NashWifi: Coordinated Management of Heterogenous Home 802.11 Networks Using Bargaining

Aditya Akella
University of Wisconsin-Madison

Pratap Ramamurthy
University of Wisconsin-Madison

Almir Mutapcic
Stanford University

***Abstract*—In this paper, we present an economically principled framework for coordinated configurations of heterogeneous wireless networks. Our approach, called NashWiFi, is based on the concept of Nash bargaining. We argue that NashWiFi can improve both the global performance of a large diverse 802.11 deployment, while also improving the local performance of individual applications in an equitable fashion. NashWiFi can accommodate applications with diverse needs ranging from file transfer to HD streaming. Extensive experiments using an implementation of NashWiFi support in the MadWifi driver and large scale simulations using Qualnet establish the salient features of our approach.**

I. Introduction

Wireless networking technology has made vast forays into our daily lives. An increasing number of homes use 802.11 devices for their Internet access. Two important trends have come to define WiFi deployment and usage in recent years. First, recent studies have shown growing densities of wireless deployments in metropolitan areas [1]. Today, tens of wireless networks exist in very close proximity to each other in densely populated areas. Second, 802.11 network usage has moved well beyond just browsing the Internet. Today's home users run a variety of sophisticated and diverse applications over their WiFi networks. These include receiving high definition content over WiFi to XBox receivers (which are in turn connected to TVs) [2], streaming music from desktops to stereos and streaming videos to portable devices such as PSP [3], running VoIP over WiFi [4], and finally, P2P file sharing. While the requirements of some of these applications are very stringent (e.g., HD streaming needs a minimum of 3Mbps of bandwidth), others such as file transfer have elastic demands.

A key challenge arising from the high deployment densities and diverse applications is how to share the available wireless spectrum among the contending wireless transmissions. Optimal sharing of the spectrum entails meeting two sets of requirements, one local and the other global. First, different applications within a single home wireless network should be able to meet their *local* performance requirements. Second, wireless networks, each owned by a different individual entity, must be able to coexist and share the available spectrum in a *globally* optimal and fair fashion. In this paper, we propose a coordination-based framework for configuring home wireless networks that helps simultaneously meet these two requirements.

978-1-4244-5489-1/10/$26.00 © 2010 IEEE

Today, there are few ways to configure a wireless network to meet its application requirements. One option is to employ a unilateral myopic configuration, such as using the highest transmit power to ensure a good signal quality for high bandwidth transmissions, and not yielding transmission opportunities to contending APs (e.g. by ignoring carrier sense). Although this may be locally optimal for some applications, it could result in poor global performance and can cause unfairness in the spectrum allocation. Another option is to unilaterally adopt conservative configurations [1], [5], e.g., for selecting frequencies and transmit powers. Unfortunately, this could result in some wireless networks obtaining much poorer performance than even default configurations, affecting bandwidth-demanding applications in such networks.

In this paper, we develop an economically-principled framework called NashWiFi which *incents* independently-deployed home wireless networks to participate in *coordinated configuration*. Our approach is based on the concept of *Nash Bargaining* [6], [7]. In a sense, home wireless networks participating in our framework bargain or negotiate to arrive at mutually-acceptable configurations. In the event that such configurations cannot be found, wireless networks simply revert to their default configurations.

Nash bargaining offers several important properties that make it an attractive primitive to employ in coordinated configuration of future home networks. (1) The outcome of Nash bargaining is inherently *Pareto optimal* and *fair* [6], [7]. At the optimal configuration, any unilateral change to the configuration of a wireless network will hurt the performance of applications in one or more wireless networks; this ensures that spectrum resources are being used efficiently. In terms of fairness, Nash bargaining can yield equitable improvements to the applications in all participating wireless networks. NashWiFi selects a set of configurations that "distributes" the benefits across all wireless networks and applications, allocating more spectrum resources to applications and networks with greater requirements. In some scenarios (e.g., extremely dense or extremely sparse settings) we find that using default configurations is the best. In all cases, NashWiFi intrinsically converges to whatever configuration setting—default or something else—is best for the deployment on hand. (2) NashWiFi can guarantee *universal improvement*: either all home networks will see improvements in their applications' performance, or their performance will be the same as under default configurations (but never worse). (3) NashWiFi can accommodate networks running diverse applications with very different needs. For instance, wireless networks where the traffic is dominated by high definition TV and wireless networks where regular HTTP traffic is predominant can both coordinate to arrive at mutually-beneficial configurations using our framework.

NashWiFi requires a centralized bargaining server accessible via the wired back-end connections of each wireless network. The server could be run by the broadband ISP or by a participating network. Individual wireless networks periodically take measurements of the RF environment they are operating in (e.g., path loss between various transmit-receive pairs) and transmit them to the bargaining server along with their local application requirements. The server simulates bargaining based on the input and computes the best settings for each wireless network.

We address several practical issues in realizing this framework, such as: (1) modeling and specifying diverse requirements of participating networks and applications, (2) using hybrid passive-active probes to collect the necessary input information about RF properties of a deployment in a scalable fashion, while imposing low overhead on active applications, (3) accommodating churn in the system (i.e., users entering and leaving) and explicitly coordinating the actions of different participating networks (such as when they take the input measurements), and, (4) helping the bargaining server to quickly derive the optimal configuration with the aid of a fast throughput estimator.

We have implemented NashWiFi in the MadWiFi Linux driver for commodity wireless cards. We have implemented the bargaining protocol and the underlying coordination framework on a Linux desktop. We have experimented with this implementation in a variety of small lab settings. We have also implemented NashWiFi in the Qualnet simulator. Our simulations explore the properties of our approach under different topologies and deployment densities.

We find that NashWiFi offers substantial improvements compared to the current default configurations: for 10 home wireless networks spread randomly in a 300m × 300m area, the aggregate throughput improves by a factor of 2.1× in simulation. Furthermore, the throughput of each node improves by a positive extent, in some cases by a factor of 24×. We do find that under the most dense settings, e.g. 10 wireless networks spread in a 50m × 50m area, our approach converges to the default configurations. We also experimented with myopic configurations as well as other simplistic coordination approaches; we found that such approaches are unsuitable because they *starve* or otherwise hurt the performance of some home networks. We also experimented with networks having a wide variety of utility functions and found our approach to be highly flexible. In particular, our approach automatically assigns enough spectrum resources to first meet the needs of with stringent bandwidth needs and then reallocates remaining resources to applications with more elastic needs.

II. Related Work

Unilateral schemes. In [5], the authors presents a spectrum sharing technique based on APs and clients performing random hopping over a sequence of frequencies. However, it is not clear if this results in fair and optimal allocation of the medium. Commercial products such as [8], [9] offer intelligent spectrum management, where each base station snoops all channels every T seconds to determine which channel is the least active and then uses it for the next T seconds. If all WLANs adopt this approach, it could lead to oscillations of channels due to lack of explicit coordination.

Coordination based approaches. Some past studies have proposed coordination based approaches to wireless network configuration. In [10], the authors present a framework for jointly selecting the transmit powers and carrier sense thresholds of wireless networks in a dense deployment (The authors in [11] use similar techniques to configure multiple base stations in a single campus wireless network). The authors do this by formulating an optimization problem where the objective is to optimize the aggregate throughput achieved by all the wireless networks. In [12], the authors have re-considered the benefits of this joint optimization framework when applied along with other approaches for frequency and user assignment (for load balancing). However, because these approaches do not take local performance into account explicitly, they could cause the performance of some networks to deteriorate creating negative incentives for such networks to participate in coordination. Also, unlike these approaches, ours accommodates networks with different performance requirements.

Bargaining in other contexts. The idea of enabling coordination among independent competing entities has been applied in in prior work on inter-domain traffic engineering protocols [13]–[15].The key difference in our setting is that there are multiple agents (tens of WLANs as opposed to two ISPs). Furthermore, the agents in our setting can have complex requirements (i.e. utility functions), while Shrimali et al only consider simplistic objectives like average delay.

III. NashWiFi

In this section, we present background and details of our NashWiFi bargaining-based coordinated configuration approach. As stated in §I and as we show in Section V, lack of coordination could lead to inefficient configurations of home wireless networks. While, coordination can address some of these ills, not all coordination based approaches are likely to be applicable. A naive approach such as centrally deriving configurations which optimizing the sum total throughput of transmissions in the system is unsuitable. While such an approach could guarantee that the medium is utilized efficiently overall, it cannot guarantee that the requirements of individual home networks are met in a fair fashion. As shown in Section V, in some cases, a handful of home networks could observe poorer performance than under no coordination. In other cases, some home networks may see little or no improvement while others enjoy substantial benefits. Home networks which get the short end of the stick are unlikely to be incentivized to participate in coordination and will switch to default or myopic configurations.

An ideal coordination based approach must have the following properties for home networks of the future to participate in: (1) **Efficiency:** It should improve the overall performance of the deployment to the best possible extent. (2) **Fairness:** It should treat each home wireless network in a fair fashion. Under no circumstances should some networks be allowed to gain while others lose from participating in coordination. More generally, all networks should observe equitable improvements. (3) **Support for heterogeneous applications:** It should accommodate networks that run diverse sets of applications with very different performance requirements. We argue that coordination based on the principles of Nash bargaining satisfies these requirements.

Next, we present an overview of the concept of Nash bargaining and discuss its key properties. Then, we describe how bargaining is incorporated into our NashWiFi framework. Finally, we provide details of how bargaining based configurations can be obtained in practice.

A. Nash Bargaining

We explain the concept of Nash bargaining using a simple example of two "players"; our description extends in a straightforward fashion to multiple players. Either player in this discussion is a home wireless network competing for the wireless spectrum. We assume that the agent utilities are interdependent and cardinal, where by cardinal we mean that the actual values of utilities matter – as opposed to ordinal utilities where only the relative ordering of outcomes matters.

The "feasible region" for the wireless networks is defined as the region of possible wireless network configurations where both would do better off (in terms of observed performance) compared to the current local configuration. The configuration of a wireless network can include various settings which impact how the wireless network contends with others, such as: (1) the transmit power along with the carrier sense threshold; (2) the frequency; and (3) the transmission schedule. In this paper, we focus on configuring parameters in #1 above, although our approach can be extended to jointly compute #1, #2 and #3.

The *breakdown point* in bargaining reflects the situation where wireless networks use uncoordinated local strategies such as default or myopic configurations. Bargaining is said to be conducted with respect to the breakdown point as a baseline: in other words, wireless networks are trying to bargain for configurations which offer improved performance relative to the breakdown point; if no such outcome can be obtained, then the networks fall back to the breakdown configurations (i.e. the bargaining breaks down).

A fair and Pareto efficient outcome, also know as the Nash solution, can be obtained when the bargaining entities collectively maximize the *Nash product*:

$$\begin{array}{ll}\text{maximize} & \prod_{i=1}^{n} u_i \\ \text{subject to} & u_i \geq u_i^*,\end{array}$$

where u_i is a function which reflects the performance of the i^{th} player as a function of the configurations of all players in the system. We refer to the u's as the *utility functions* of the wireless networks, and u_i^* denotes the value of the utility function at the breakdown point.

We now discuss salient properties of Nash bargaining. A good bargaining solution should satisfy five properties [7]:

(1) **Pareto efficiency:** The bargaining outcome should be *Pareto optimal*. As mentioned earlier, this ensures that the available resources are used efficiently.

(2) **Symmetry:** The solution should provide equal gains to participating players when the feasible region is symmetric (or agnostic of player identities).

(3) **Independence of affine transformations:** The solution should be unaffected by any affine transformations (that is, shifts and scalings) applied to the utility functions of the bargaining entities. So, if the solution is given by $\{u_i^{NB}\}_{i=1}^n$ for some performance functions $\{u_i\}_{i=1}^n$, and u_1 is scaled and shifted to $\alpha_1 u_1 + \beta_1$, then the solution should change to $(\alpha_1 u_1^{NB} + \beta_1, \{u_i^{NB}\}_{i=2}^n)$.

(4) **Independence of irrelevant alternatives:** The addition of irrelevant alternatives should not change the solution. That is, for feasible regions F and G, if $\{u_i^{NB}\}_{i=1}^n \in soln(F)$, $G \subset F$, and $\{u_i^{NB}\}_{i=1}^n \in G$ then $\{u_i^{NB}\}_{i=1}^n \in soln(G)$.

(5) **Universal improvement:** This is the most important fairness requirement. It means that *no* participating entity, i.e. wireless networks in our setting, should observe a *decrease* in performance from using the solution. This is important because a wireless network whose performance decreases will simply not participate.

Nash bargaining [6], [7] is the only approach that is known to satisfy all five criteria. Also, the product objective function underlying Nash bargaining allows different wireless networks to have different utility functions. The improvement observed by a wireless network due to Nash bargaining is relative to the utility function the wireless network is using locally. Because of Nash bargaining's globally efficiency, its ability to meet local requirements satisfactorily, and its ability to accommodate heterogeneous requirements of network, we believe that Nash bargaining is a useful primitive to embed into coordinated configuration approaches for heterogenous home networks.

B. Approach

We now present the details of our NashWiFi framework.

In the NashWiFi framework, bargaining is run by a centralized bargaining server which optimizes the Nash product for a collection of home wireless networks. All participating wireless networks update the server with the relevant variables required for solving the Nash product. The server calculates the configuration parameters (e.g. power, or frequency or a combination of both) for each wireless network and configures the wireless networks with them. The configuration server could be run by an ISP, or one of the wireless networks could act as a server. We assume that the IP address of the server is known to all wireless networks, and vice-versa. Thus, communication between the bargaining server and the wireless networks could happen over the wired Internet.

Before presenting the details of the optimization performed by the bargaining server, we outline the assumptions made in our framework. We assume, for simplicity that all home wireless networks are on the same frequency. We focus exclusively on configuring the transmit powers and the carrier sense thresholds on the wireless networks in a coordinated fashion. We assume that each AP could use different transmit power but the same carrier sense threshold for transmissions to each of its clients. Also, clients use the same transmit power and carrier sense thresholds as their APs. Furthermore, the model we employ assumes that all transmissions use multi-rate adaptation, and RTS/CTS is off.

Now, suppose that we have a dense wireless deployment with n home networks in all, and that all networks participate in the NashWiFi framework. Suppose that each home network i has c_i clients. These clients may be involved in diverse applications including browsing, file transfer, streaming video, streaming audio, VoIP etc.

With each transmitter j, associate transmit power $P_j > 0$ and a carrier sense threshold $cs_j > 0$. (Recall that for simplicity, we have assumed that traffic exists only from APs to clients.) Denote the vectors of power and carrier sense thresholds as $P = (P_1, \ldots, P_n) \in \mathcal{P}$, and $c = (cs_1, \ldots, cs_n) \in \mathcal{CS}$, where $\mathcal{P}$ and $\mathcal{CS}$ are the sets of available power levels and carrier sense thresholds.

The goal of the bargaining server is to solve the following optimization problem for the $(power, carrier\text{-}sense\ threshold)$ configurations of the n wireless networks:

$$\begin{array}{ll} \text{maximize} & U = \prod_{i=1}^{n} U_i \\ \text{subject to} & U \geq U_{\text{break}} \\ & 0 \leq P \leq P_{\max} \\ & c_{\min} \leq c \leq c_{\max}, \end{array} \qquad (1)$$

where the optimization variables are powers P, and the carrier sense thresholds cs between the AP-client pairs $1, \ldots, n$. U_i is the utility function of the ith network and is some function of client throughput in the network. U_{break} is the vector of values of the utility functions of the wireless networks at the break-down point.

To aid the bargaining server in its optimization, the participating networks provide it a *path loss matrix*, represented by G, as input. An element G_{ij} denotes the path loss between any transmitter j (an AP) and the receiver i (a client not associated with the AP). G_{ii} is the path loss between receiver i and its AP. Thus, G has $\sum c_i$ rows and n columns. In addition to G home networks also provide their utility functions and the values of the utility functions at the breakdown configurations to the bargaining server. We describe how to collect G and the breakedown utilities scalably in §IV.

Given the input, the goal of the bargaining server is to search through the space of configurations to obtain one which optimizes the Nash product. The key challenges are taking *heterogeneous requirements* into accounts and to *realistically estimate the performance* that different wireless transmissions are likely to achieve under a given configuration.

We describe how to model diverse utility functions in the §III-B1. We then describe an approximate throughput estimation model in §III-B2 using which the bargaining server estimates the throughputs achieved by the clients in different networks, and hence the utility functions of the different networks, for various network wide configurations in a fast manner.

1) Heterogenous requirements: A key requirement of our coordination-based framework was to accommodate heterogeneous requirements. Heteregeneous requirements of home networks could be modeled by each home network employing a different utility function. For example, a simple utility function is the average throughput achieved by clients. Another example is a binary utility function which is 1 if the throughput achieved by a certain client (say an Xbox streaming high definition video) is above a certain threshold (e.g. 3Mbps [3]), and 0 otherwise.

A more complex utility function could employ a combination of terms: for instance a home network with two clients may want to ensure reasonable throughput performance for a client involved in file transfers, along with a certain minimum throughput for its HD streaming client. Such a network could employ the following utility function: $U = t_1 * step(t_2, 3Mbps)$, where t_1, t_2 are the throughputs of the two clients, and $step()$ is a step function such that $step(t, T) = 1$ if $t > T$, $step(t, T) = 0$ otherwise for some constant T. This utility function gives much more weight to the streaming client. Alternately, to give more weight to the browsing client, the network could instead use $U = t_1 * (w + step(t_2, 3Mbps))$, for some $w > 0$.

We evaluate the effectiveness of our bargaining approach under a variety of different utility functions in V-E.

2) Throughput estimation: Throughput estimation in 802.11 is a notoriously difficult problem but recently there has been success in accurately estimating link capacity from path loss measurements [16]. While the approach we propose is similar at a high level, it differs in the goals. Our ultimate goal is to find a global configuration that *satisfies the requirements* of all participating entities. Thus, an absolutely accurate throughput estimator is not necessary and an *approximate estimator* is sufficient. Relaxing the accuracy requirement also enables us to design approaches that meet other practical goals such as efficient reconfiguration and scalability with deployment size.

The approach we propose for approximate throughput estimator attempts to derive worst-case estimates of throughputs of various transmissions. Only when the worst-case estimates all satisfy the constraints imposed by the individual networks do we consider a configuration to be a likely candidate for bargaining. We divide the approximate throughput estimation problem into the following sub-parts to help obtain reasonable lower bounds on throughputs achieved by transmissions in an 802.11 deployment:

1. Receiver side interference model. This model accounts for the factors that contribute to the bit error rate (BER) of the channel. BER depends on the channel properties and the noise sources. So, in order to improve the BER under the given environment, we can reduce the transmit power of other transmitters in the area, which would help increase the SINR. SINR can be calculated for the ith receiver as:

$$SINR_i = \frac{G_{ii}P_i}{\sigma^2+\sum_{j\neq i} G_{ij}P_j}, \qquad i,j = 1,\ldots,n$$

In the above equation the $SINR_i$ at any receiver i is a function of the transmit powers P_i of all transmitters in the region, G_{ij} the path loss between any transmitter j and the receiver i. However, the above equation is an underestimate of the SINR because, in 802.11 (which uses CSMA), any transmitter in the same collision domain as transmitter i would defer its transmission and therefore would not contribute to the interference noise at the receiver. Including this factor gives us a more accurate SINR estimate

$$SINR_i = \frac{G_{ii}P_i}{\sigma^2+\sum_{j\neq i}(1-N_{ji})G_{ij}P_j}, \qquad i,j = 1,\ldots,n$$
$$\text{where} N_{ji} = \begin{cases} 1 & \text{if } H_{ji}P_i \geq cs_j, \\ 0 & \text{if } H_{ji}P_i < cs_j \end{cases} \tag{2}$$

H is a matrix similar to the path loss matrix G with one key difference: the H matrix is the path loss between transmitters and measured at transmitters. N is the neighborhood matrix. N_{ji} is either one or zero respectively depending on whether the transmitter j can carrier sense transmitter i and defer to it.

2. Transmitter side deference model. This model takes into account the fraction of the time transmitter gets an opportunity to transmit. A transmitter has to wait if it senses the channel busy. The channel might be busy because of a nearby transmission and the amount of time taken by this transmission depends on the datarate of that transmission and the size of the packet.

Assuming the channel is equally shared between the transmitters and based on the above factors we approximate the throughput of the ith receiver (T_i) using the following equation,

$$T_i = \frac{PS_i}{t_{wait} + \frac{PS_i}{R_i} + \sum_{j\neq i} N_{ij}\left[\frac{PS_j}{R_j} + t_{wait}\right]}, \tag{3}$$

Here, PS_i is the size of the packet and R_i is the datarate — this is simply the highest possible datarate for which the BER is negligible given $SINR_i$. t_{wait} is the short waiting time for each packet and is equal to $(SIFS + DIFS)$. Notice here that to calculate the throughput of any BSS, the datarate and the packet size of all the transmitters in the region are also needed. The datarate can be found using the equation 2. For the maximum packet size, we can either conservatively assume maximum possible size for all transmitters, or this could be provided as input by the home networks.

3. Starvation conditions. Equation 3 gives the throughput of any link, however, the model is insufficient in many ways because 802.11 links suffer from starvation which is unaccounted for above. We identify three sets of conditions under which starvation occurs and we add corrections to the throughput estimated in Equation 3. We describe these next.

- Noise above CS level: Let us consider the case where the carrier sense threshold is $-60dBm$ and noise level at the transmitter due to other transmitters is $-55dB$. A high noise level could be because of multiple active transmitters each of which is below the carrier sense range (if a transmitter can be carrier sensed its transmission will not contribute to interference), and the sum total of the interference could be higher than the carrier sense threshold. When this happens, the transmitter will continuously go into defer state and will never get an opportunity to transmit even if the SINR is high for its link. We detect this condition and add a correction by using the following check:

$$T_i = \begin{cases} T_i & \text{if } NoiseAtTrans_i \leq cs_i, \\ 0 & \text{if } NoiseAtTrans_i > cs_i, \end{cases}$$
$$\text{where} NoiseAtTrans_i = \sigma^2 + \sum_{j\neq i}(1-N_{ji})H_{ij}P_j \tag{4}$$

- Carrier sense asymmetry: Consider the case where there are two transmitters T1 and T2. Let the carrier sense thresholds of two transmitters be $-70dBm$, and suppose T1 uses higher transmit power than T2. Asymmetry arises when T2 can sense the packet transmitted by T1 but T1 is not able to sense the packet transmitted by T2. In such situations, T2 defers continuously causing it to be starved. We add a correction to account for this form of starvation:

$$T_i = \begin{cases} 0 & \text{if } (N_{ji} = 0) \text{ and } (N_{ij} = 1) \forall j = 1,\ldots,n, \\ T_i & \text{Otherwise} \end{cases} \tag{5}$$

- Neighborhood clique: In this type of starvation can be explained with a simple example consisting of three transmitters T1, T2 and T3. Lets say, T1 and T2 are neighbors and T1 and T3 are neighbors, but T2 and T3 are not. Lets say T2 is sending the first packet and T1 defers for T2. As T1 is waiting for T2 to complete the transmission, T3 notices that there are no transmissions (because T3 is not T2's neighbor) and starts transmitting. So, effectively T2 waits for T1 and T3 to stop transmitting, and since T1 and T3 are not neighbors they can simultaneously transmit and starve T2. This is called *flow in the middle* starvation and was described in [17]. Such starvation can be generalized to larger topologies. Such starvation can be avoided altogether by enforcing that all neighbors of transmitter i should be neighbors amongst themselves.

Our evaluation in §V and §VI confirm that the approximate throughput estimation approach is accurate for our purposes and fast in practice.

Next, we describe some of the missing details from the above approach, namely the measurement of the path loss matrix, measurement of the values of the utility functions at breakdown configurations, and the algorithm employed to search and find the optimal configuration. While there can be several ways of performing each of these actions, we describe specific approaches that we have adopted in the context of our NashWiFi implementation.

IV. NashWiFi Architecture and Implementation

In order to be viable, the NashWiFi architecture must satisfy a few key properties: (1) **Seamless operation**: The NashWiFi framework and its configuration updates and changes should have minimal effect on clients that might have ongoing flows. (2) **Quick configuration/reconfiguration**: The NashWiFi framework should take negligible time to determine the optimal configuration for a particular deployment and operational setting. (3) **Robustness**: The implementation must be robust in the face of transient noise, and other time varying properties of the medium.

We describe our implementation next, higlighting alongside how we meet the above requirements. In the interest of brevity we only describe the most important details of our implementation.

A. Implementation of NashWiFi Components

Bargaining server control channel: The bargaining server is the central controller of the system. The server sends a command to the APs to periodically conduct measurements of the path loss matrix, and the values of their utility functions at the breakdown configuration. The APs then communicate to the configuration server with details about path loss, and available throughput. The configuration server then uses a Simulated Annealing Engine (described next) to find the optimal configuration with which it configures the networks. The configuration server can be located and operated from anywhere in the Internet, which makes it very attractive to be operated by a third party. It is best, however, for the server to be located in close proximity to the dense home wireless networks' deployment.

Bargaining emulation using simulated annealing: To aid the bargaining server obtain an optimal bargaining solution, we adopt a simulated annealing based approach. The input to the simulated annealing module of the bargaining server are: (1) Path loss measurements, (2) Measurements of the values of the utility functions at the breakdown configurations, (3) The utility function chosen by each AP and (4) The physical layer properties (transmit power levels, and carrier sensing range). Given the above inputs, we create the AP-client path loss matrix G, inter-AP path loss matrix H and constraints based on the breakdown utilities. Our simulated annealing based algorithm works in rounds. We start with some initial configuration, e.g., default transmit power and default carrier sense threshold for each home network. In the first round, we conduct simulated annealing over transmit powers. Following this, we optimize carrier sense thresholds. We start with the set of transmit powers determined at the end of the above process as a baseline and then iterate over the carrier sense thresholds. These above two rounds are repeated in sequence for 50 times (thus, there are a total of 100 rounds).

APs and Clients: The AP and clients run a Madwifi driver modified according to the NashWiFi framework. The modifications in the driver are mainly for taking the necessary measurements, which we describe next. All required measurements are controlled and initiated by the server.

Path losses. To measure the path losses, APs transmit a management packet identified by a specific subtype (we use unused subtype `07`). At the driver level management packets with subtype 07 are received by a client even if transmitted by other APs that the client is not associated with. Whenever NashWiFi clients receive such a management packet they note the RSSI (Received Signal Strength Indicator) on the packet header and the sender of the packet, and send this information back to the AP. Since the management packets are transmitted at a fixed high power, each AP can obtain the path loss between other APs and its clients from the RSSI values reported by the clients. Each AP then communicates the path loss information that it has learned to the bargaining server. In this manner, the bargaining server collects the G path loss matrix. The H path loss matrix which captures path loss between APs can be collected in a similar fashion.

At the beginning of the path loss measurement phase all the nodes (APs and clients) change their configuration to the settings that can enable path loss measurements to be taken in a robust fashion; this is crucial to ensure that the path loss matrix G can be estimated as completely as possible. To ensure this, APs need to use high transmit power. Furthermore, all APs and clients should use the lowest possible reception threshold.

Breakdown utiilities. We now describe the measurement of the breakdown utility values. Since the utility functions are all based on throughputs achieved by clients associated with the APs, this boils down to estimating the throughput at the breakdown configurations.

To obtain the throughput measurement, the APs fill their transmission queues in the driver with the appropriate number of dummy data packets to each of its clients. The packets are transmitted by the AP at the fastest rate allowed by the medium and by the underlying data rate adaptation algorithm. The clients measure the throughput based on the time taken to receive a certain number of packets. In order to obtain reliable measurements, we measure for 500ms.

An important issue in the design of NashWiFi is *how frequently to measure or reconfigure*. In the general case, path loss and breakdown utility measurements could take place even tens of minutes to an hour (this depends on the timescale of the change in the path losses). However, if a new client associates with an AP, or if the demand of a client changes, then NashWiFi should be reconfigured immediately. In such situations the AP can inform the bargaining server of the change; the server can immediately trigger a system-wide measurement. Not all measurement phases need be immediately followed by a reconfiguration. In particular, reconfiguration needs to take place only if the new configuration is significantly different from the old one and is likely to result in widely different throughputs. Also, reconfiguration does not have to start from a random initial configuration; instead the current configuration could be used as a good starting point. Reconvergence thus takes just a few 10s of iterations under moderate churn.

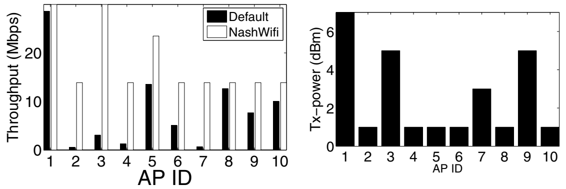


Fig. 1. (a) Throughputs (Mbps) per AP from NashWiFi, Default. (b) Tx powers used by NashWiFi.

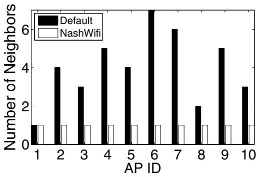


Fig. 2. Comparison of the number of neighbors for Default and NashWiFi.

V. NashWiFi Simulation Results

In this section, we describe results from an extensive simulation-based study of the NashWiFi bargaining-based configuration approach. Our simulation experiments are based on multiple "baseline" topologies where each topology has 10 home 802.11a wireless networks with one AP and a client deployed randomly within a 50m × 50m 2-D area. To simulate the impact of the density of independent uncoordinated deployments on the benefits of our approach, we construct several "stretched" topologies from the above baseline 50m × 50m topologies. Each stretched topology is identical to the baseline except that all inter-AP distances are a factor s larger in the stretched topology compared to the baseline distances.

Our main goal is to understand the relative benefits offered by the NashWiFi approach compared to the current prevalent default configurations (i.e. highest transmit power, and default carrier sense threshold), which we call *Default*. The carrier sense threshold for Default is -69dBm and all transmissions are at 15dBm.

A. Analysis of a single setting

We present an initial set of observations highlighting the benefits offered by NashWiFi. We select a specific baseline topology with a stretch $s = 6$ implying that nodes are spread in a 300 × 300m area. For this topology, the mean number of neighbors of an AP is 3 under Default configuration. Two APs are "neighbors" if they can carrier sense each other.

The throughput comparison for NashWiFi and Default are shown in Figures 1(a). We note that there is a substantial throughput improvement from NashWiFi compared to Default: The aggregate throughput from bargaining-based coordination is 180Mbps, and that from Default is only 83Mbps, which is roughly 2.1X lower. The extremely low performance of the Default configuration arises due to several reasons, including APs backing off upon carrier sensing [18], as well as packet collisions due to interference and the resulting retransmissions. Furthermore, NashWiFi helps improve the throughput of each home wireless network; in particular, the throughput of APs 2, 3, 4 and 7 increase significantly.

Figure 1(b) shows the transmit powers selected by NashWiFi. The mean transmit power used is 4.5dBm, one tenth of the maximum (in mW), yet all nodes see an increase in throughput. Figure 2 shows a comparison of the number of neighbors of each AP. We note that in the NashWiFi configuration, the number of neighbors is one throughout which means no AP is in transmission range of any other AP. This reduction in the number of neighbors is an artifact of our explicitly modeling starvation conditions in the approximate throughput estimator.

B. Impact of Density

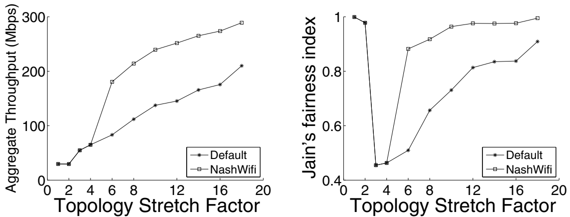


Fig. 3. (a) Comparison of aggregate throughput. (b) Jain's fairness index.

In Figure 3(a), we examine the effects of the deployment density on the performance of Default and NashWiFi configurations on the network's aggregate throughput. We focus on a specific baseline topology. The x-axis of this graph shows increasing stretches, or decreasing densities. A couple of salient features emerge from this analysis. First, NashWiFi outperforms Default or always at least matches the Default performance (Figure 3(a)), regardless of the density. At a stretch of 6, the aggregate throughput of NashWiFi is more than twice the aggregate throughput of Default. Second, for the topologies smaller than 200x200m NashWiFi is unable to find a better solution and so it intrinsically falls back to Default configuration. Note that we used a 2D topology and two-ray ground reflection model where the signal strength attenuation can be approximately modeled as being inversely proportional to the fourth power of distance between transmitter and receiver. The attenuation in an urban setting is much more complex. However, the key observation is that there is an upper bound on the density of deployment, beyond which NashWiFi falls back to Default configuration. Finally, as the density reduces, the aggregate throughput of Default configuration increases due to the decrease in the average number of neighbors. In all cases, none of the 10 APs saw a decrease in individual throughput compared to Default.

Fairness. In Figure 3(b) we show the trend in the fairness index when plotted against the stretch factor. For small stretch factors the fairness index for Default is very high (close to 1) because at the smallest stretch factor=1, the topology has 10 APs in a 100mX100m area, and so the entire set of APs is in carrier sense range. Therefore, they all share the medium equally and only limited by the packet level fairness of the 802.11 protocol. As we increase the stretch factor, we see that the fairness index decreases at first. This is because, some nodes have few neighbors and so could potentially get more opportunity to transmit. These nodes could get more throughput and this reduces the fairness index. For stretches

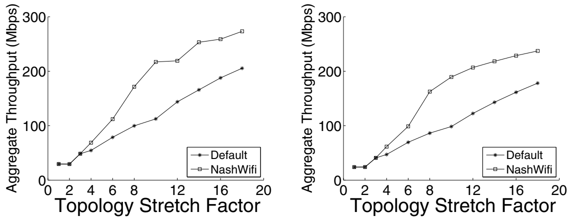


Fig. 4. Aggregate throughput averaged over 10 baseline topologies for (a) UDP and (b) TCP.

up to 4 (200X200m) the NashWiFi and Default have the same fairness because at these high densities, NashWiFi uses Default configuration as we noticed in Figure 3(a). However, at stretches of 6 and above where NashWiFi finds a more optimal configuration, we see that the fairness index is also much higher.

C. Impact of Topology and Traffic

The above results were specific to a particular baseline topology. To understand the bias introduced by the underlying deployment structure we also conducted simulations over nine other baselines. The observations for the ten baselines (including the one discussed above) are shown in Figure 4. We note that the relative benefits of coordination at each stretch vary with the underlying topology. NashWiFi continues to maintain its superiority under a variety of situations. The aggregate throughput performance of NashWiFi is significantly better than Default under almost all situations.

Figure 4 (b) shows the comparison of throughputs for Default and NashWiFi for a TCP workload. As we can see in Figure 4 (b), the NashWiFi performs better than Default configuration even for TCP traffic.

D. Multiple clients per AP

In the previous section we saw the performance of NashWiFi and compared it to the performance of Default. In the baseline topology that we used, each home wireless network had one client per AP. Next, we analyze the performance characteristics of NashWiFi when there are multiple clients associated to an AP. For this purpose, we use a different baseline topology, with 10 APs that are randomly positioned in a 300m X 300m area. Each AP has 2 clients randomly located within a circle of radius 10m centered on the AP. We use download traffic, from the AP to each client.

Figure 5 (a) is similar to 1(a) where we compare the throughputs of each home wireless network with Default and NashWiFi. In Figure 5 all APs use the *AVG* utility function that is the average throughput of all the clients. In this Figure, client 1 and client 2 are associated to AP 1, client 3 and client 4 are associated to AP2 and so on. As before, NashWiFi gives much better aggregate throughput. The aggregate throughput of all APs using Default configuration is 88Mbps, and NashWiFi gives 222Mbps a 2.5X improvement in aggregate throughput. Also, every client sees improved performance.

E. Heterogeneous utility functions

In the previous section we saw the throughput improvements of NashWiFi over Default in a multi-client topology, when all the APs use a homogeneous utility function. The utility function is a choice local to an AP which means, each AP could have a different utility function. We experimented with several situations where APs use very different utility functions (such as the ones illustrated earlier in §3). For illustrative purposes, to show how NashWiFi accommodates different utility functions and apportions resources across networks with different requirements, we present results from a specific setting. In Figure 5(b) we show results from the setting where APs 1-5 in the 10-network topology above each use *AVG* as the utility function. APs 6-10 each use *MIN* as the utility function: here, the utility function of an AP is step function over the minimum of the throughputs achieved by all clients of an AP. If the minimum throughput is above some threshold (different networks use different thresholds in this setting), then the utility is 1, and it is 0 otherwise.

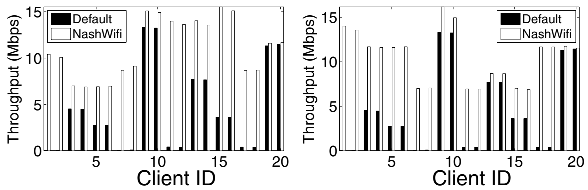


Fig. 5. Impact of multiple clients and different utility functions. Clients 1 and 2 are associated to first AP, clients 3 and 4 associated to second AP and so on. In (a) All APs use *AVG* as the utility and in (b) APs 1 through 5 use *AVG* and APs 6 through 10 use *MIN*.

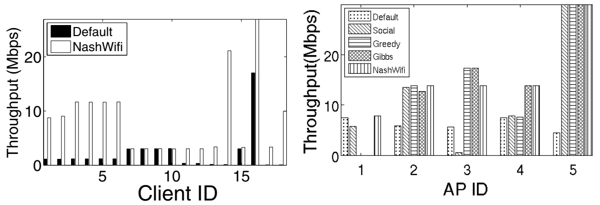
Compared to Figure 5(a), the throughput improvements are drastically different. For example, in Figure 5(a) clients 15 and 16 associated to AP 8 get close to 16Mbps when they use *AVG* as the utility function. However in Figure 5(b) (*MIN* utility function) we see that the throughput improvement is reduced significantly for these clients. Also, clients 1-6 see a greater increase in throughput compared to the case when the average utility function was used by all the networks.

Note that *MIN* is a less stringent utility function than *AVG*. For networks which use MIN, NashWiFi needs to satisfy the minimum throughput requirements and there is no incentive to provide any better throughputs to the individual clients. With the utility function *AVG* there is some incentive to provide more throughput even beyond the minimum.

In Figure 6(a), we show another scenario with more complex utility functions. In this topology we used 9 APs each with 2 clients. The first three APs use *AVG* as the utility function, the second three APs use *MIN* (with a threshold of 3Mbps), and the last three APs use $step(T_1, 3Mbps)) \times T_2$ where T_1 is the throughput of the first client and T_2 is the throughput of the second client. Notice, that clients 13, 15 and 17 get what they need (3Mbps) and the rest of the spectrum at the corresponding access points is utilized by the second client (client 14, 16 and 18, respectively) to give them high throughput performance.

F. Comparison with Other Approaches

In this section, we compare Default and NashWiFi configurations about other unilateral and cooperation based alternatives. In particular, we study, unilateral *socially responsible* configurations, so called because all the WLANs in this



(a) (b)

Fig. 6. (a) More diverse utilities. (b) Comparison with other approaches.

configuration unilaterally reduce transmit power to improve the global interference noise. At the other extreme, we examine a unilateral myopic or *Greedy* approach in which each wireless device uses high transmit power and high carrier sense threshold (to not yield transmission opportunities to others). Finally, we study a recently proposed joint configuration approach described in [10]. In this framework the objective function for the global coordination is approximately the sum of throughputs of all clients. However, the approach does not have the inherent notion of a breakdown point or breakdown configurations. The optimal solution is obtained using *Gibbs sampling*, a heuristic similar to simulated annealing. We implemented this approach in Qualnet and we refer to it as "Gibbs".

We compared the above approaches against Default and NashWiFi under a wide variety of topologies. To ease the exposition of difference between the five schemes we show the results from a simple 5 home network topology in an area of 150x150m. As before, the client is positioned at a random distance of 1-10m from the AP it is associated to. The utility function of each AP is simply the throughput of the client associated with it. Our observations are similar across other topologies we have studied.

Figure 6(b) compares the five strategies. In Default all 5 APs get approximately 6Mbps. All APs are in carrier sense range of each other and therefore the throughput is equally distributed. Social does have the effect of reducing the total noise in the network, but because of the poor signal qualities arising from unilateral reductions in transmit power, Social hurts the performance of AP1 and AP3 relative to Default. These APs would have no incentive to unilaterally reduce their transmit power. At the other extreme Greedy results in equal or better performance than Default for four of the APs, but it ends up starving AP1. We have noticed such starvation due to the Greedy strategy (much worse than the starvation that can arise in 802.11 with Default configuration) in other topologies as well. Hence it is not a good strategy to adopt in general. We note that NashWiFi offers better aggregate throughput and improved local performance to all APs (relative to Default) in comparison with the unilateral strategies Greedy and Social.

Gibbs and NashWiFi have similar aggregate throughputs (71Mbps vs 75Mbps[1]). However, Gibbs starves AP1. NashWiFi is able to provide 7Mbps to AP1 same as that at Default. Because of the starvation in Gibbs, AP1 has no incentive to stick with the coordinated configurations derived by Gibbs. Also, the additive objective function of Gibbs cannot easily accommodate diverse utility functions of the APs and hence cannot support networks with heterogeneous application requirements.

To summarize, both unilateral approach and prior cooperative approaches either result in poor aggregate throughput or starve some APs (or both). NashWiFi offers improved aggregate performance and better local performance compared to Default. Hence home networks have a greater incentive to adopt NashWiFi today.

VI. Experimental Results

We now present results from an experimental evaluation of NashWiFi in a small scale testbed.

Experimental setup. We installed NashWiFi in a small scale wireless testbed. The testbed has 10 Soekris ($Net4826-$50) nodes located in 3 adjacent floors. Each node has an Atheros card, and 128MB RAM and runs Linux 2.6, on a processor with clock speed of 233MHz. The Atheros cards support 8 levels of transmission powers. These nodes can be used as either APs or clients and we chose the mode based on the desired topology. We download UDP traffic from AP to the client in all the experiments below unless specified otherwise. To study NashWiFi, we created two different topologies and compared the throughput obtained from NashWiFi with default configurations.

Scenario 1: Flow in the Middle. Consider the topology shown in Figure 7(a).

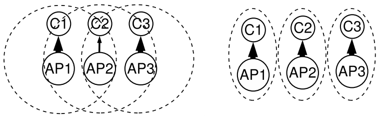


Fig. 7. *Scenario 1*: Topology with 6 nodes, Default at left and NashWiFi at right.

AP1 and C1 represent the access point and client in the first BSS and so on. Notice the dashed ring around each AP: this envelops all APs whose transmissions can be sensed by AP1. The carrier sense envelop was inferred ex post facto from the inter-AP path loss matrix H, transmit power and the carrier sense threshold.

The topology shown in Figure 7(a) shows a typical flow in the middle (FIM) starvation case. Table I shows the results for Default configuration and NashWiFi. With default, AP1 and AP3 get 5Mbps throughput but AP2 is starved. NashWiFi allotted 1mW transmit power to all APs and because of this lower transmit power, none of the APs can carrier sense the transmissions from other APs, as pictorially represented by dashed rings in Figure 7(b). Table I also shows the datarate of the transmissions. Notice that all the APs have chosen lower datarates under NashWiFi. Under default when AP1 and AP2 (likewise for AP2 and AP3) carrier sense and defer to each other, the only simultaneous transmission was due to AP3 (likewise, due to AP1) and so the interference noise was lower. This gives which gives a higher SINR and enables the use of

[1] Gibbs' aggregate throughput is lower although it attempts to maximize the sum of throughputs because the search heuristic could have been stuck in a local optimum. Such problems could also arise with simulated annealing in NashWiFi.

higher datarate. In the case of NashWiFi, since all transmitters are simultaneously active and the transmit powers are lower, the SINR for each transmission is not as high resulting in the lower data rates.

	BSS1	BSS2	BSS3
Default Throughput (Mbps)	5	0	5
NashWiFi Throughput (Mbps)	6	7	6
Default TxPower (dBm)	20	20	20
NashWiFi TxPower (dBm)	0	0	0
Default datarate (Mbps)	36	36	36
NashWiFi datarate (Mbps)	12	12	12

TABLE I
Scenario 1

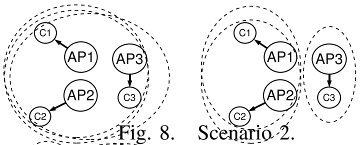


Fig. 8. Scenario 2.

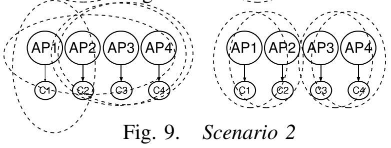


Fig. 9. *Scenario 2*

Scenario 2: A High interference richer topology. The topology shown in Figure 9 uses four pairs of nodes. We call it a high interference topology because AP1 cannot carrier sense transmissions from AP3 and AP4 and these transmissions add to BSS1's noise and therefore reduces the SINR. Results in Table II show that BSS1 cannot support even a 6Mbps datarate due to interference.

	BSS1	BSS2	BSS3	BSS4
Default Throughput (Mbps)	0	2	2	3
NashWiFi Throughput (Mbps)	6	5	3	3
Default TxPower (dBm)	20	20	20	20
NashWiFi TxPower (dBm)	10	0	10	10
Default datarate (Mbps)	0	24	12	24
NashWiFi datarate (Mbps)	12	12	6	12

TABLE II
Scenario 2

In the Default configuration the carrier sense envelops show that AP1 can only carrier sense AP2; AP2 can carrier sense AP1, AP3 and AP4; AP3 can carrier sense AP2 and AP4; AP4 can carrier sense AP2 and AP3. Under NashWiFi, all APs use lesser albeit different powers, and so the carrier sense envelops are smaller and this transmissions from lesser number of APs can be sensed. In NashWiFi, AP1 can carrier sense AP2; AP2 can carrier sense AP1; AP3 can carrier sense AP4; AP4 can carrier sense AP3. As a result, under NashWiFi, AP1 is able to use a datarate of 12Mbps and gets a non-zero throughput of 6Mbps. Note also that the throughputs of two of the three remaining APs have improved while the last AP has the same throughput as in the default configuration.

VII. Conclusions

Modern home wireless network deployments are defined by two features: high deployment densities and heterogeneous applications with diverse needs. In this paper we argued that coordinated configuration is crucial in order to support robust performance in these networks and to effectively meet the needs of the diverse applications. While there can be several approaches to achieve coordinated configuration, we believe it is important for any coordination-based approach to offer the necessary guarantees to encourage home networks to participate. We argued that the concept of Nash bargaining, which has several crucial properties such as Pareto efficiency, fairness, universal improvement and support for heterogeneous requirements, is an appropriate primitive to use in coordinated configuration. We developed a coordinated configuration called NashWiFi based on bargaining. Using extensive simulations in Qualnet and real experimentation using a MadWifi-based implementation, we showed that our approach can significantly improve the global and individual performance of wireless networks and can effectively support diverse requirements of networks.

Acknowledgments: This work was supported in part by an NSF CAREER Award (CNS-0746531), an NSF NeTS FIND Award (CNS-0626889) and a Grant from the UW-Madison Graduate School.

References

[1] A. Akella, G. Judd, S. Seshan, and P. Steenkiste, "Self-management in chaotic wireless deployments," in *MobiCom*, 2005.
[2] "HDTV over WiFi," http://www.engadgethd.com/2005/12/15/hdtv-over-wifi-general-impression%s/.
[3] "Stream video (DivX, XviD etc.) to PSP without having to encode first," http://www.afterdawn.com/news/archive/9110.cfm.
[4] "Belkin Wifi phone for Skype," http://www.belkin.com/skype/.
[5] A. Mishra, V. Shrivastava, D. Agrawal, S. Banerjee, and S. Ganguly, "Distributed channel management in uncoordinated wireless environments," in *MobiCom*, 2006.
[6] J. F. Nash, "The bargaining problem," *Econometric*, vol. 28, pp. 155–162, August 1950.
[7] R. B. Myerson, *Game Theory: Analysis of Conflict*. Harvard University Press, 1991.
[8] "AccessOne/Network OWS," http://www.strixsystems.com/products/products_main.asp.
[9] "Alcatel AirView Software," http://www.alcatel.com.
[10] V. P. Mhatre, K. Papagiannaki, and F. Baccelli, "Interference mitigation through power control in high density 802.11 wlans," in *INFOCOM*, 2007.
[11] J. A. Fuemmeler, N. H. Vaidya, and V. V. Veeravalli, "Selecting transmit powers and carrier sense thresholds in csma protocols for wireless ad hoc networks," in *WICON '06: International workshop on Wireless internet*, 2006.
[12] I. Broustis, K. Papagiannaki, S. V. Krishnamurthy, M. Faloutsos, and V. Mhatre, "Mdg: measurement-driven guidelines for 802.11 wlan design," in *MobiCom '07*, 2007.
[13] R. Mahajan, D. Wetherall, and T. Anderson, "Negotiation-Based Routing Between Neighboring ISPs," in *Proc. NSDI*, 2005.
[14] R. Mahajan, "Practical and efficient internet routing with competing interests," Ph.D. dissertation, University of Washington, 2005.
[15] G. Shrimali, A. Akella, and A. Mutapcic, "Cooperative interdomain traffic engineering using nash bargaining and decomposition," in *INFOCOM*, Anchorage, AK, May 2007.
[16] A. Kashyap, S. Ganguly, and S. R. Das, "A measurement-based approach to modeling link capacity in 802.11-based wireless networks," in *MobiCom '07*, 2007.
[17] J. Shi, T. Salonidis, and E. W. Knightly, "Starvation mitigation through multi-channel coordination in csma multi-hop wireless networks," in *MobiHoc '06*, 2006.
[18] K. Jamieson, B. Hull, A. Miu, and H. Balakrishnan, "Understanding the real-world performance of carrier sense," in *SIGCOMM Workshop on Experimental approaches to wireless network design and analysis*, 2005.