Interference-Aware Transmission Power Control for Dense Wireless Networks

Xi Liu Srinivasan Seshan Peter Steenkiste

Computer Science Department, Carnegie Mellon University

***Abstract*—In this paper, we describe a protocol that manages the transmission power of 802.11 devices to maximize the performance of nodes within an area with a dense concentration of 802.11 networks. We show that it is possible to calculate the *ratio* between the transmit power of different nodes that maximizes overall network capacity. We present a protocol that implements a distributed version of our transmission power setting algorithm. Our protocol also tunes carrier sense thresholds to ensure that simultaneous transmissions occur when possible. Our evaluation of this protocol using the OPNET simulator shows that it improves network capacity by 22% to 87% over only adjusting carrier sense and by 2% to 67% over using the minimum possible transmit power level.**

I. Introduction

The availability of spectrum resources has not kept pace with wireless network popularity. As a result, data transfer performance is often limited by the number of devices interfering on the same frequency channel within an area. This problem is exacerbated by the fact that the default transmit power and carrier sense threshold (aka clear channel assesment or CCA threshold) used in 802.11 devices results in inefficient use of the spectrum. In this paper, we present an algorithm and protocol for selecting the transmit power and CCA threshold in 802.11 so as to reduce interference and increase network capacity, while maintaining fairness.

We present our design in two steps. First, we present an iterative greedy algorithm that uses perfect knowledge about the RF environment to determine the transmit power for all transmissions. The basic idea of the algorithm is to adjust transmit power such that it reduces the number of edges in the transmission conflict graph[13]. The algorithm iteratively adjusts power levels to eliminate edges until no more edges can be removed. Despite the fact that the algorithm is greedy, we show in simulation that its performance is close to the upper bound. For a network of 20 APs and 60 clients, the average improvement of network capacity over all nodes using the same power is about 80%. Furthermore, the improvement increases as more nodes join the network.

Second, we incorporate a practical version of the power control algorithm into protocol for 802.11 nodes, which addresses the following challenges: 1) How to collect necessary RF environment in a distributed manner. In the protocol, each source inserts information about current transmission into the packet. Also, every wireless node is put into monitor mode, allowing it to observe ongoing transmissions and obtain local topology information. Each time a node receives a packet, it executes the above algorithm using its locally available information and uses the computed configuration for future transmissions. 2) How to choose the appropriate CCA thresholds. Our protocol adjusts the CCA threshold using a variant of the Echos [15] algorithm that we call Altruistic Echos (AEchos). 3) How to choose the receiver thresholds, etc. Using OPNET simulations, we show that this power management protocol can improve network capacity by 22% to 87% over only adjusting carrier sense and by 2% to 67% over using the minimum power level. In addition, our protocol can help most low throughput links perform better, indicating that the scheme does not sacrifice fairness for performance.

The rest of the paper is organized as follows. Section II provides background information. Section III describes how to configure transmission power in simple two node pair scenarios. In Section IV, we present our greedy iterative algorithm. Section V discusses how to incorporate CCA tuning into the protocol. In Section VI, we present and evaluate the distributed power management protocol. Finally, Section VII reviews related work and Section VIII conclusions.

II. Background

A. System Model

We focus on infrastructure mode wireless networks, where clients use an AP to connect to the Internet. We assume that clients use pre-defined criteria to select an AP, e.g. in residential areas, a client associates with its own AP, or in campus deployments, the closest AP. Since the density of APs and clients can be highly variable, we take the approach of per-link power management, so every client contributes two links to the network, one from itself to the AP, and the other from the AP to itself. However, the bulk of our evaluation and description focuses on traffic from the AP. Note that our results are actually not specific to 802.11 AP based networks and apply to any wireless system in which sources directly transmits to receivers. For example, 802.11 ad hoc networks and bluetooth networks can use our designs without modification. However, our schemes do not maximize aggregate end-to-end performance in a multi-hop routing environment. We leave this optimization for future work. Also in this paper, we do not consider mobility, but do evaluate with small variations in path loss.

B. Concurrent Transmission Model

Researchers have used several different models to determine when concurrent transmissions can occur. We review three models (the physical SINR, protocol and circle models) and their core assumptions.

The physical SINR model [7][13] takes into account received signal strength from the source and all other sources of interference, and noise. It predicts that a packet from source n_i to destination n_j will be decoded if the signal to interference and noise ratio $SINR_{ij}$ is no less than a threshold $SINR_{thrsh}$, i.e.:

$$SINR_{ij} = \frac{RSS_{ij}}{N_a + \sum_{n_k} RSS_{kj}} \geq SINR_{thrsh}$$

where N_a is the thermal noise level, n_k enumerates over all currently transmitting sources, RSS_{ij} is the received signal strength from node n_i at node n_j. The SINR model predicts that uniformly increasing power levels increases system throughput by reducing the effects of thermal noise; however, in interference-dominated networks, the improvement is marginal.

The circle model is a much simpler model that is used implicitly in many papers [10][1][16]. In this model, every source is associated with a transmission range and an interference range. Nodes within the transmission range of a source can decode frames from the source, and nodes in interference range will be prevented from transmitting due to their carrier sensing [3]. Also, the ranges depend on the power level each source uses. The circle model predicts that using the minimum possible power level minimizes interference – exactly the opposite conclusion of using the SINR model!

The right choice of model depends on whether the interface hardware exhibits the capture effect [17] (i.e. the ability to decode one transmission when multiple transmissions collide). In particular, it depends on the behavior that occurs when the interface is receiving a frame and a stronger signal comes. If the interface captures the stronger signal, then the SINR model is more accurate – otherwise the circle model is a better predictor. To determine the behavior of modern 802.11 hardware, we performed an experiment in a wireless emulator [9]. In this experiment, we use two pairs (i.e. four) of laptops. We use *netperf* to create a single non-rate controlled UDP flow on each of the two node pairs, namely F_a and F_b, with transmit power levels P_a and P_b, respectively. The power levels are set manually. Also, we use the emulator to prevent any interference at both sources - thus eliminating the effects of carrier sense and collisions with link-layer ACKs. Figure 1(a) shows throughput achieved by F_a. We keep P_b constant and vary P_a for each curve and vary P_b between curves. The well-spaced and similar-shaped curves, with an exception when thermal noise dominates interference, show that capture effect occurs and that the circle model is not appropriate for this hardware. This is in contrast with recent work [18] that suggests otherwise.

Unfortunately, it is difficult to use the SINR model because each transmission interacts with all other transmissions. The protocol model [13][7] simplifies the SINR model by making three assumptions: 1) interference comes from the strongest source (pair-wise interference assumption) 2) thermal noise can be ignored, and 3) all nodes use the same power level. The first assumption is reasonable as long as we make sure that transmit power levels are high enough. Recent work [5] has shown that the pair-wise assumption usually holds. Also, pairwise assumption is consistent with conflict graph [13], which we use to represent spatial reuse. We show how to relax the third asssumption below. Formally, in the original protocol model, a transmission from source n_i to destination n_j will be successful if $d_{kj} > (1+\Delta) \times d_{ij}$, where d_{ij} denote the distance between node n_i and node n_j, and $\Delta (> 0)$ denotes the guard zone provided by the particular protocol [7]. We modified the model to accomodate heterogenous power levels. In the new variant of protocol model, transmission from source n_i to destination n_j will be successful if $\frac{RSS_{ij}}{\max_{n_k} RSS_{kj}} \geq SINR_{thresh}$ where n_k enumerate over all other sources that are currently transmitting, and RSS_{ij} denotes the received signal strength from node n_i to node n_j.

We have modified OPNET to support the SINR model, and our algorithm design and conflict graph simulation are based on the protocol model variant. We will show that despite the simplifying assumptions, our protocol has better performance.

C. *Conflict Graph*

To concisely represent the interference that is present in a network, researchers have used a conflict graph [13]. Each node in the conflict graph represents a link in the wireless network and there is an undirected edge between two links if they cannot be active at the same time (i.e. either link interferes with the other). Clearly the conflict graph depends on the transmit power used by the nodes. Figure 1 shows an example of wireless networks and its corresponding conflict graph. We construct the conflict graph based only on SINR layer interference, independent of any MAC protocol. As a result, it is sufficient to use undirected edges in the conflict graph: concurrent transmissions should be avoided if both links interfere with each other or if interference happens in only one direction.

Past work uses a variety of optimization metrics, e.g. maximum capacity [13], maxmin fair [2], and minimum mean delay [12]. Our goal is to increase spatial reuse (i.e. allow more concurrent transmissions). Since the lack of an edge in conflict graph is equivalent to enabling concurrent transmissions, our goal is equivalent to minimizing the number of edges in the conflict graph.

III. Power Control

In this section, we analyze optimal power selection for a two link scenario and evaluate how often it improves link throughput. This analysis uses the protocol model and ignores all MAC layer effects.

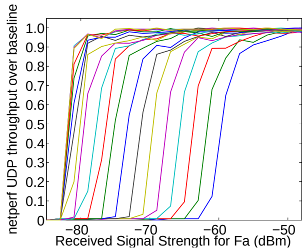
A. *Enabling Concurrent Transmisssions*

Let's consider a pair of transmissions (Figure 2(a)) where AP_1 is transmitting to n_1 and AP_2 is transmitting to n_2. The SINR at receivers n_1 and n_2 are $SINR_1 = P_1 - L_{11} - P_2 + L_{21}$ and $SINR_2 = P_2 - L_{22} - P_1 + L_{12}$, respectively, where P_i is the transmit power level from AP_i to n_i, and L_{ij} is the path loss from AP_i to n_j $(i, j \in \{1, 2\})$[1]. In order to enable concurrent transmission, we need both $SINR_1 \geq SINR_{thrsh}$ and $SINR_2 \geq SINR_{thrsh}$, which implies that

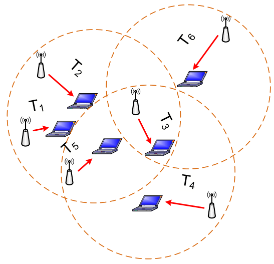
$$L_{12} - L_{22} - SINR_{thrsh} \leq P_1 - P_2 \leq L_{11} - L_{21} + SINR_{thrsh}$$

Note that this range can be empty, i.e. concurrent transmission is impossible regardless of transmit power level. We refer to setting transmit power using

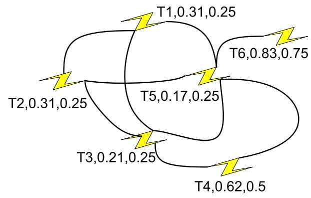
[1]Although metric space is used in some examples and in simulation, we do use this assumption in our algorithm and protocol, i.e. path loss can be arbitrary.



(a) Packet Capture With Interfering Signals



(b) Example of wireless networks



(c) Corresponding conflict graph

Fig. 1. Background Measurements and Examples

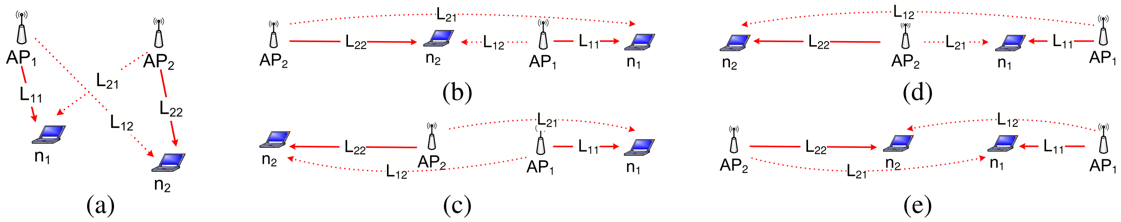


Fig. 2. Topologies for two transmissions, (a) in general, (b)(c)(d)(e) some common scenarios.

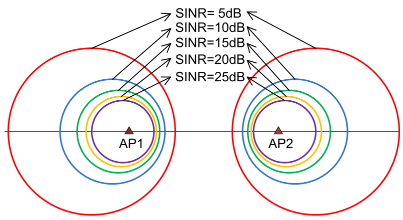
this relationship as OPT.

We now examine what happens if nodes use no power control (i.e., all nodes use the same power level or *equal power*) or nodes use the the minimum power level to reach the receiver (i.e., *minimum power*). With equal power, we have $SINR_1 = L_{21} - L_{11}$, and $SINR_2 = L_{12} - L_{22}$. In Figure 2, equal power performs well in scenario (c) & (d) because $SINR_1 \approx SINR_2$ but poorly in (b) & (e) because $SINR_1 >> SINR_2$. With minimum power, we have $SINR_1 = L_{21} - L_{22}$, and $SINR_1 = L_{12} - L_{11}$. Minimum power performs well in scenario (b) & (e) but poorly in (c) & (d).

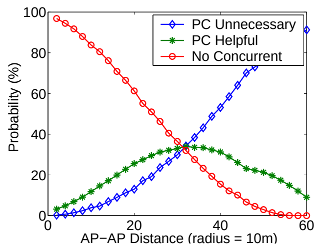
B. How Much Improvement Can We Expect?

Intuitively, power control helps in networks with diverse client-AP distances. Figure 3(a) show a scenario with two APs and two clients. The AP locations are marked while the circles demark areas where clients will observe different SINR, assuming both APs use the same power level, and RF propagation is modelled based on free-space propagation. The middle circle corresponds to SINR=15dB, and the next large (or small) circle corresponds to SINR=5dB (or SINR=20dB), etc. We can identify three cases: 1) **Power control helps** when one client is inside a smaller circle than the middle, the other is inside a larger circle than the middle, and the sum of SINR is no less than 30dB. For example, one client can be in the SINR=10dB ring and the other is in SINR=20dB. 2) **Power control is unnecessary** when both clients are inside SINR=15dB circle. 3) **Concurrent transmission is impossible** in all other settings.

Figure 3(b) shows how often each case occurs for different AP-AP distances. We randomly place the client in a 10m radius disk centered on its AP. We see that when APs are close, concurrent transmission is often impossible. In contrast, when APs are far away, it becomes likely that concurrent transmission is possible. Power control works best when AP-AP distance is about 3 times the radius size and enables concurrent transmission 35% of the time in this setting.



(a) where power control helps



(b) how often power control helps

Fig. 3. For a pair of transmissions in metric space

We also use data collected in Wean Hall at Carnegie Mellon University to estimate how often power control might help in real environments. The data includes the signal strength collected in each room for each AP. Data for a total of 67 rooms and 26 APs were recorded. We assume that there is one client inside each room and it associates with AP with strongest signal. There are totally 769 link pairs that can decode each other's packets, and power control enables 103 (13.4%) link pairs transmit simultaneously. Note that this is a sparse network, i.e. APs are placed carefully as opposed to chaotic deployment [1], and the measurement does not include link pairs that cannot decode each other's signal. Also, 88 (11.4%) link pairs can transmit together by just using equal power, and 578 (75.2%) link pairs can never transmit at the same time.

IV. Power Control Algorithm

We now present a greedy, iterative power control algorithm. It assumes that every node has complete knowledge of the network topology (i.e. path loss between nodes) and current configuration (i.e., power levels sources are currently using). We also present simulation results that show that this greedy approach performs quite close to the possible upper bound.

A. Algorithm

We have shown how to set ratios between two transmissions to allow them to happen at the same time. However, if there are multiple transmissions, the choice made for one other transmission may not be compatible with the choice for another transmission. Thus, we use a greedy algorithm that iteratively allows more concurrent transmissions.

We use the following notations in Algorithm 1. $src(t), dst(t)$ are the source and destination of link t while $P(t)$ is the power level currently used on that link. For any two nodes n, n', $L(n, n')$ is the path loss from n to n'. We assume that each wireless devices has a range of allowable power levels, e.g. typically limited by noise consideration (lowerbound) and power/FCC restrictions (upperbound).

In each iteration in the algorithm, and for each link t, it examines the power level used on all other links (line 6), the topology (line 7-10), and it determines what power level would allow simultaneous transmission with the other links (line 11-12). It then picks the power level that can remove the most edges from the conflict graph (line 18). The new power level will be used if it allows more concurrent transmissions than that in the last iteration (line 19-21). Also, by using this algorithm, the source that needs maximum power level will hit the power limit, and then all other sources will keep an appropriate ratio to that source.

Algorithm 1 Iterative Power Control Algorithm

1. **while** not stable **do**
2.   /* For each link t, use $v[i]$ to determine how often concurrent transmission is possible with power level i */
3.   **for all** t **do**
4.   clear(v)
5.   **for all** $t' \neq t$ **do**
6. $P' \leftarrow P(t')$
7. $L_{11} \leftarrow L(src(t), dst(t))$,
8. $L_{12} \leftarrow L(src(t), dst(t'))$
9. $L_{21} \leftarrow L(src(t'), dst(t))$
10. $L_{22} \leftarrow L(src(t'), dst(t'))$
11. $P_{min} \leftarrow L_{12} - L_{22} - SINR_{thrsh} + P'$
12. $P_{max} \leftarrow L_{11} - L_{21} + SINR_{thrsh} + P'$
13.   /* $[P_{min}, P_{max}]$ is the range of $P(t)$ such that t and t' can transmit together */
14.   **for** $i = P_{min}$ to P_{max} **do**
15. $v[i]$++
16.   **end for**
17.   **end for**
18.   Find the P_m such that $v[P_m]$ is maximum.
    /* $last[t]$ is used to ensure convergence: change power level only when more concurrent transmissions are allowed */
19.   **if** $v[P_m] > last[t]$ **then**
20. $P(t) \leftarrow P_m$
21. $last[t] \leftarrow v[P_m]$
22.   **end if**
23.   **end for**
24. **end while**

B. Evaluation

We use simulation to evaluate the iterative algorithm. The goal of this part of our evaluation is to see how effectively the algorithm maximizes capacity while ignoring any MAC interactions. The simulator first computes power levels using the above algorithm and generates a conflict graph based on the power configuration. Then, it simulates the actual packet transmissions for a workload in which every source is persistently backlogged. The packet transmission simulation is simplified in a number of ways. First, we replace the MAC protocol with a centralized scheduler that for any of the n links that are backlogged and do not interfere with ongoing transmissions, randomly schedules one of them to initiate transmission. In Figure 1(b), the first number on each link shows the expected air time each link would get with this scheduling. As a comparison, the second number is the maxmin fair throughput for that link.

Figure 4 shows the results from a simulation where APs and clients are uniformly distributed in a $100{\times}100$ grid, and each client associates with closest AP. Each data point is the average over 100 random different topologies, and 1000 rounds of scheduling in each topology. Also, although the iterative algorithm always performs better, the variance of improvement is high. This is because in configurations where clients are all associated with one/few APs, there would be no/little improvement at all. We compare equal power, minimum power, and two versions of the iterative algorithm, i.e. per-BSS and per-link. In the per-BSS version of the iterative algorithm, all nodes in a BSS uses the same power level, and in the algorithm, each vertex in conflict graph represents a BSS. Also, for reference, we introduce two baselines. The first is an upper bound, which is the minimum conflict graph, where an edge between a pair of transmissions can be removed in the conflict graph if they can transmit together, assuming no other transmissions. The second is the default power level and CCA threshold used in 802.11 devices. Note that the second baseline differs from equal power in that the centralized scheduler will resolve MAC layer contention in equal power, but in default configuration, MAC layer uses carrier sensing.

Figure 4(a) presents the average link throughput when there are 20 APs. It shows that minimum power and equal power perform similarly, and that the iterative algorithm is very close to the upper bound. (b) shows the improvement over using default 802.11 configurations is linear to the number of APs in the networks. This is because the default configuration used is not scalable for dense networks. (c) shows the tradeoff of per-link versus per-BSS power control. When there are more clients per AP, indicating more diversity within a network, the improvement increases. (d) shows the number of rounds to converge. The results are an upper bound for distributed protocol.

In the second simulation setup, there are 10 APs and 30 clients. We use a clustered node placement model, where APs are uniformly distributed in a $100{\times}100$ grid, and the clients are uniformly distributed in the disk around a randomly chosen AP. Figure 5 shows the improvement over equal power with various radius sizes. It shows that when clients are very close to their

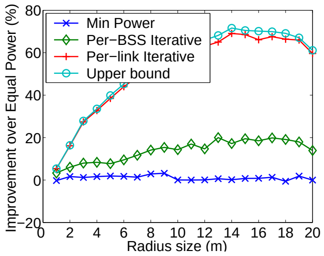


Fig. 5. Conflict graph simulation results for clustered placement

AP, and far away from other APs, there is no need for power control. If all clients are far away from their APs, where there are many cross links, there is high probability that no concurrent transmission is possible. Note that the peak improvement is about 65%, which is slightly less than the improvement in random topology. We have also simulated other node placement models, and observed that the peak improvements are all less than that in random topology. This suggests that the heterogeneity in node placement increases possible power control gains.

V. Interactions with Carrier Sense

In the previous section, we assumed a centralized scheduler is used both to avoid collisions and to enforce a fairness policy. In practice, systems like 802.11 rely on random access and carrier sensing for both functions. While the random access behavior is largely orthogonal to power control, the carrier sense mechanism is not. Past work has noted the importance of CCA tuning when doing power control [12][1] and several CCA tuning techniques exist. In Alpha [6], every source uses a fixed product α (product in watts, or sum in dB) for power level and CCA threshold. This ensures that the symmetry property holds [12]. In Echos [15], every source picks a CCA threshold that allows it to hear all the transmissions that interfere with its current transmission.

Our simulations show that in particular scenarios, Echos can lead to starvation for some transmissions. The cause is that each node in Echos greedily optimizes for its own transmissions. To address this problem, we propose Altruistic Echos (AEchos), where the CCA threshold is set to hear all the transmission that interfere with current transmission or will be interfered by current transmission. Both Echos and AEchos use localized decisions and can easily be incorporated into the power management protocol. Alpha involves a network-wide decision as described in [12], so it is more complex to integrate. In our simulation, we manually set the α.

VI. Protocol

In this section, we present the distributed power management protocol based on the algorithm described in Section IV. We describe topology information collection and packet reception/transmission processing. We also discuss practical issues such as receiver thresholds and variable transmission rates.

A. Data Collection

Using the notation from Figure 2(a), AP_1 needs the following information to determining power level: $L_{11}, L_{12}, L_{21}, L_{22}, P_2$. We collect this by having each source insert the power level used into each transmitted packet. Then, P_2 can be extracted from packet when AP_2 is transmitting. L_{11} can be measured when n_1 is transmitting, L_{12} can be measured when n_2 is transmitting, and L_{21} can be measured by n_2 when AP_2 is transmitting and sent to AP_1. L_{22} can be measured and inserted into a packet by AP_2.

A node may choose to use a low power level and, thus, other transmitters may not be able decode its packets and collect the above information. To avoid this problem, nodes periodically (every 2 seconds) transmit announcements at the maximum power level. We piggyback this announcement on ACK packets, to minimize interference caused by this high power packet.

In summary, when transmitting a packet, every node inserts the power level used, the path loss estimate to the destination, a bit to indicate high power frame, and the path loss estimate to a chosen node. In our current design, a transmitting node choses the node for the additional path lost estimate in a round-robin fashion. It might be desireable to adapt this selection to how the path loss estimates are changing over time. The extra information introduced into each packet is 129 bits, which is less than 1% overhead for a full-sized data packet.

Each node in the system operates in promiscuous/monitor mode and upon receiving a packet, the node updates topology information and configuration. For each active flow, every node keeps several lists: T_{11} to store power level used, CCA threshold used and path loss measured for the destination; T_{12} to store path loss from itself to other nodes; T_{21} to store path loss from the destination to other nodes; and T_{22} to store the power level, and path loss of other flows.

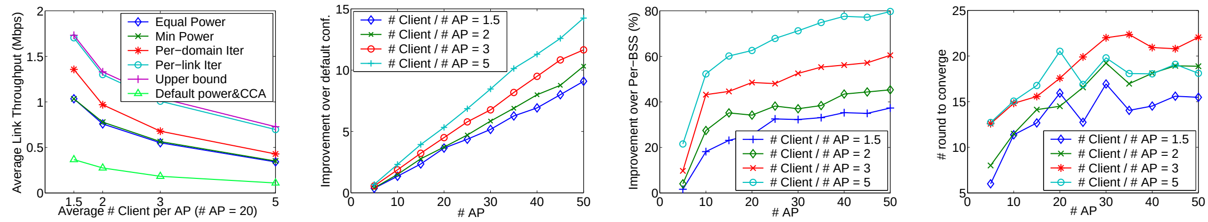
B. Packet Transmission/Reception

When node n_1 is about to transmit a data to destination n_2, n_1 will first search for the power level, CCA threshold, and path loss for n_2. If this information is unavailable, it will use the default power level and CCA threshold, and an invalid value for path loss. This occurs when the client first sends an association request to AP. If n_1 has the data, it will set power level and CCA threshold appropriately. In addition, n_1 will also piggyback power level, path loss, and the picked entry from the T_{12} list onto the frame. The frame is then added into the device queue for transmission.

Upon receiving a packet from n_2, node n_1 first measures the RSS of that packet, and extracts power level, path loss from n_2 to another node, and path loss from n_2 to its destination from the frame. The path loss from n_2 to n_1 is calculated by subtracting RSS from power level used for the packet. Then, n_1 will determine which lists to update. If the packet is destined to n_1, or the packet is from its AP to another client, then it will update T_{11} and T_{21} lists. Otherwise, the node will update T_{12} and T_{22} lists. If there is an update to any list, n_1 will recalculate the configurations for all its destinations.

C. Simulation

We use the OPNET simulator to study the performance of our protocol. In the simulation, we use



(a) Average link throughput (b) Iterative vs. default (c) Per-link vs. Per-BSS (d) Convergence

Fig. 4. Conflict graph simulation result for randomly placed wireless nodes

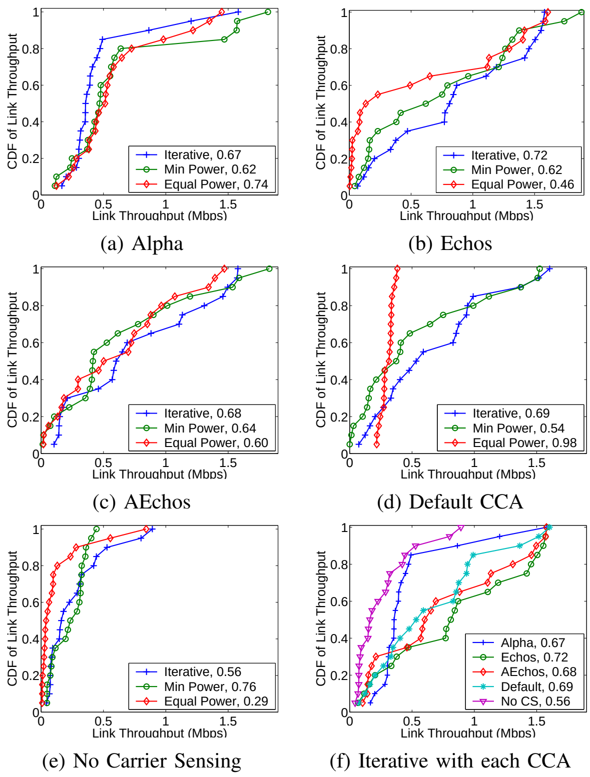
the clustered node place mode, as in Section IV, where APs are uniformly randomly placed inside the grid, and clients are randomly placed inside the disk centered at the AP. In the first simulation, we setup 10 APs and 10 clients within 100m×100m grid, and the radius size ranges from 3m to 24m. In the second setup, we change the number of APs, while keeping the ratio of AP-AP distance to client-AP distance. We assume all APs operate on the same channel and use the same data rate (11Mbps). Traffic is generated using an exponential on-off process with a traffic demand for each node about 2Mbps. The traffic start time is uniformly distributed in the first 20 seconds and we run each experiment for 20 minutes. The results of network capacity presented are averaged over three different random topologies with the same system parameters.

For each configuration, we simulate all combinations of CCA tuning and power control. For power control, the choices are our *iterative* algorithm, the *minimum power* to reach a reciver, and all nodes using *equal power* settings. For CCA tuning, the choices are Alpha tuning, Echos and AEchos. We also use two other naive CCA tuning mechanisms. The first is default CCA threshold, i.e. $-95dBm$. Equal power combined with default CCA is essentially the default configuration in 802.11. The second is disabling carrier sensing, which is suggested in [8]. For Alpha tuning, a relatively high α is aggressive and would lead to more collisions and a relatively low α is conservative and would lead to more exposed terminal problems. We pick the α manually such that the lowest link throughput would be similar to what we observe using iterative power control with AEchos.

Table I presents the network capacity for different radius sizes. Echos provides the best capacity when radius is longer than 12m, and Alpha has the best capacity when radius is short. The reason that Alpha's performace degrades with radius is that the iterative power control does not consider the CCA tuning mechanism, and can create asymmetric physical layer interference. For example, one link may use low power to enable concurrent transmission with another link, while other links uses high power level, causing asymmetric interference. Thus, if the α is set aggressively, then the low power link would starve. As a result, we set α more conservatively to prevent starvation. However, this causes the network capacity to decrease dramatically. Note that this poor interaction does not occur between Alpha and minimum power or equal power. The results also show that AEchos performs similarly to Echos, with the difference ranging from 2 to 20% for iterative. When equal power and default CCA threshold are used, the simulated regions is small enough that every source defers to each other. Thus, the network throughput is about the same, almost independent of radius size. Minimum power with default CCA threshold works well when radius is short, but the performance degrades when radius is long. Disabling carrier sense does not work very well in the simulation, since the traffic demand is high and many collisions happen. However, consistent with the observation in [8], the throughput can be higher than using the default configurations. Finally, iterative power control improves capacity by 2 to 75% over minimum power, and by 8% to 82% over equal power. This improvement roughly increases as average radius size increases, i.e. more diversity in client-AP distance.

Figure 6 shows the CDF of link throughput in one scenario, where there are 10 APs and radius size is 15m. The numbers in legend are the Jain fairness indices of each curve. In Echos (b) and AEchos (c), the curve of the iterative algorithm is roughly to the right of the other two curves indicating that it provides better throughput for almost all links. Iterative also has a better fairness index. In Alpha (a), the throughput of iterative is worse than that of min and equal power, but fairness is slightly better. Using default power and CCA threshold (equal power line in (d)) provides the best fairness but the worst network throughput. Graph (f) compares the CCA tuning mechanisms, all using the iterative algorithm. In comparing Echos with AEchos, we see that AEchos has a slightly worse fairness index but a slightly better minimum link throughput. Since we observed starvation for some nodes when using Echos, especially in settings where the radius was large, we believe AEchos, which had no such starvation, is a better choice. Among the CCA schemes, AEchos seems to provide a reasonable balance between fairness and throughput.

We have observed the following interesting properties of the iterative protocol in simulations that we do not show here. First, we have found that the periodic announcement is critical to the good performance of the protocol. Second, we found that the convergence time for the system is worst during heavy traffic workloads since data collisions prevent nodes from overhearing other transmissions. At worst, convergence takes around two minutes. Finally, variations in path loss over time (e.g. due to mobility or environmental changes) impacts iterative with AEchos more than minimum power with the default CCA. However, at no times did our proposed scheme perform worse. Finally, as we increase the number of APs in an area, the benefits of using the iterative algorithm increase.



(a) Alpha

(b) Echos

(c) AEchos

(d) Default CCA

(e) No Carrier Sensing

(f) Iterative with each CCA

Fig. 6. CDF of link throughput

VII. Related Work

Several efforts have explored adjusting either the transmit power of CCA threshold. The goal of [14] is to use the minimum transmit power to reduce power consumption. Similarly, [1] uses minimum power to improve capacity based on the author's analysis using the circle model. Like our study, [4] explores the ability of power control to improve fairness and capacity. However, it does not consider the interaction of CCA with transmit power control. Finally, Echos [15] tunes the CCA threshold while using fixed power levels. We make some key modifications to the Echos design to eliminate starvation.

Recent work has explored jointly tuning transmit power and CCA threshold to improve performance. In [11], the all nodes use the same configuration. However, our results show significant benefit from heterogenous configurations. In [6] and [12], the authors observe that keeping the product of power level and CCA threshold a constant ensures symmetric MAC layer behavior. The authors of [12] limit their design to use the same settings for all clients in a BSS. However, we found that adapting to each client is critical and can provide a 45% improvement over per-BSS tuning, when there are 20 APs and 60 clients.

TABLE I
Network Capacity (Mbps) with different radius size (m)

CCA	PC	3	6	9	12	15	18	21	24
Echos	Iter	31.4	27.8	28.1	20.3	17.9	17.4	15.2	9.82
	Min	29.9	25.4	24.2	17.8	15.9	12.5	8.70	7.17
	Equal	29.0	19.5	19.3	14.2	13.4	11.9	9.46	8.09
AEchos	Iter	30.8	26.6	27.6	19.8	17.2	15.7	12.7	8.75
	Min	30.2	22.0	19.8	16.3	14.1	12.1	8.37	7.35
	Equal	25.3	19.3	18.0	13.0	11.3	9.03	7.00	6.20
Default	Iter	31.4	27.0	23.5	17.3	14.0	11.0	8.15	9.25
	Min	30.3	26.0	22.5	17.3	13.8	10.6	7.61	6.83
	Equal	6.40	6.31	6.31	6.12	6.12	6.02	5.96	5.87
Alpha	Iter	32.0	29.8	28.4	11.8	9.60	12.8	8.60	8.91
	Min	30.3	26.4	24.1	18.1	12.5	9.88	6.48	9.08
	Equal	28.2	22.0	18.7	15.7	12.1	10.8	7.75	8.72
No	Iter	16.7	17.0	11.8	7.39	6.51	4.73	4.32	2.02
	Min	15.6	11.4	11.7	6.31	5.37	3.21	3.09	0.71
	Equal	15.2	8.31	9.05	3.39	2.98	2.28	2.49	0.90

VIII. Conclusion

We have presented an interference-aware power management protocol. In the protocol, nodes collect information about the RF environment by piggybacking signal strength and path loss information on their own transmissions and promiscuously listening for transmissions from other nodes. Using the collected data, each node executes a power control algorithm that iteratively increases the number of concurrent transmissions that can take place. Our protocol also incorporates an altruistic version of the Echos CCA tuning algorithm. Our evaluation of the protocol using the OPNET simulator shows that it improves network throughput by 22% to 87% compared with only tuning the CCA threshold, and by 2% to 67% compared with using minimum power levels.

References

[1] A. Akella, G. Judd, S. Seshan, and P. Steenkiste. Self-management in chaotic wireless deployments. In *MobiCom*, Cologne, Germany, aug 2005.

[2] Y. Bejerano, S.-J. Han, and L. E. Li. Fairness and load balancing in wireless lans using association control. In *MobiCom*, New York, NY, USA, 2004. ACM Press.

[3] 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.

[4] I. Broustis, J. Eriksson, S. V. Krishnamurthy, and M. Faloutsos. Implications of power control in wireless networks: A quantitative study. In *PAM*, 2007.

[5] S. M. Das, D. Koutsonikolas, Y. C. Hu, and D. Peroulis. Characterizing multi-way interference in wireless mesh networks. In *WiNTECH Workshop*, 2006.

[6] J. A. Fuemmeler, N. H. Vaidya, and V. V. Veeravalli. Selecting transmit powers and carrier sense thresholds for csma protocols. Technical report, UIUC, 2004.

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

[8] K. Jamieson, B. Hull, A. K. Miu, and H. Balakrishnan. Understanding the real-world performance of carrier sense. In *E-WIND Workshop*, Philadelphia, PA, August 2005.

[9] G. Judd and P. Steenkiste. Using emulation to understand and improve wireless networks and applications. In *NSDI*, May 2005.

[10] V. Kawadia and P. R. Kumar. Principles and protocols for power control in ad hoc networks. *IEEE JSAC*, 2005.

[11] T.-S. Kim, J. C. Hou, and H. Lim. Improving spatial reuse through tuning transmit power, carrier sense threshold, and data rate in multihop wireless networks. In *MobiCom*, New York, NY, USA, 2006. ACM Press.

[12] V. Mhatre, K. Papagiannaki, and F. Baccelli. Interference mitigation through power control in high density 802.11 wlans. In *Infocom*, 2007.

[13] J. Padhye, S. Agarwal, V. N. Padmanabhan, L. Qiu, A. Rao, and B. Zill. Estimation of link interference in static multi-hop wireless networks. In *IMC*, 2005.

[14] N. Poojary, S. Krishnamurthy, and S. Dao. Medium access control in a network of ad hoc mobile nodes with heterogeneous power capabilities. In *ICC*, 2001.

[15] A. Vasan, R. Ramjee, and T. Y. C. Woo. Echos - enhanced capacity 802.11 hotspots. In *Infocom*, 2005.

[16] R. Wattenhofer, L. E. Li, V. Bahl, and Y.-M. Wang. Distributed topology control for power efficient operation in multihop wireless ad hoc networks. In *Infocom*, APR 2001.

[17] K. Whitehouse, A. Woo, F. Jiang, J. Polastre, and D. Culler. Exploiting the capture effect for collision detection and recovery. In *IEEE EmNetS-II*, May 2005.

[18] J. Zhu, B. Metzler, X. Guo, and Y. Liu. Adaptive csma for scalable network capacity in high-density wlan: a hardware prototyping approach. In *Infocom*, 2006.