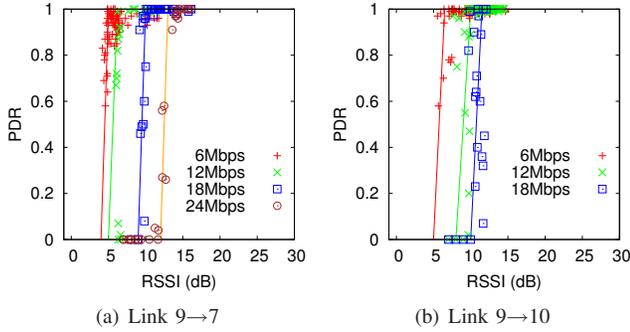


Fig. 4. PDR/RSSI curves for two different links.

second for 10 s. Thus, we can probe all 121 orientations on a single Dytenna node in about 25 minutes, including some additional time to adjust the antenna from one orientation to another. Like in Section III-A, each element in an RSSI map is the average RSSI value over a 10 s duration for an orientation. We recorded the RSSI map for each link every 25 minutes over a period of 25 hours. This measurement experiment therefore yields a sequence of RSSI maps over time. We denote R_l^t as the RSSI map for link l measured during time slot t .

To understand the changes of RSSI over time, we computed the difference matrix $\Delta R_l^t = R_l^t - R_l^{t-1}$. We observe that over 90% of elements in ΔR_l^t have a value of 0 or 1 dB: 49%, 60%, and 73% of the RSSI readings do not change from one slot to the next for links incident to nodes 8, 9, and 26, respectively; 90%, 96%, and 99% of the RSSI readings change by at most 1 dB for links incident to nodes 8, 9, and 26, respectively. These measurements suggest that most of the time, the RSSI values typically do not change from one 25-minute slot to the next.

Next, we analyzed the maximum absolute change in the RSSI readings over time and plot the elements in ΔR_l^t with the largest absolute value (i.e., largest change) versus time in Fig. 5 for three links: 9→7, 26→6, and 9→19. We selected these three representative links to illustrate the different link characteristics we observed from our data. Similarly, we plot the Chebyshev distance between the orientation with the peak RSSI at time 0 and time t , for the same three links in Fig. 6.

Link 9→7 showed significant variations in RSSI over time, with maximum changes within ± 6 dB throughout the experiment. The antenna orientation with the highest RSSI (or the peak), however, did not vary much (note that the largest change in RSSI may not occur at the peak RSSI orientation). This result is ideal since it implies that once we find a good antenna orientation, we can leave the antenna in the same orientation without significant change in performance. Unfortunately, out of the 15 links measured, only three fall in this category.

Link 26→6 showed less RSSI variation over time, but had a sudden change in RSSI during the 10th to 12th hours. There are two plausible reasons for such sudden changes. The first is due to the weather conditions, where rain would affect the reliability of a link, causing a drastic shift in the peak RSSI

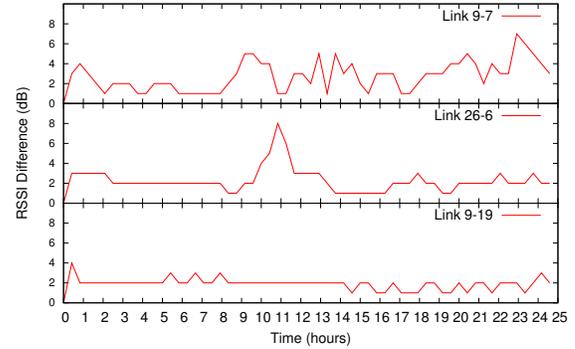


Fig. 5. Maximum RSSI differences between two consecutive 25-min time slots.

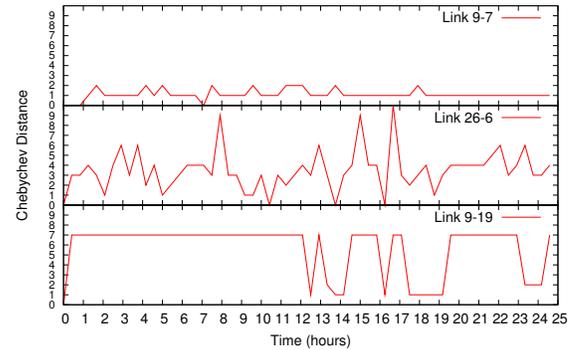


Fig. 6. Chebyshev distance between current and initial peak RSSI positions.

orientation. The second is node churn, where a node that was previously off was switched on, or vice versa.

For links 26→6 and 9→19, the antenna orientation with the peak RSSI changes significantly over time. Such change is observed even for links such as 9→19 that did not show significant changes in RSSI values. This result is due to another characteristic of our testbed, where some links have multiple RSSI peaks in their RSSI map as illustrated in Fig. 3(d). As a result, minor variations in the RSSI can cause the maximum to oscillate between such peak orientations, leading to the phenomenon observed for links 26→6 and 9→19.

In summary, the optimal orientation for an antenna is likely to vary over time and there is hence a need to periodically adjust the antenna. Fortunately, the changes also happen at a time scale slow enough that despite the fact that it takes several minutes to adjust the antenna orientation, it is possible to maintain good link quality by adjusting the antenna periodically.

IV. ANTENNA ADJUSTMENT ALGORITHM

Our antenna adjustment algorithm aims to orient an antenna to improve the throughput of a node to its neighbors, considering link asymmetry, short-term changes to traffic patterns, and long-term temporal changes in RSSI.

The high-level idea of our algorithm is as follows. First, a Dytenna node m probes a set of *anchor orientations* and measures the RSSI values at these orientations. These initial

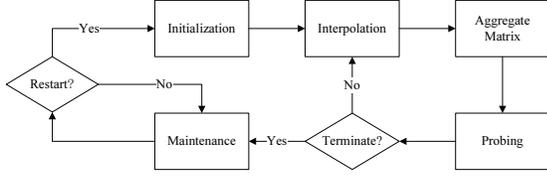


Fig. 7. Overview of algorithm for antenna adjustment.



Fig. 8. Possible initial anchor configurations. Left: 5-anchor constellation. Right: 9-anchor constellation.

RSSI readings are used to interpolate a set of estimated RSSI maps, based on which the optimal orientation for the antenna is estimated, and we adjust the antenna accordingly. Once the antenna is in the new orientation, more RSSI readings are taken and our estimates of the RSSI maps are updated via interpolation with the new data point. The process is repeated until the antenna converges to local maximum. The node continuously monitors the RSSI of its neighbors and reverts to adjustment mode, if a significant change in the traffic pattern or RSSI reading is observed. An overview of the Dyntenna adjustment algorithm is shown in Fig. 7.

A. Initialization

The algorithm first initializes the RSSI maps by probing a 5-point constellation of anchor orientations (see Fig. 8). At each orientation, a Dyntenna node m measures its RSSI readings to each neighbor, and simultaneously, each neighbor measures the RSSI of m 's transmissions and communicates this RSSI value to m . Dyntenna node m uses these readings to initialize two 11×11 RSSI maps per neighbor, for the incoming and the outgoing links, respectively.

We also explored using a 9-point constellation (shown in Fig. 8). To understand the difference between these two initial constellations, we simulated our algorithm in MATLAB, using our RSSI map data. We found that the algorithm converges to the optimal orientation more quickly starting with the 5-point constellation rather than with the 9-point constellation. Thus, our final algorithm uses the 5-point constellation.

B. Probing

When probing each orientation, m stays at the orientation for a *probe interval* of 10 s, during which m sends 1,500-byte unicast packets with RTS/CTS enabled to each neighbor at the rate of 10 packets per second. Each neighbor also sends similar 1,500-byte unicast packets to the Dyntenna node at the same rate. In other words, m sends 100 probe packets to each neighbor and also receives 100 probe packets in return. In

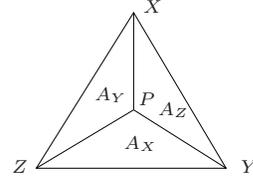


Fig. 9. Using Barycentric Coordinates for interpolation.

addition to RSSI readings, we also estimate the PDR from the number of probe packets received. Probe packets are sent at 6 Mbps at the anchor orientations. Subsequently, probe packets are sent at the same link data rate as the actual traffic on each link.

RSSI Threshold. As described in Section III-B, we observed that there exists a sharp RSSI threshold for each link, above which the PDR for the corresponding link is close to 1. Conversely, if the RSSI value is below this threshold, the PDR falls rapidly to zero. We estimate the RSSI threshold θ_l for each (directional) link l by comparing the measured RSSI r_l to the measured PDR p_l . θ_l is initialized to zero and is continuously updated to r_l under either of the following conditions:

- If $r_l < 0.9\theta_l$ and $p_l > 0.5$; or
- If $r_l > 1.1\theta_l$ and $p_l < 0.5$

Since measurement errors in both r_l and p_l are likely, the factors 0.9 and 1.1 are introduced to allow a 10% allowance before adjusting the threshold, adding a small amount of *hysteresis* to the system and making the algorithm more stable.

C. Delaunay-triangulation-based Linear Interpolation

In this section, we describe the interpolation algorithm used to estimate the RSSI maps using the readings from only a small number of probed orientations. In our work, we explored several interpolation methods and found that a Delaunay-triangulation-based linear interpolation is sufficient to achieve a good balance between accuracy and computational complexity.

An RSSI map R_l typically consists of only a small number of probed orientations. To estimate the RSSI values of the unprobed orientations, we first decompose the space into a set of Delaunay triangulations [19] based on the probed orientations. Since we probe all four corners during the initialization phase, each unprobed orientation is guaranteed to lie within a triangle with a probed orientation at each of the three corners. The RSSI value of an unprobed orientation is then interpolated using Barycentric Coordinates as illustrated in Fig 9, i.e., the interpolated value v_P for an unprobed orientation P would be

$$v_P = v_X \frac{A_X}{A_{XYZ}} + v_Y \frac{A_Y}{A_{XYZ}} + v_Z \frac{A_Z}{A_{XYZ}}, \quad (1)$$

where v_i is the RSSI value at point i , A_X , A_Y and A_Z are the areas of triangles PYZ , PXZ and PXY , respectively, and A_{XYZ} is the area of triangle XYZ .

D. Determination of Next Probe Position

After interpolation, a Dytenna node m will have two RSSI maps for each neighbor, containing the estimated transmission and reception RSSI values respectively at all antenna orientations. From these RSSI maps (for all the neighbors), we want to compute the next best orientation to probe. The best orientation for an antenna is necessarily a compromise between the various optimal orientations with respect to each neighbor.

Considerations. In orienting the antenna, there are several considerations. First, we want to achieve good-quality links to each of the neighbors. Second, we want to avoid disconnecting the network by ensuring that none of the links to any of the neighbors will become unusable. Finally, we want to adapt to the traffic pattern, e.g., improve the link to the only neighbor the node is receiving from or transmitting to.

Aggregate Matrix. The selection of the optimal orientation for a Dytenna node m is based on calculation of an *aggregate matrix*, A_m , which gives the utility value if we move the antenna of m to an orientation (i, j) . In other words, A_m is like a *utility function*. Let $L(m)$ be the set of all links incident on m (including incoming and outgoing links). We define the aggregate matrix for a node m as:

$$A_m(i, j) = \begin{cases} \sum_{l \in L(m)} w_l R_l(i, j), & \text{if } \forall l \in L(m), R_l(i, j) > \theta_l \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

where w_l is a coefficient that is a function of the traffic load for link l , and it provides us the control knob to regulate the links with different traffic loads (e.g., larger w_l for link l with higher traffic). For simplicity, we set w_l to 1, and it remains as future work to study the impact of different policies for w_l .

The intuition behind the aggregate matrix is as follows: we set $A_m(i, j)$ to 0 if there exists a non-reliable connection either to or from one of m 's neighbors (RSSI below threshold) when we orient the antenna to (i, j) , to avoid orientations that would break a link even if it would improve the connectivity of other links. Among all the other antenna orientations that do not cause a link to be disconnected, we favor the one with the largest total weighted RSSI.

Next Probe Position. After A_m is computed, the next orientation P to be probed is the orientation that has the largest value in A_m among all the *unprobed orientations*. Once the antenna moves to P , the node m probes its neighbors to update the corresponding RSSI maps as well as A_m . Let P_{max} denote the probed orientation with the highest value in A_m found thus far. We stop probing when we cannot find an orientation with a value in A_m that is higher than the value of P_{max} in the last K probes. Clearly, if K is too small, the algorithm might prematurely stop at a local maximum; if K is too large, we might waste time probing non-optimal orientations. Based on offline MATLAB simulation using available RSSI maps, we found that, with $K = 3$, more than half of the

simulated cases would find the optimal orientation while only requiring an average of 10 probes. Thus, we set $K = 3$ in our algorithm. Once we stop probing, the antenna moves to P_{max} and Dytenna goes into the maintenance phase.

E. Maintenance Phase

After a Dytenna node goes into the maintenance phase, the antenna remains stationary, but we continue to monitor the RSSI readings to its neighbors in the background. If we detect a 3 dB or larger change in the RSSI values for the received or transmitted packets for any neighbor, we reset all current estimates and start the probing process by probing the 5-point initialization constellation again. The value 3 dB is chosen because our measurement study (see Section III-C) and existing work [21] suggest that short-term fluctuations of up to 3 dB are common.

In addition, we see in Section III-C that RSSI tends to change in the order of hours. So, even if we do not detect any significant changes in the RSSI within an hour, we still have to probe the 5-point initialization constellation every hour. If the probed RSSI values are unchanged, we will stay in maintenance mode at orientation P_{max} . If we find that the RSSI values for the 5-point initialization constellation are different, we will continue probing until the system finds a new P_{max} as described in Section IV-D.

F. Coordinating Between Dytenna Nodes

To guarantee that at most one Dytenna node is adjusting its antenna at one time, we have implemented a lock mechanism. Before the initialization phase, a Dytenna node will send a move request to each of its neighboring Dytenna nodes, and it can start adjusting its antenna only if all its neighboring Dytenna nodes respond with an accept message. Otherwise, it remains stationary and keeps sending move requests periodically until all its Dytenna neighbors accept its request to move, or if a reject message is received. A Dytenna node will reject a move request from a neighbor if (i) it is itself in the process of requesting to move or (ii) it is already in the process of adjusting its antenna. It remains as future work to investigate how multiple Dytenna nodes should adjust their antennas simultaneously.

V. PERFORMANCE EVALUATION

In this section, we present the evaluation of our antenna adjustment algorithm on our 20-node 802.11 wireless mesh testbed. Our goal is to understand to what extent a Dytenna node can improve the throughput of the system.

We set up our experiments in the following way. We first run Srcr [3], which is the default routing algorithm for MIT Roofnet [16]. Once the routing tables are stable, we retrieve the routes from all the nodes. We use these routes to conduct three sets of experiments under different settings: (i) single-hop single-flow, (ii) multi-hop single-flow, and (iii) single-hop multi-flow. To measure the throughput between a pair of nodes,

we send UDP traffic using Iperf from one node to another at a sending rate that saturates the link, and record the average throughput for 15 seconds.

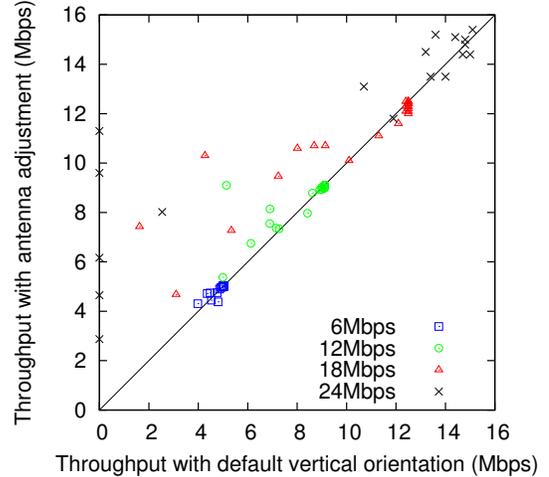
A. Single-Hop Single-Flow

We first evaluate how Dyntenna improves the throughput of *single-hop* flows. We choose 26 pairs of connected neighbors from the Src routes (initialized with Src running at 6 Mbps). For each one-hop (directed) flow formed by a Dyntenna node and its neighbor, we set the antenna to its default vertical orientation and measure the throughput using different data rates: 6 Mbps, 12 Mbps, 18 Mbps, and 24 Mbps. Note however that some links were not able to transmit at the higher data rates. We then repeat these measurements after the antenna moves to a stable orientation as determined by the antenna adjustment algorithm. Let T_r^{up} and T_r^{dyn} denote the throughput of a flow at link data rate r , with the antenna at the *default vertically-upright orientation* and the *orientation from our antenna adjustment algorithm*, respectively. To avoid interference from other nodes, only the pair of nodes that we are measuring are active during the measurement. Since the antenna adjustment algorithm orients an antenna in a way as to maximize the RSSI for links, the comparison of T_r^{up} and T_r^{dyn} for single-hop flows therefore provides us an understanding of the upper bound in the improvement that we can possibly achieve from using Dyntenna nodes.

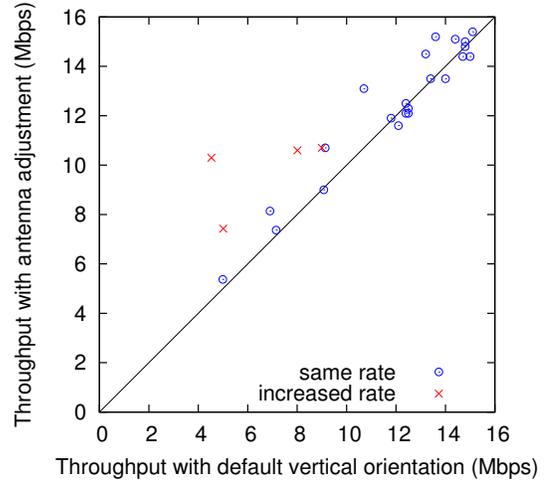
To differentiate the throughput changes due to Dyntenna and due to environmental factors, we separately quantified the baseline throughput variation due to environmental factors by measuring the throughput of a large set of randomly chosen links each 10 times. We found that the average maximum variation was 7% for our testbed. In other words, if the difference between T_r^{up} and T_r^{dyn} is less than 7%, we cannot be sure that the difference is not due only to environmental variations. Henceforth, we say that a throughput change due to Dyntenna is *significant* only if the difference in throughput is at least 7%.

Fig. 10(a) shows the measurement results. 67 (73%) of the total 92 cases saw very little change in throughput, and significant changes ($> 7\%$) were observed for the remaining 25 (27%) cases. Among these 25 cases, only one case suffered a small throughput reduction of 9%. These results show that our antenna adjustment algorithm does not typically have any negative impact on one-hop links. Among the remaining 24 cases, five cases were found on the vertical axis in Fig. 10(a). They correspond to the situation where there was originally no connectivity between the two nodes with the Dyntenna antenna in the default vertically-upright orientation, and our antenna adjustment algorithm was able to re-orient the antenna to allow the two nodes to communicate. The cumulative distribution of all the single-hop cases with significant throughput improvement is shown in Fig. 11, and the median improvement is 31%.

To investigate the potential throughput gain for an *ideal rate adaptation algorithm*, which maximizes throughput by



(a) Throughput for individual data rate.



(b) Maximum throughput.

Fig. 10. Throughput improvements due to Dyntenna (1-hop).

selecting the optimal data rate, we plot the maximum achievable throughput for each link in Fig. 10(b), i.e., we plot $\max_r\{T_r^{dyn}\}$ versus $\max_r\{T_r^{up}\}$. We found that 10 (38%) out of the 26 links could achieve a significant throughput improvement with antenna adjustment. Among them, the average increase in throughput was 25% and in the best case, the throughput can be improved by up to 127%. Four of the links were able to achieve a higher throughput by running at a higher data rate. This observation suggests that integrating antenna adjustment into existing rate adaption algorithms could further boost link throughput.

B. Multi-Hop Single-Flow

Next, we evaluate the performance of Dyntenna for *multi-hop flows*. We selected a set of 2-hop and 3-hop flows that included exactly one Dyntenna node according to the observed Src routes for different data rates r . The Dyntenna node could be the sender, the receiver, or a relay node. For each selected

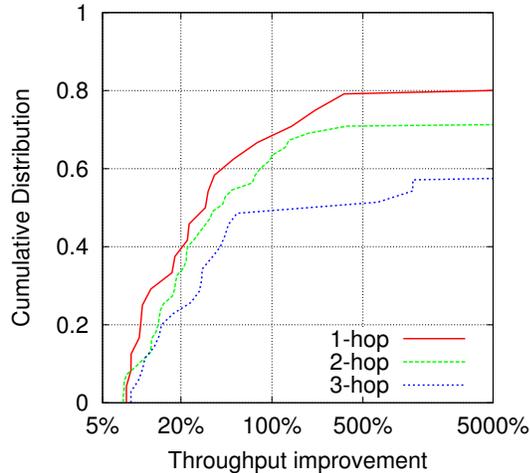


Fig. 11. Cumulative distribution of the cases with significant throughput improvement, for different hop counts.

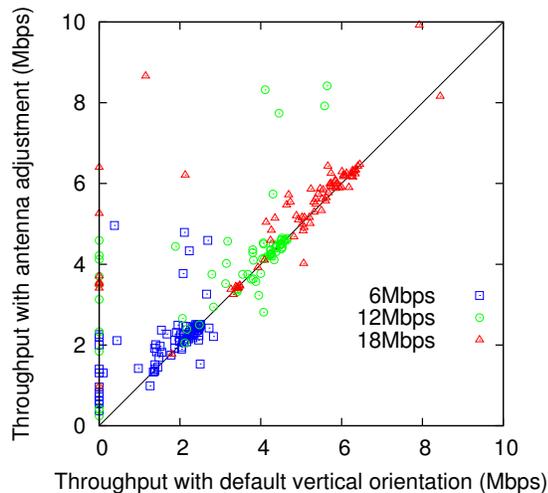


Fig. 12. Throughput for individual data rate (multi-hop).

flow, we measured T_r^{up} and T_r^{dyn} as before for the three data rates 6 Mbps, 12 Mbps, and 18 Mbps. We did not use higher data rates since there were few usable multi-hop paths on our testbed at the higher data rates. Similarly, we did not use longer paths since there were few paths with more than 3 hops.

In Fig. 12, We plot T_r^{dyn} against T_r^{up} for different data rates r . For 159 (61%) of the 260 cases, the throughput change was insignificant. 11 cases (4%) suffered an average throughput reduction of 24%. For the remaining 90 cases (35%), 31 cases achieved non-zero throughput only after antenna adjustment. The cumulative distribution of the 55 2-hop and 35 3-hop cases with significant improvement is shown in Fig. 11. The median gains for the 2-hop and 3-hop cases were 39% and 54%, respectively. In summary, for 35% of the multi-hop paths, we can achieve a median throughput improvement of 46%.

When we examined the data, we found that the throughput improvement in the multi-hop scenario is not always due to improved link quality for the links in the route. Route changes occurred in some cases. Among the cases with significant

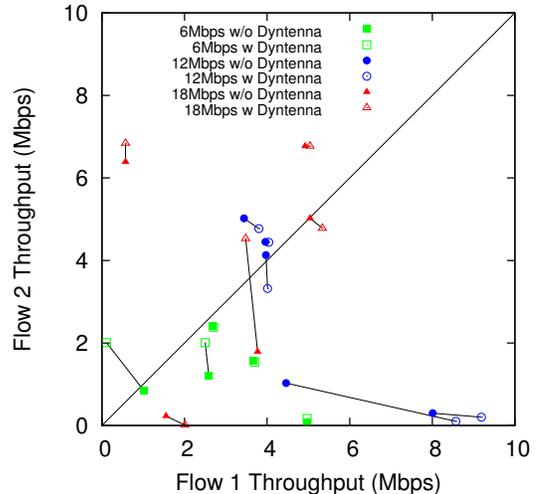


Fig. 13. Single-hop multi-flow performance.

improvement in throughput, we found that the hop count was sometimes reduced from 3 to 2 hops or from 2 hops to 1 hop after the antenna was adjusted. Similarly, there are also a few cases where changing the antenna orientation caused the hop count to increase, leading to a drop in throughput. These results reveal a non-trivial interaction between antenna adjustment and routing algorithm, and suggests that the integration of antenna adjustment with the routing algorithm deserves further investigation, i.e., there is a need for antenna adjustment to be *route-aware* if we want to improve end-to-end throughput.

C. Single-Hop Multi-flow

Finally, we investigate the effect of antenna adjustment on overall throughput and fairness between flows. To isolate the effect of hidden-node collision and focus on the effect of antenna adjustment, we only consider the scenario with all three nodes in the *same collision domain*. In the experiment, two nodes sent data to a single Dyntenna node simultaneously. We measured the throughput of both flows at the link data rates 6 Mbps, 12 Mbps, and 18 Mbps, without and with adjusting the antenna of the Dyntenna node. In Fig. 13, we plot the throughput of the two flows against each other for the various experiments, and we connect the two data points corresponding to the same link to indicate the change in the throughput caused by antenna adjustment.

The plot can be interpreted in two ways. First, The line $x = y$ represents the fairness line with the two flows having equal throughput. If a point moves towards the fairness line, then the resulting allocation of throughput is more fair. Second, each point falls on a line $x + y = K$ (not shown in the figure), where K is the total throughput. As K increases, the line moves towards to the top-right corner of the figure. We can see in our results that using Dyntenna either improves the fairness of the flows, or increases the total throughput. Both of these are desirable outcomes of antenna adjustment.

D. Convergence Time

We measured the time it took for our algorithm to converge to a peak orientation and become stable on our testbed. We found that the minimum convergence time is about 90 seconds, corresponding to 9 probes (5 at the anchor points, 1 at the optimal position, and an additional $K = 3$ probes to confirm). The average convergence time is 104.8 seconds, corresponding to 10 probes, which is consistent with our MATLAB simulation results. In other words, we need only to probe five additional orientations on average beyond the anchor orientations to converge, i.e., our algorithm is very efficient and converges after probing less than 10% of all 121 possible orientations.

VI. RELATED WORK

There is a large body of work investigating the relationship between RSSI and PDR in practical 802.11 wireless networks [2, 8, 14, 15, 21]. While Aguayo et al. suggested that there was only a weak relationship between packet loss rate and RSSI [2], Raman et al. found that, without severe external interference, the correlation between PDR and RSSI was strong on outdoor 802.11bg links [14]. Like us, Reis et al. also studied the correlation between PDR and RSSI for *individual links* and found that the correlation was quite significant and that there exists a rather sharp threshold [15].

There are a small number of measurement studies on the impact of dynamic (or steerable) antennas in wireless networks [5, 9, 20]. Like what we found on our testbed, they showed that (i) RSSI could opportunistically increase as the antenna orientation changes [9], and (ii) the change of RSSI is smooth [5, 20].

Previous work on steerable antenna typically used either directional antennas [17] or phased array antennas [10, 11, 12, 13, 22], and focussed on increasing spatial diversity. Like us, many of them also use RSSI as one of the key metrics for making adjustments. There are also several significant differences: (i) most are focused on infrastructure-based networks (i.e., access points) [10, 11, 12, 13, 17], instead of 3D wireless mesh networks; (ii) most of them used expensive hardware like phased array antennas, whereas we work with low-cost hardware; and (iii) some assumed link symmetry [12], while we do not make such an assumption because we have observed severely asymmetric links on our testbed.

VII. CONCLUSION

We built and evaluated a 3D 802.11 wireless mesh testbed with Dyntenna nodes and showed that adjusting the orientation of the omnidirectional antenna can improve throughput, when the default vertically-upright orientation does not give the best RSSI. We showed that it is possible to efficiently move the antenna to a good orientation using RSSI interpolation by probing less than 10% of all possible orientations. We believe that our work lays the foundation for a new class of 3D mesh networks with steerable omnidirectional antenna. Possible future work

includes the coordination between Dyntenna nodes, a more detailed investigation of the traffic coefficient w_l used in the computation of the next probe position, the integration of the antenna adjustment algorithm with routing and rate adaptation, and also the application to multi-antenna networks such as 802.11n.

REFERENCES

- [1] Antenna RD-2458-5-OTDR. Website. <http://www.lairdtech.com/>.
- [2] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris. Link-level Measurements from an 802.11b Mesh Network. In *Proceedings of SIGCOMM '04*, Aug. 2004.
- [3] J. Bicket, D. Aguayo, S. Biswas, and R. Morris. Architecture and Evaluation of an Unplanned 802.11b Mesh Network. In *Proceedings of MobiCom '05*, Aug. 2005.
- [4] V. Brik, S. Rayanchu, S. Saha, S. Sen, V. Shrivastava, and S. Banerjee. A Measurement Study of a Commercial-grade Urban WiFi Mesh. In *Proceedings of IMC '08*, Oct. 2008.
- [5] M. Buettner, E. Anderson, G. Yee, D. Saha, A. Sheth, D. Sicker, and D. Grunwald. A Phased Array Antenna Testbed for Evaluating Directionality in Wireless Networks. In *Proceedings of MobiEval '07*, Jun. 2007.
- [6] Click. The click modular router. <http://read.cs.ucla.edu/click/>.
- [7] F. Comtech. Phocus antenna array system. Website, 2009. <http://www.fidelity-comtech.com/>.
- [8] D. Halperin, W. Hu, A. Sheth, and D. Wetherall. Predictable 802.11 Packet Delivery from Wireless Channel Measurements. In *Proceedings of SIGCOMM '10*, Aug. 2010.
- [9] S. Lakshmanan, K. Sundaresan, S. Rangarajan, and R. Sivakumar. Practical Beamforming based on RSSI Measurements using Off-the-shelf Wireless Clients. In *Proceedings of IMC '09*, Nov. 2009.
- [10] X. Liu, A. Sheth, M. Kaminsky, K. Papagiannaki, S. Seshan, and P. Steenkiste. DIRC: Increasing Indoor Wireless Capacity Using Directional Antennas. In *Proceedings of SIGCOMM '09*, Aug. 2009.
- [11] X. Liu, A. Sheth, M. Kaminsky, K. Papagiannaki, S. Seshan, and P. Steenkiste. Pushing the Envelope of Indoor Wireless Spatial Reuse using Directional Access Points and Clients. In *Proceedings of MobiCom '10*, Sep. 2010.
- [12] V. Navda, A. P. Subramanian, K. Dhanasekaran, A. Timm-Giel, and S. R. Das. MobiSteer: Using Steerable Beam Directional Antenna for Vehicular Network Access. In *Proceedings of MobiSys '07*, Jun. 2007.
- [13] K. Ramachandran, R. Kokku, K. Sundaresan, M. Gruteser, and S. Rangarajan. R2D2: Regulating Beam Shape and Rate as Directionality Meets Diversity. In *Proceedings of MobiSys '09*, Jun. 2009.
- [14] B. Raman, K. Chebrolu, D. Gokhale, and S. Sen. On the Feasibility of the Link Abstraction in Wireless Mesh Networks. *IEEE/ACM Transactions on Networking*, 17(2):528–541, Apr. 2009.
- [15] C. Reis, R. Mahajan, M. Rodrig, D. Wetherall, and J. Zahorjan. Measurement-Based Models of Delivery and Interference in Static Wireless Networks. In *Proceedings of SIGCOMM '06*, Aug. 2006.
- [16] Roofnet. The MIT Roofnet, 2009. <http://pdos.csail.mit.edu/roofnet/>.
- [17] A. A. Sani, L. Zhong, and A. Sabharwal. Directional Antenna Diversity for Mobile Devices: Characterizations and Solutions. In *Proceedings of MobiCom '10*, Sep. 2010.
- [18] J. Shi, O. Gurewitz, V. Mancuso, J. Camp, and E. W. Knightly. Measurement and Modeling of the Origins of Starvation in Congestion Controlled Mesh Networks. In *Proceedings of INFOCOM '08*, Apr. 2008.
- [19] D. Sinclair. S-hull: A fast sweep-hull routine for Delaunay triangulation. <http://www.s-hull.org>.
- [20] A. P. Subramanian, V. Navda, P. Deshpande, and S. R. Das. A Measurement Study of Inter-Vehicular Communication Using Steerable Beam Directional Antenna. In *Proceedings of VANET '08*, Sep. 2008.
- [21] A. Vlavianos, L. K. Law, I. Broustis, S. V. Krishnamurthy, and M. Faloutsos. Assessing Link Quality in IEEE 802.11 Wireless Networks: Which is the Right Metric? In *Proceedings of PIMRC '08*, Sep. 2008.
- [22] H. Yu, L. Zhong, A. Sabharwal, and D. Kao. Beamforming on Mobile Devices: A First Study. In *Proceedings of MobiCom '11*, Sep. 2011.