Symphony: Synchronous Two-phase Rate and Power Control in 802.11 WLANs

Kishore Ramachandran†, Ravi Kokku‡, Honghai Zhang‡, and Marco Gruteser†
†WINLAB, Rutgers University, North Brunswick, NJ, USA
‡NEC Laboratories America, Princeton, NJ, USA
†{kishore, gruteser}@winlab.rutgers.edu, ‡{ravik, honghai}@nec-labs.com

Abstract

Adaptive transmit power control in 802.11 Wireless LANs (WLANs) on a per-link basis helps increase network capacity and improves battery life of Wifi-enabled mobile devices. However, it faces the following challenges: (1) it can exacerbate receiver-side interference and asymmetric channel access, (2) it can incorrectly lead to lowering the data rate of a link, (3) mobility-induced channel variations at short timescales make detecting and avoiding these problems more complex. Despite significant research in rate and power control, state of the art solutions lack comprehensive techniques to address the above problems.

In this paper, we design and implement Symphony*—a Synchronous Two-phase Rate and Power control system, whose agility in adaptation enables us to systematically address the three problems, while maximizing the benefits of power control on a per-link basis. We implement* Symphony *in the Linux MadWifi driver, and show that it can be realized on hardware that supports transmit power control with no modifications to the 802.11 MAC, thereby fostering immediate deployability. Our extensive experimental evaluation on a real testbed in an office environment demonstrates that* Symphony *(1) enables up to 80% of the clients in 3 different cells to settle at 50% to 94% lower transmit power than a per-cell power control solution, (2) increases network throughput by up to 50% across realistic deployment scenarios, (3) improves the throughput of asymmetry-affected links by 300%, and (4) opportunistically reduces the transmit power of mobile clients running VOIP calls by up to 97%, while causing minimum impact on voice quality.*

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Wireless Communication*

General Terms

Algorithms, Design, Performance, Experimentation

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
MobiSys'08, June 17–20, 2008, Breckenridge, Colorado, USA.
Copyright 2008 ACM 978-1-60558-139-2/08/06 ...$5.00.

Keywords

WLANs, Rate adaptation, Transmit power control, Battery life, Mobility, Interference, Asymmetry, VOIP

1. INTRODUCTION

Projections that sales of mobile video phones and WiFi-enabled smartphones will exceed $100 billion in 2010 suggest that voice and data applications are fast converging from portable onto mobile devices and that WiFi is increasingly being used for last-mile network access [3, 6]. In light of these two trends, WLANs face (1) higher user mobility, (2) more stringent energy budgets, and (3) higher node densities. While these requirements will likely be addressed through a combination of several mechanisms, we believe that an adaptive transmit power and rate control solution will be an integral component of the overall system. Adaptive transmit power control can compensate for link changes due to mobility, improve spatial reuse by increasing simultaneous transmissions on multiple links, and reduce energy consumption for frequently transmitting handheld devices ((e.g., video call from smartphone)[1].

Despite decades of power and rate control research, few systems exist that jointly adapt transmit power and rate for WLANs on a *per-link* basis (partly due to the lack of hardware support for *fine-grained* power control until recently [36, 48]). In particular, while rate control is performed on a per-link basis, current solutions either use static or coarsely dynamic transmit power configurations on a per-AP or *per-cell* basis [2, 31]. These approaches forego per-link power control's benefits in a mobile environment. Current industry wisdom even seems to view power control as incompatible with VoIP applications [5], further discouraging its adoption. In general, adaptive transmit power control for WLANs faces the following challenges: (1) it can exacerbate receiver-side interference (also known as the hidden-terminal problem) and asymmetric channel access [31], thereby leading to unnecessary packet retransmissions and reduced fairness; (2) it can incorrectly lead to lowering the data rate of a link, thereby increasing the air-time on the channel; (3) mobility-induced channel variations at short timescales make detecting and avoiding these problems more complex.

[1] For instance, Broadcom's 802.11b/g interface consumes about 625mW and 425mW when transmitting at 15dBm and 7dBm [1], and 295mW when receiving. Assuming the device transmits half of the active time and receives in the rest half, this translates to an active-mode battery life increase of up to 27% when operating at 7dBm instead of 15dBm.

While some of these problems have been identified previously [23, 25, 31], to our knowledge, no integrated systems solution exists that *simultaneously* addresses all three challenges. Further, no per-link solution exists to address the asymmetric channel access problem in WLANs. The only work on WLAN asymmetry [31] is focused on a per-cell solution that cannot be easily extended to the per-link case, especially in the presence of user mobility, since it converges very slowly (30 seconds in one of their experiments).

In this paper, we propose Symphony—a novel synchronous two-phase rate and power control system that addresses all the above challenges in a unified framework. The key idea is to use a periodic reference phase during which all nodes operate at maximum power (as if no power control were used) to detect whether power control leads to adverse effects. Symphony is robust to user mobility, and can be easily realized on state-of-the-art hardware with no new modifications to the 802.11 MAC protocols. We implement Symphony in the Linux MadWifi driver [30], and demonstrate its efficacy through a detailed prototype evaluation. Extensive experiments on a real testbed in an office environment and on the indoor ORBIT testbed [42] demonstrate that Symphony (1) enables up to 80% of the clients in 3 different cells in an office environment to settle at 50% to 94% (3 to 12 dB) lower transmit power than a per-cell power control solution, (2) opportunistically reduces the transmit power of mobile clients running VOIP calls by up to 97% (15 dB), while causing minimum impact on R-score—a popular performance metric for quantizing voice quality, (3) addresses the problems of hidden terminals and asymmetric channel access, while improving the throughput of asymmetry-affected links by 300%, and (4) across 4 realistic deployment scenarios, increases network throughput by up to 50%.

In summary, we make three contributions in this paper:

1. We propose Symphony, the first per-link rate and power control system for WLANs that simultaneously addresses the problems introduced by power control in a comprehensive and easily realizable manner. By implementing Symphony in the Linux MadWifi driver [30] running over off-the-shelf hardware, and requiring no extensions to 802.11 protocols, we demonstrate that the solution is readily deployable.

2. As part of Symphony, we propose a novel and simple mechanism based on expected transmission time (ETT) to detect WLAN channel access asymmetry in a distributed manner. Unlike state-of-the-art per-link solutions in the ad-hoc domain, this mechanism does not require information about the topology [44] or the source of interference [47].

3. As part of Symphony, we propose RRAA+, an improved version of the state-of-the-art RRAA rate adaptation algorithm [50], which significantly reduces packet loss and makes links more robust to user mobility. RRAA+ is also useful standalone.

The rest of the paper is organized as follows. In Section 2, we describe the challenges of performing adaptive transmit power control, and discuss related work and their drawbacks. In Section 3, we discuss the design and implementation of Symphony, and present evaluation results in Section 4. Section 5 discusses open issues and limitations. Section 6 concludes.

2. PROBLEM FORMULATION

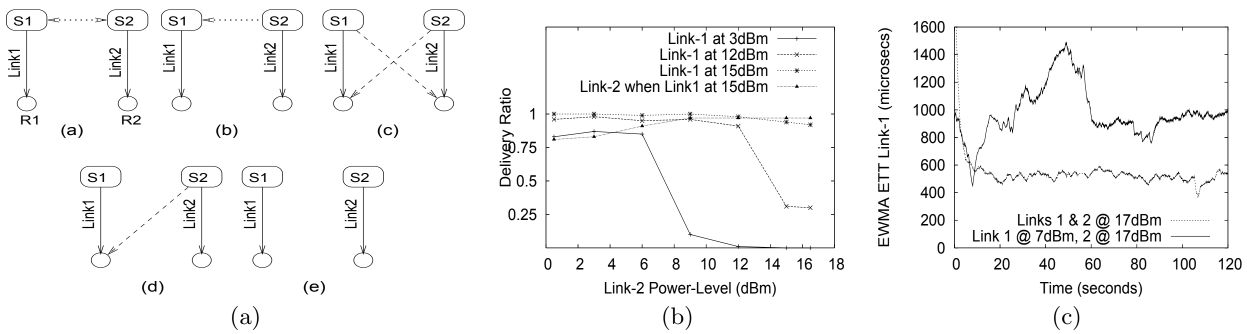
As users increasingly make WLANs the first choice for last-mile network access, both spatial reuse and battery life are crucial metrics for better user experience. With emerging mobile applications leading to increased data transfer over Wifi interfaces, and hardware [7] and protocol [9] improvements reducing the idle-time power consumption of these interfaces, transmit power becomes the dominating factor influencing battery lifetime. Secondly, with increasingly dense deployments of WLANs for continuous coverage to users, mitigating interference to maximize spatial reuse is a crucial design goal. Adaptive transmit power control on a per-link basis promises to improve both the above metrics.

2.1 Power Control Challenges

While per-link adaptive power control is beneficial, doing so can be challenging due to several reasons.

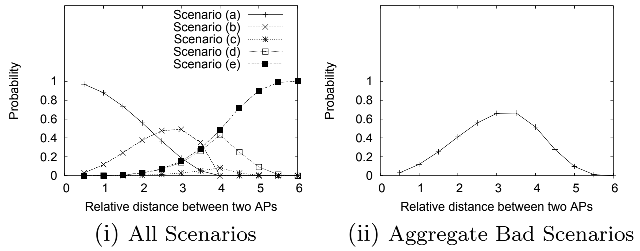
Receiver-side Interference and Asymmetric Channel Access: Transmit power control can introduce link asymmetry that leads to two problems: Receiver-side interference and Asymmetric channel access. Several previous works have already observed these two problems with power control [23, 25, 31, 39]. To provide a more quantitative characterization of (1) their *effect on performance in realistic settings* and (2) their *likelihood of occurrence*, consider a canonical network of four nodes—two senders and two corresponding receivers using the same 802.11 channel. Figure 1(a) identifies the different scenarios of interaction when the two links use different transmit powers. Solid arrows indicated that nodes are in communication range. Dotted arrows indicate that the senders are in carrier-sense range; in (b) S1 can hear S2, but not vice versa. Dashed arrows indicate unintended interference at the receivers. Scenario (a) represents fair channel sharing since S1 and S2 can carrier-sense each other. Scenario (b) represents the case of asymmetric channel access; whenever S2 has data to transmit, S1 does not get a fair chance to transmit. Scenarios (c) and (d) are two instances of receiver-side interference; while the senders are oblivious of each others' presence, packets sent by them collide at their receivers. Finally, scenario (e) is the ideal case of no interference and simultaneous transmissions on each link. Scenarios (b), (c) and (d) for any two links in the network can degrade the link and network throughput and fairness.

Setup: We use four laptops (with Atheros PCMCIA wireless cards) to emulate this canonical network—two laptops are configured as APs while the remaining two are configured as clients (one associated with each AP). Both AP-client links are on the same channel (802.11a channel 40). Each AP starts backlogged UDP transfers to its corresponding client and we fix the 802.11 MAC bit-rate to 54 Mbps. To emulate receiver-side interference, APs (senders) are positioned such that they do not carrier sense each other, and clients are placed such that their receptions may be corrupted by interfering transmissions on the other link (scenarios (c) and (d)). We use packet delivery ratio (PDR), calculated as the ratio of packets successfully received at each client to those sent out by its corresponding AP, as the metric. For asymmetric channel access, we place both receivers such that the network cannot be in scenarios (c) and (d). As the metric, we use the expected transmission time (ETT) [15, 11] of packets, with the minor modification that we only consider



(a) (b) (c)

Figure 1: Problems introduced by power control: (a) Scenarios of interaction between two links (b) Receiver-side interference on Link-1, and (c) Asymmetric channel access on Link-1.



(i) All Scenarios (ii) Aggregate Bad Scenarios

Figure 2: (i) Probability of scenarios (a)-(e), (ii) Total probability of problematic scenarios (scenarios (b), (c) and (d)).

packets that succeed without any retries, to approximate the *channel access delay* for each frame.

Results: Figure 1(b) shows the PDR observed by both links, in the experiment emulating receiver-side interference, when the transmit power of each link is changed. In this figure, the lines "Link-1 at 3dBm", "Link-1 at 12dBm", and "Link-1 at 15dBm" show that Link-1 is able to tolerate (due to physical layer capture) varying degrees of interference from Link-2 (Scenario (d)) depending on the relative difference in transmit powers. Similarly, the line "Link-2 when Link-1 at 15 dBm" shows that Link-2's PDR is affected by receiver-side interference at low transmit powers. In Figure 1(c), we plot the smoothed (using EWMA) ETT of packets on Link-1. When links are asymmetric (scenario (b)), Link-1 has higher channel access delay and hence higher ETT than when both links are symmetric (scenario (a)).

We now address the question, how frequently do the problematic scenarios ((b),(c) and (d)) occur? Since an experimental approach cannot sufficiently answer this question, we take an analytical approach. We derive the probability that each of the scenarios occur in a random geometrical graph with four nodes as above when the senders employ transmit power control. For brevity, we present the detailed mathematical formulation and analysis in a technical report [40], and just state the results here. Figure 2(i) shows the probability of occurrence of each scenario with distance between the senders, where the distance is shown as a factor of the communication range of the senders. Figure 2(ii) shows the sum of probabilities of all problematic scenarios. The graph clearly shows that these scenarios can occur very frequently in a real deployment. Detecting and avoiding these problems in mobile environments is even more challenging since they can be dynamically introduced for short time periods.

SINR Range (dB)	Rate	SINR Range (dB)	Rate
$\geq$ 24.56	54	[10.79,17.04)	18
[24.05,24.56)	48	[9.03,10.79)	12
[18.8,24.05)	36	[7.78,9.03)	9
[17.04,18.8)	24	[6.02,7.78)	6

Table 1: SINR vs. Rate (Mbps) for BERs $\leq 10^{-5}$.

Interaction with Rate Adaptation: In 802.11a/b/g wireless LANs, senders use one of multiple transmission rates for sending packets. The choice of the rate is determined by an estimate of the channel conditions either through packet loss [24, 28, 30], delivery ratio [50], throughput [11], or Signal to Interference and Noise Ratio (SINR) estimates [27, 21]. Conceptually, a link is expected to perform well at a chosen rate if the SINR at the receiver is above a threshold (Table 1) [27]. Rate selection and transmit power control are tied together; power control without considering rate can reduce the SINR, leading to reduction in rate and hence the link and network throughput. In this paper, we take a system perspective and choose a (minimum) power level for a link that does not compromise the achievable rate. From the table, it can be seen that for supporting 54Mbps, the transmit power can be reduced until it reaches the SINR threshold limit of 24.56dB. Similarly, say, if 54Mbps and 48Mbps cannot be supported even at maximum allowed transmit power, then power can be reduced until the SINR approaches 19dB, which still allows operation at 36Mbps.

While such rate and power selection is easily realizable with precise knowledge of receiver SINR [27], reliable SINR measurements and reports in the presence of mobility cannot be achieved at fine timescales due to their overhead. Consequently, we rely on *estimating* the channel conditions based on the delivery ratio of a window of packets, similar to past works [50]. Such an approach, however, makes rate and power selection non-trivial. To illustrate, consider Figure 3. If the link is in a state of rate and power allocation (r_j, p_k) at a given instant, and the delivery ratio deteriorates (negative feedback), the reaction can either be to reduce rate or increase power. While increasing power appears to be a natural choice (as in PARF [8]), it is possible that even at the maximum power, the current rate cannot be supported, in which case reducing rate is the right choice. Lack of knowledge of whether a rate can be supported by increasing power to the maximum, can increase the convergence time, which is prohibitive in the presence of mobility. A similar dilemma exists for positive feedback.

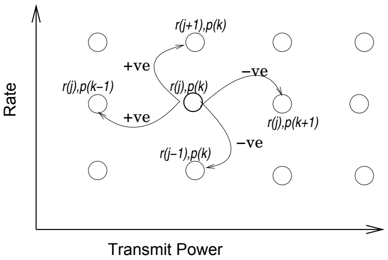
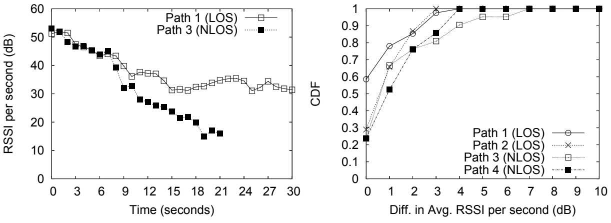


Figure 3: The rate and power adaptation dilemma.



(a) (b)

Figure 4: Effect of mobility (at 0.75 m/s) on RSSI.

Mobility: Adaptive transmit power control becomes more challenging in the presence of user mobility. Link conditions change frequently due to distance-based path loss, short-lived hidden terminals and destructive multipath interference at certain locations in a user's path.

To illustrate and quantify these effects, we conducted an experiment with one moving client, at walking speed (approximately 0.75 m/s), sending voice call traffic (50 packets per second) to a stationary AP. The client moves away from the AP along four different paths in our office building, remaining in line-of-sight (LOS) of the AP on two of the paths, and in non-line-of-sight (NLOS) on the others. Figure 4 (a) shows the average RSSI per second over time for one LOS and one NLOS case. In Figure 4 (b), we show the CDF of the difference in avg. RSSI per second. Although 95% of the time, the avg. RSSI in one second changes by at most 5dB, we also observed changes up to 15db within two seconds.

In such environments, both rate and power control algorithms need to address the following questions effectively: *How frequently* should the adaptation take place? and *At what granularity* should rate and power be adapted? For the first question, solutions have to strike the right balance between reliability and responsiveness: waiting for enough samples avoids reacting to short-lived drops in link conditions, while waiting too long can also be detrimental to performance. For the second question, changing power at coarse granularity allows adapting less frequently, but compromises on battery life and spatial reuse; whereas fine granularity changes require frequent adaptation.

2.2 Existing Solutions and Drawbacks

Transmit power control and rate adaptation are well researched topics in wireless networks. However, our study reveals that no solution provides a comprehensive and easily realizable approach to *simultaneously* address the challenges discussed in the previous section. In particular, our survey across the fields of WLANs, Adhoc, Cellular and Sensor networks research, reveals that (i) there are no existing per-link solutions to address the asymmetric channel access problem in WLANs, and (ii) although approaches exist that address a subset of the challenges, Symphony is the first system to simultaneously address the complete set and provide a solution for mobility-enabled WLANs. Table 2 provides a taxonomy of the relevant related work and identifies their drawbacks. Solutions having deployability constraints due to significant protocol modifications or impractical assumptions such as the requirement of precise interference measurements, are marked by × under 'Deployability'. Solutions not realized in practice and hence not addressing system-level challenges are marked by × under 'Realization'. Remaining columns provide information on the granularity of each solution, its objective (reduced energy consumption or increased capacity or both), and whether it addresses receiver-side interference, asymmetric channel access, and jointly adapting rate.

Several per-link solutions in the adhoc domain address channel access asymmetry (Table 2). However, the solutions are not applicable in the WLAN domain because of their assumption that any node is allowed to communicate with any other node in range; whereas in WLANs, clients can only communicate with the associated AP, and vice versa. Further, these solutions require knowledge about the source of interference, which is impractical to identify at fine timescales in WLANs with mobility (hence marked by × under 'Realization'). Among per-cell [31] and per-network [35] solutions, COMPOW [35] addresses the asymmetric channel access problem by making every node in an ad-hoc network transmit at a common optimum power. In WLANs, Mhatre et. al. [31] propose maintaining symmetry by jointly tuning the CCA threshold and transmit powers to maximize network-wide throughput for a *given placement* of APs and clients. However, these solutions do not lend themselves well to be extended for the per-link case for two reasons: (a) they take a long time to converge after a topology change (e.g., Mhatre et. al's [31] solution takes 30 seconds in one of their experiments), which is unacceptable in the presence of user mobility, and (b) they operate many links in the network at a significantly higher transmit power than necessary since, in either case, the weakest link determines the transmit power. In general, we note that per-cell approaches have to necessarily degenerate into per-link approaches to efficiently deal with mobility. Finally, we identify the lack of a solution for this problem in approaches proposed for other domains such as sensor networks ([49, 20, 29, 22]) or CDMA cellular networks ([17, 51, 13, 18]) mainly because they rely on different multiple access techniques.

With regards to joint rate and power adaptation for WLANs, till date, we are aware of techniques that address only a subset of the problems, and have mostly been implemented in simulators [38, 12, 8]. In contrast, Symphony is the first effort to implement a complete system. The use of transmit power to mitigate the hidden terminal problem is also novel as compared to the use of RTS/CTS messages.

We acknowledge that the related work discussed here is by no means complete and refer the reader to our survey in [40] for more details.

3. DESIGN AND IMPLEMENTATION

This section describes the design and implementation of Symphony, a synchronous two-phase rate and power control framework to increase battery life of mobile devices and improve spatial reuse, while addressing several challenges with adaptive transmit power control.

Domain	Solution	Granularity	Realization	Deployability	Objective		Rate Adaptation	Channel Access Asymm.	Hidden Nodes
					Energy	Capacity			
WLANs	[Sheth02] [45, 46]	Per-link	√	√	√	√	√	×	×
	MiSer [38]	Per-link	×	√	√	×	√	×	√
	PARF, PERF [8]	Per-link	√	√	×	√	√	×	×
	[Chevillat05] [12]	Per-link	×	√	√	√	√	×	×
	[Mhatre07] [31]	Per-cell	√	√	×	√	×	√	×
Adhoc Networks	PCMA [32]	Per-link	×	×	×	√	×	×	√
	BASIC, PCM [23]	Per-link	×	√	√	×	×	×	√
	PCDC [33]	Per-link	×	×	√	√	×	×	√
	POWMAC [34]	Per-link	×	×	√	√	×	×	√
	SHUSH [47]	Per-link	×	×	√	√	×	√	√
	PRC [27]	Per-link	×	×	√	√	√	×	×
	TACP [44]	Per-link	×	×	√	√	×	√	√
	COMPOW [35]	Per-network	√	√	×	√	×	√	√

Table 2: Taxonomy of existing transmit power control algorithms.

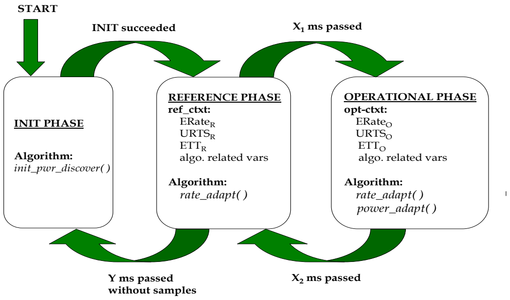


Figure 5: Two-phase synchronous strategy.

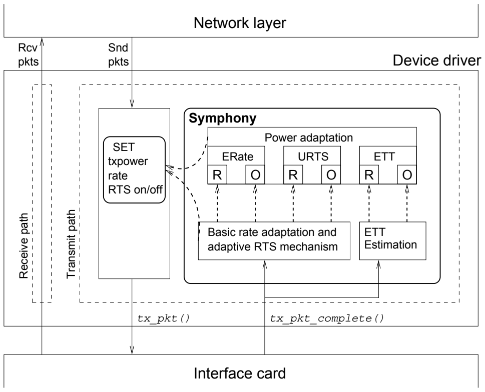


Figure 6: Architecture of Symphony. The blocks R and O represent REF and OPT contexts.

3.1 Overview

Due to the possible adverse effects of power control, the goal of Symphony is to control the transmit power and rate of each link in a WLAN such that *the link's performance is at least as good as in the baseline maximum-power network*. At the core of Symphony is a synchronous two-phase execution (Figure 5) strategy, in which all nodes (APs and clients) in the WLAN cycle through two phases in synchrony—the REFERENCE (REF) phase and the OPERATIONAL (OPT) phase. In the REF phase, Symphony estimates for each link the best achievable performance, and in the OPT phase, it tunes the link to the lowest transmit power to achieve the same performance as in the REF phase. Due to mobility (of users or the environment) the best attainable performance may continuously change, and a power/rate setting may suddenly be affected by asymmetry. The reference phase provides a convenient solution to periodically verify that power and rate control have not unnecessarily degraded system performance.

Similar to most rate control algorithms, Symphony executes on each unidirectional $(sender, receiver)$ link at the sender side. The sender can be either an AP or a client, and the receiver a client or an AP. In the REF phase, each sender performs rate adaptation for each link at the maximum power to choose the best data rate for the current channel conditions. In the OPT phase, the sender performs both rate and power adaptation. The rate and power adaptation algorithms maintain two contexts—ref_ctxt and opt_ctxt, one for each phase for each link. Each context contains several performance metrics and other variables needed for executing the rate and power adaptation algorithms. We choose three metrics—EWMA (Exponential Weighted Moving Average) of data rate, utility of RTS and EWMA of ETT—to help detect and avoid the problems outlined in Section 2. The performance metrics in the ref_ctxt serve as reference values for the OPT phase. In the OPT phase, each link is tuned to the lowest power such that each performance metric in the opt_ctxt is *no worse* than the corresponding metric in the REF phase by more than a threshold.

Before entering the REF-OPT cycle, each link goes through an INIT phase, in which the sender starts from the minimum power level and rapidly discovers the initial power level necessary for communication with the receiver. To avoid affecting applications, we use probe packets at the highest rate and additively increase power for each packet till a probe packet succeeds in reaching the receiver. If we reach the maximum power and still do not succeed, we lower the rate and start over at the lowest power. After succeeding, the sender initializes the OPT phase with the successful power level, and enters the REF phase with the successful rate at the appropriate synchronized time. If the sender is idle for a threshold number of seconds (=2 in our prototype), and then a packet from the network layer arrives that does not succeed in reaching the receiver, we determine that the rate and power information is stale (e.g., due to mobility) and reset the sender to the INIT phase to repeat the rapid discovery process.

We achieve synchronized phase execution on all APs and clients in two steps. First, the APs are synchronized to a

Algorithm 1 RRAA

```
if ($loss > HT_k$) then
    k ← next_lower_rate
else if ($loss < LT_k$) then
    k ← next_higher_rate
end if
```

global real-time clock by a central controller[2]. The controller configures the lengths of the two phases on each AP, and specifies at what point in time the phases should start executing. Second, for each phase change, each AP broadcasts a message (at maximum power) informing the change to the clients, and the clients switch phase. These broadcast messages are sent at high priority to ensure minimum skew across nodes; in our prototype, we use the hardware queue reserved for voice traffic in the Atheros cards.

We implement Symphony in the MadWifi driver 0.9.3.1 [30]. Figure 6 shows the architecture of Symphony. As shown, Symphony executes in the transmit path. We represent rate adaptation as a separate block to make Symphony extensible. Any rate adaptation algorithm can fit into Symphony as long as it executes in the two contexts and provides rate information for power adaptation. Similarly, different mechanisms can be implemented for ETT estimation and determination of RTS utility. In what follows, we describe the important components of Symphony.

3.2 Rate Adaptation

Rate adaptation involves choosing one of several transmission bitrates supported by 802.11a/b/g standards based on the channel conditions. In our context, a rate adaptation algorithm should satisfy at least three requirements: (1) it should be agile to user mobility in typical WLAN environments, (2) it should converge to an appropriate rate for each link rapidly to help with power adaptation, and (3) it should not drop rate due to receiver-side interference, and instead aid the power adaptation algorithm to correct it.

Rate adaptation is a well researched topic in 802.11 wireless networks [11, 19, 21, 24, 28, 30, 37, 50]. Wong et. al. [50] and Ramachandran et al. [41] provide a survey of several rate adaptation algorithms. In this paper, we focus on two state-of-the-art algorithms that we analyze in detail for our purpose: *SampleRate* [11] and *RRAA* [50].

SampleRate: The SampleRate [11] algorithm selects a transmission rate that minimizes the mean packet transmission time. The algorithm maintains an EWMA of expected packet transmission time (ETT) for each rate. For each successfully sent packet, the ETT is updated based on the number of retransmissions, packet length and protocol timing overheads. The algorithm periodically attempts transmission at other data rates and if these sample transmissions indicate lower mean transmission time at other rates, the algorithm switches the rate.

RRAA: This algorithm [50] uses short-term loss estimation of 802.11 frames (in a window of tens of frames) to opportunistically guide rate adaptation. The basic RRAA algorithm works with *high* and *low* thresholds HT_k and LT_k on loss rate at the current data rate k selected; Algorithm 1 depicts its behavior. Further, it uses selective RTS/CTS for avoiding unnecessary rate adaptation in response to collision-induced losses (i.e. receiver-side interference). Briefly, selective RTS/CTS works as follows: on detecting loss of frames[3], RRAA enables RTS/CTS on a selective number of frames. If RTS/CTS reduces the frame loss data rate is not reduced, instead RTS/CTS is increasingly enabled on greater number of frames. Otherwise, RRAA determines that losses are not because of receiver side interference, and reduces rate while disabling RTS/CTS. Note that while turning on RTS/CTS on *all* frames avoids receiver side interference completely, it also reduces the link and network throughput due to protocol overhead and hence, is not generally enabled in practice.

Algorithm 2 RRAA+

```
if ($loss > HT_k$) then
    p[k] /= $\alpha_1$
    k ← next_lower_rate
else if ($loss < LT_k$) then
    for (all rates $j \leq k$) do
        p[j] *= $\alpha_2$
    end for
    if ($rand() < p[next\ higher\ k]$) then
        k ← next_higher_rate
    end if
end if
```

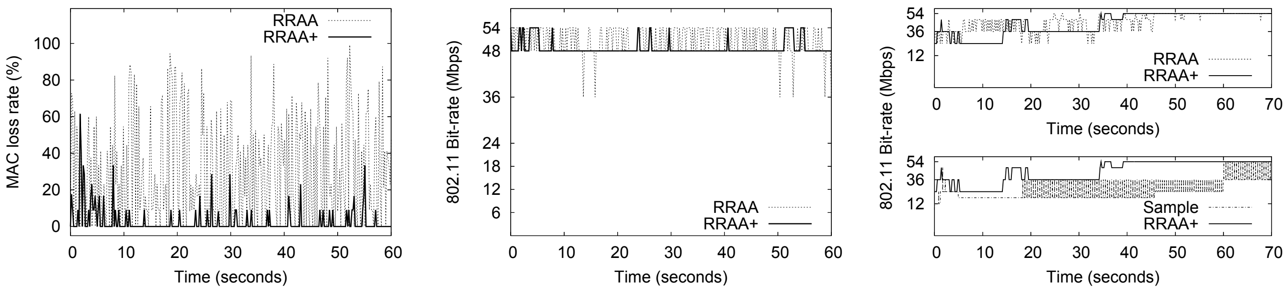
Basing rate adaptation on loss estimation over a window of packets helps a transmitter to not react adversely to one or a few packet losses, which are more common with mobility and power control. The second feature of not reducing rate in response to collision-induced losses is a useful feature [26, 50], and is missing in SampleRate. One drawback of RRAA, however, is that it does not converge to a particular rate, if the next higher rate is inappropriate. For instance, if the 54Mbps rate causes frame loss higher than the high threshold, and 48Mbps causes frame loss lower than the low threshold, RRAA flips between 54 and 48Mbps; ideally the algorithm should converge to 48Mbps.

RRAA+: To make RRAA converge, we propose a fix to the algorithm as shown in Algorithm 2, and call the new algorithm RRAA+. In brief, RRAA+ maintains for each data rate, the probability that it transitions to this rate from the next lower rate. Every time the loss at a rate exceeds the high threshold, the probability of returning to this rate is reduced before transitioning to the next lower rate. Also, every time the loss is lower than LT_k, the probability of the current data rate and the rates below is increased, assuming that all rates below the current rate can also be supported with the current channel conditions. Notice that the probability is only *used* on positive feedback (when the loss is less than LT_k) for transitioning to a higher rate; transitions on negative feedback are deterministic. The MIMD (multiplicative increase, multiplicative decrease) parameters $\alpha 1$ and $\alpha 2$ are chosen such that the algorithm becomes stable. In our prototype, $\alpha 1 = 2$ and $\alpha 2 = 1.0905$; it takes 8 increments to match one decrement.

To compare the performance of the algorithms, we implemented RRAA and RRAA+ in the Linux MadWifi driver. An implementation of SampleRate exists already in the Mad-

[2] The controller and thin-AP architecture is the most common way WLANs are built today.

[3] Frame is at the MAC level, whereas packet is at the driver level. Each packet can lead to multiple frame exchanges due to retransmissions.



(a) Loss rate comparison

(b) Bit-rate for static client

(c) Bit-rate for mobile client

Figure 7: Efficacy of RRAA+ over RRAA.

Wifi package. In our implementation, RRAA and RRAA+ is invoked every 200ms or after 40 packets[4] have been received (defined as an *interval*), and the algorithms use loss rate estimated in the last interval for rate adaptation. We define the interval as above in order to be agile to user mobility, and at the same time receive enough packets to reliably estimate channel conditions. In particular, VOIP calls generate 10 packets in the 200ms interval, which is a reasonable number of samples. We now perform two experiments to demonstrate the benefit of RRAA+ over RRAA and SampleRate.

Experiment 1: We consider one sender and one receiver (two laptops with Atheros PCMCIA cards), with the receiver kept in non-line-of-sight with the sender at a distance of 15 meters. We emulate a VOIP call between the nodes using the DITG traffic generator [10] that generates 50 packets per second with 20 byte payload, emulating a G.729.2 codec. We repeat the experiment with RRAA and RRAA+. Figure 7(a) and (b) show the 802.11 frame loss and the data rate chosen by RRAA and RRAA+. The graphs show that RRAA incurs greater frame loss and hence increased number of retries because of not "learning" that 54Mbps rate is not appropriate for the link. In contrast, RRAA+ learns to avoid 54 Mbps. Note that increased frame loss leads to reduced overall network throughput.

Experiment 2: We again consider one sender and one receiver, but we make the receiver mobile (by placing the laptop in a chair and dragging the chair). We setup the voice call as above again. The receiver starts at a distance of 40 meters and in line-of-sight with the sender, and moves towards the sender to within a meter. The receiver is initially stationary and starts moving 15 seconds after the start of the experiment. We repeat this experiment for RRAA, RRAA+ and SampleRate. In Figure 7(c), we observe that SampleRate takes longer to converge than both RRAA and RRAA+ while RRAA has the same oscillatory behavior as in Figure 7(b). Such conservative rate selection by SampleRate leads to inefficient channel usage and reduces overall network throughput. Further, as shown in [50], SampleRate can incorrectly lower the bit rate in response to collision induced losses.

In summary, we choose RRAA+ over others for rate adaptation in Symphony due to its three features—agility, convergence to appropriate rate and avoidance of rate adaptation because of collision-induced packet losses.

3.3 Power Adaptation

Our goal for power adaptation is to tune each (sender, receiver) link in a WLAN to the lowest transmit power such that the performance metrics in the OPT phase are no worse than the corresponding metrics in the REF phase. Algorithm 3 shows the basic behavior of power adaptation in Symphony. The three conditions in line 1 detect undesirable rate adaptation, receiver side interference and asymmetric channel access introduced by power control. Similar to RRAA+, the power control algorithm learns the lowest appropriate power level by maintaining the probability with which it should transition to a particular level. The algorithm executes once for every two intervals of the rate adaptation algorithm to adapt to user mobility. Further, several rate adaptation intervals can occur in each of the REF and OPT phases, depending on traffic.

Preventing undesirable rate adaptation: For detecting and preventing undesirable rate adaptation due to power control, for each link, the two contexts maintain an EWMA of the rate chosen by the rate control algorithm in response to the measured packet loss: for each $rate_i$ chosen in interval i, we set $ERate = ERate * \psi + rate_i * (1 - \psi)$ at the end of interval i. Everytime the power control algorithm is triggered, if the EWMA of rate in OPT phase ($ERate_O$) is lower than that in the REF phase ($ERate_R$) by a threshold τ_1, transmit power is increased. In our implementation, the EWMA parameter ψ is chosen as 0.8. We choose τ_1 to be 3 Mbps if $ERate_R$ is above 48Mbps or below 24 Mbps, and 6 Mbps otherwise. We make this choice because of the non-uniformity in 802.11 a/g bit rate granularity. Our idea is to place the threshold between the two consecutive rates.

Preventing receiver side interference: To detect that power control introduces receiver side interference we use the adaptive RTS/CTS mechanism. Similar to RRAA [50], RRAA+ includes a mechanism to detect if packet losses are happening due to collisions as opposed to degraded channel conditions. Our implementation of the adaptive RTS/CTS mechanism, however, differs significantly from RRAA; while RRAA was implemented on a per-frame basis (because of the availability of card firmware), we implement it on the basis of a window of packets, both for more reliable estimation of receiver side interference, and for obviating the need of modifying the firmware.

Using this mechanism, we maintain a performance metric—the utility of RTS (URTS)—that is set to one if the loss rate with RTS/CTS is less than the total loss rate in at

[4] Wong et. al [50] observe that 10-40 samples is a good enough number to reliably estimate the channel conditions.

Algorithm 3 power_control

```
1: if (($ERate_R - ERate_O > \tau_1$) OR
       ($URTS_O > URTS_R$)   OR
       ($ETT_O - ETT_R > \tau_2$)) then
2:    p[curpwr] /= $\beta_1$
3:    curpwr ← next_higher_pwr
4: else
5:    for (all pwr levels $i \geq curpwr$) do
6:       p[i] *= $\beta_2$
7:    end for
8:    if ($rand() < p[next\_lower\_pwr]$) then
9:       curpwr ← next_lower_pwr
10:   end if
11: end if
```

Algorithm 4 ETT Estimation

```
VARIABLES mark_seqno, itime, etime

FUNCTION driver_send_pkt (seqno)
if (mark_seqno not set) then
   mark_seqno = seqno
   etime = curtime
end if

FUNCTION card_sent_pkt (seqno)
if (mark_seqno = seqno) then
   ETT = curtime - MAX (itime, etime)
   itime = curtime
   unset mark_seqno
else
   ETT = curtime - itime
   itime = curtime
end if
```

least 2 out of 4 last rate adaptation intervals, i.e. enabling RTS/CTS is helpful to reduce losses. Otherwise, URTS is set to zero. The rationale for waiting for 4 intervals is to determine the utility with greater reliability, while trading off responsiveness; further, unless the receiver interference problem is sustained, we do not increase transmit power and let adaptive RTS address the problem. Now, if URTS is 0 in the REF phase and 1 in the OPT phase, it indicates that power control introduced the receiver side interference that didn't exist in the REF phase. Line 1 in Algorithm 3 captures this condition and triggers a power increase.

Preventing asymmetric channel access: To detect that power control introduces asymmetric channel access, we measure the EWMA of the expected transmission time (ETT) of each packet. The key idea here is that if a sender does not get a chance to transmit as frequently as in REF phase due to asymmetry in the OPT phase, the ETT in the OPT phase increases compared to the REF phase. If the ETT increases by more than a threshold τ_2, we trigger power increase (as in Algorithm 3). In our implementation, $\tau_2 = 100\mu$s.

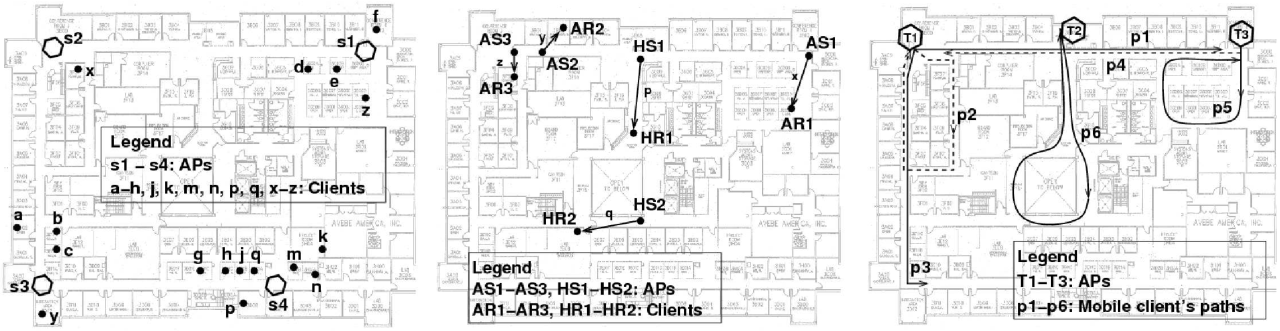
If the interface card provides to the device driver the transmission time for each packet, EWMA of ETT can be easily calculated. However, Atheros cards currently do not provide this information. Further, multiple packets can be queued by the driver in the buffer of the card for efficiency, which makes ETT estimation non-trivial. In our implementation, we overcome the above problem with the two functions in Algorithm 4 that exploit the interaction between the device driver and the interface card. We use the unique sequence numbers in packets that are sent to the card, and keep one outstanding marked packet. We estimate when the packet transmission started using two variables: *etime* that represents the time a marked packet was sent to the card, and *itime* that represents the implicit time when the previous packet transmission was completed. Note that we configure the card to provide per-packet transmission status feedback using `sysctl -w dev.wifi0.txintrperiod=1`. This method ensures that ETT can be estimated even on packets that are buffered back-to-back and hence are not explicitly marked. Further, we only consider packets that do not incur any retransmissions to reliably estimate the channel access delay, and consider packets sent at the same rate as in REF phase to avoid false positives due to rate-based ETT changes.

Finally, since packet sizes affect the ETT because of the varied transmission time on the air, we maintain ETT in terms of 1500 byte packets. For smaller packets, the ETT is scaled to 1500 bytes before using it to calculate EWMA. To do so, we set for each packet $fixedETT = ETT + (1500 - pkt\_size)/rate$, where rate is the data rate used to transmit the packet. We validate experimentally that this approach of scaling ETT is reasonably accurate [40].

Granularity of power control: Learning from our observations in Figure 4, and given that power adaptation gets triggered at least twice in a second, in our implementation, Symphony increases and decreases power at a granularity of 3dB, between [MIN_PLEVEL, MAX_PLEVEL]. This ensures agility to typical user mobility in WLANs. Further, [48] observes that transmit power control at a finer-granularity than 3dB may not always be useful in indoor environments. The minimum and maximum transmit power values can be different on different 802.11 cards, vary with the frequencies used (such as in the 5GHz band) and also vary based on the gains of the external antennas connected to the cards. However, we assume that the levels are discrete at a granularity of 3dB. In our prototype with Atheros cards, we vary the power levels between 0 and 18 dBm.

The process of increasing and decreasing power is similar to rate adaptation in RRAA+. Symphony maintains for each power level, the probability that it transitions to this level from the next higher level. Every time at least one of the conditions on the performance metrics satisfies, the probability of returning to this power level is reduced before transitioning to the next higher power. The MIMD parameters $\beta1$ and $\beta2$ are chosen to make the algorithm stable. In our prototype, $\beta_1 = 3$ and $\beta_2 = 1.14$; it takes 8 increments to match one decrement. Again, the choice of β_1 and β_2 strikes a tradeoff between the benefits of power control and stability of the algorithm; we arrive at the above values after experimenting with several scenarios.

With Symphony's approach of maintaining probability per power level, the transmit power of each sender will eventually converge to a point where the performance of each link is at least as good as in the REF phase. If the performance of a link at a given power level is similar to or better than that of the REF phase the probability of returning to that power level and to any higher power level will converge to 1 due to the multiplicative increase of the probability. Otherwise, the probability of returning to that power level will converge to a small value due to the multiplicative decrease.



(a) Static nodes (b) Problematic scenarios (c) Mobile clients

Figure 8: Experimental setup.

In our implementation, we bound the probability on the lower side to $\frac{1}{64}$ to be responsive to changing channel conditions and mobility.

4. EVALUATION

To demonstrate the achievement of the design goals outlined earlier, in this section, we carry out a systematic and extensive set of experiments, in both controlled and uncontrolled environments. In what follows, we describe the experimental setups and then present our results.

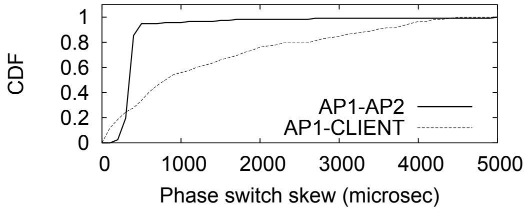


Figure 9: Skew between phases on two APs and an AP and its client (observed by an external monitor).

4.1 Setup

Figure 8 shows our setups for the different experiments. We broadly classify the setups into four categories: (1) *static nodes* (Figure 8(a)) in which clients are randomly placed in different cubicles and office rooms (represented by lower case letters), and APs are placed close to the operational WLAN APs, (2) *problematic scenarios* (Figure 8(b)) in which we carefully choose client and AP locations so as to emulate channel access asymmetry and receiver-side interference, (3) *mobile clients* (Figure 8(c)) in which APs are placed close to operational WLAN APs (the AP is at location T1 for paths p1,p2 and p3, at T2 for path p6, and at T3 for paths p4 and p5), and clients associated with one of the APs move on different paths in the office building, and (4) the indoor ORBIT testbed [42]. For the setup involving client mobility, the rationale behind choosing the different paths is to approximate a reasonable mix of mobility patterns (moving towards/away from an AP), and channel conditions (LOS or NLOS) that occur in real-world situations.

In the first three setups, Symphony APs and clients are Dell laptops, while the ORBIT testbed uses custom-made, small form-factor PCs. All setups use Atheros cards (PCMCIA or mini-PCI), which transmit at a default (maximum) power level of 18dBm[5] for 54Mbps data rate, and allow Symphony to change transmit power at the desired granularity of 3 dB. The APs are connected by a 100Mbps wired network and all nodes use the same 802.11 channel. While both clients and APs can execute Symphony, we perform most of our experiments with APs as senders and clients as receivers. This is because of the current limitation of Atheros cards that do not implement per-packet power control for ACK and CTS packets. To overcome this limitation for proper evaluation, unless specified otherwise, we run Symphony on APs, and disable per-packet transmit power control on the clients. Instead, APs append transmit power information to each outgoing frame, and a slightly modified client driver extracts this information and sets the card to this power level (similar to how iwconfig athN txpower $val sets power). This ensures that all frames, including ACKs are returned at the configured power. Finally, except for the experiments that focus on interactions with Symphony-non-compliant nodes (e.g. legacy WLANs), all experiments are conducted on 802.11a channels to avoid interference from/to our office's WLAN that uses all three 802.11g channels (1,6,11).

For AP synchronization and configuration, instead of using a different central controller, we just use one AP as the master AP, and synchronize the others with the master through NTP on the wired network. Further, for all the experiments in this section, we use 200ms and 800ms as the length of REF and OPT phase resp., and all APs are configured to start the REF phase at the beginning of each one second boundary. The choice of phase lengths is a tradeoff that ensures collecting reliable estimates in the REF phase, while not compromising too much on energy savings [40].

Two-phase Synchronization: To test that Symphony indeed executes synchronously on APs and clients, we first perform a micro-benchmark in which we consider a network of two APs with one client each. In this experiment, both clients and APs run Symphony. We place a laptop with its wireless card in monitor mode, close to the APs. We start two UDP transfers of 200 pkts/sec between each (AP, client) pair and collect packets on the monitor. The packets are appended with the transmit power used to send them and the executing phase (REF/OPT). Figure 9 plots the CDF of skew between two APs and one AP and its client as seen by the monitor. The graph shows that the nodes make corresponding phase transitions within 5 ms of each other.

[5] While the specifications mention maximum transmit power of 15±2dBm, we observe that the card can use up to 18dBm.

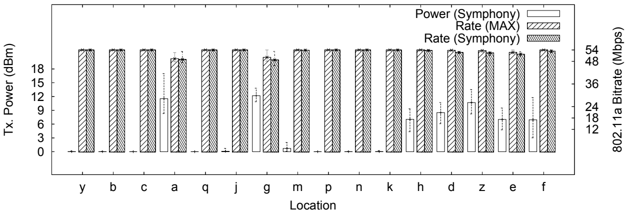
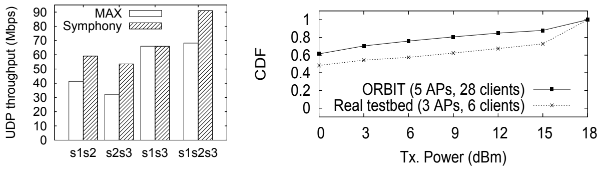


Figure 10: Transmit power reduction for clients in several representative office locations.



(a) (b)

Figure 11: Experiments demonstrating (a) Increased spatial reuse. (b) significant power reduction even in a dense setup.

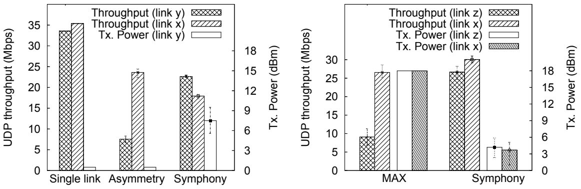
4.2 Results

We now describe our results from several experiments on the above setups to demonstrate Symphony's efficacy.

4.2.1 Static Experiments

Transmit Power Reduction: To get a measure of achievable transmit power reduction in near-typical indoor office environments, we consider all nodes in Figure 8(a). We setup ten 1 minute VOIP calls for each client. From the associated APs, the calls start at different times (separated by 5 minutes) for each client. The white bars in Figure 10 show the average transmit power used by the Symphony APs for each client; in most typical user locations, the required transmit power can be substantially lower than the default 18dBm. Further, the error bars plot the minimum and maximum of the average transmit power per call, showing that the optimal power for maintaining a link's performance can vary with time for even static locations. The other two bars show the rate chosen by Symphony and the rate when transmitting at maximum power. Symphony causes minimum effect on the data rate chosen. *In this setup, for the three cells with S1, S3 and S4 as APs, a per-cell solution would operate all links at the worst client's transmit power, which is about 12 dBm. In contrast,* Symphony *enables 75-80% of the clients in the cells to settle at 3 to 12 dB (i.e. 50% to 94%) lower transmit power than 12 dBm.*

Spatial Reuse: Several studies demonstrate that transmit power control leads to increased spatial reuse [36, 27]. These works show simulation results over large topologies that can not be easily realized in a prototype testbed. Here, we perform a small-scale spatial reuse experiment, where we consider links (S1, z), (S2, x), and (S3, y) as in Figure 8(a) and make a subset of them operate simultaneously. We consider two cases: when the links operate at maximum power and when they operate with Symphony. *Figure 11(a) shows the aggregate throughput of the links, and clearly demonstrates 30-50% increased throughput for different combinations.*



(a) (b)

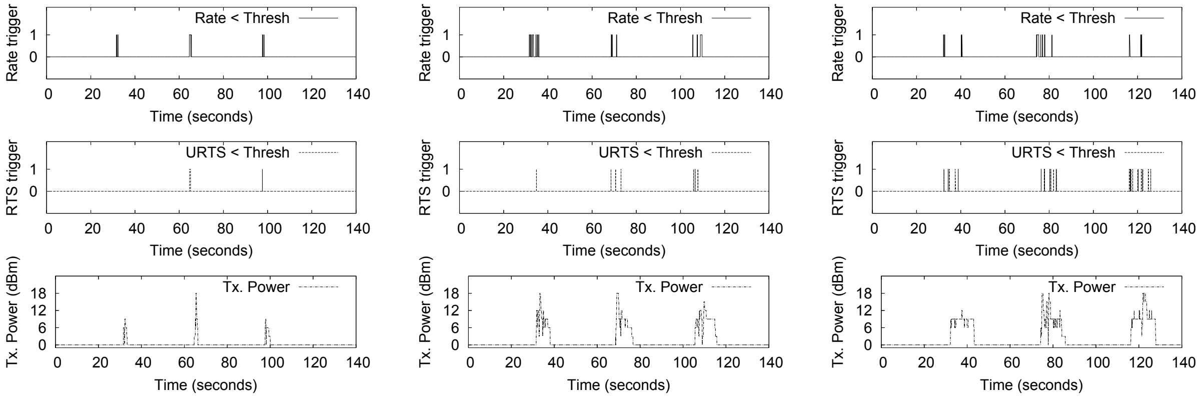
Figure 12: Preventing asymmetric channel access. (a) Symphony's ability to detect and avoid channel access asymmetry, (b) power control removes inherent link asymmetry.

Large-scale experiments: To assess the effect of high node density on Symphony, we emulate dense deployments on the indoor ORBIT testbed [42] with 5 APs and 28 clients, and on our office testbed with 3 APs and 6 clients (placed close to s4 in Figure 8(a)). The inter-AP distance is 5 meters on ORBIT and 15m in our office testbed. Clients are within 15 meters of each AP in both cases. We setup bi-directional traffic between each client and its associated AP, and enable Symphony on all nodes. Figure 11(b) shows the CDF of the average transmit powers used by clients in each second, over a period of 60 seconds. We observe that Symphony *enables clients to settle at much lower power levels in both sets of experiments. For instance, in the ORBIT experiment, clients settle at transmit power of 0dBm over 60% of the time and within 9dBm over 80% of the time.*

4.2.2 Problematic Scenarios

Avoiding Channel Access Asymmetry: To demonstrate that Symphony avoids channel access asymmetry by intelligently adapting the transmit power of a link, we perform two sets of experiments. In the first set, we consider links x and y as in Figure 8(b) (where AS1 and AS2 are senders and AR1 and AR2 are receivers), and setup backlogged UDP traffic to measure the link throughput. Link x always operates at the maximum power. We consider several cases—(i) links x and y running one-at-a-time with y running Symphony, (ii) links x and y running simultaneously with y at a fixed 0 dBm, and (iii) links x and y running simultaneously with y running Symphony. Figure 12(a) shows the results over ten runs. In case (i) Symphony enables link y to operate at 0 dBm and still achieve full throughput. If link y is operated at 0 dBm together with link x, however, the throughput of link y drops significantly compared to x, as shown in case (ii) due to asymmetric channel access. We validate this by observing the difference in ETTs observed by links x and y. In case (iii), we show that Symphony increases the transmit power of link y to between 6 and 9 dBm to avoid asymmetry. Observe that link y didn't have to operate at the maximum power to let link x perceive its transmission.

In the second set, we show that when there is inherent asymmetry between two links at default power, power control is beneficial in removing it and increasing the throughput and fairness of links. As in Figure 8(b), a sender at location AS3 gets significantly affected when running in conjunction with a sender at AS1. For this experiment, we use links x and z and consider two cases: (i) both x and z running together at maximum power, and (ii) x and z running together with Symphony. Figure 12(b) shows results aver-



(a) 1 Second (b) 5 Seconds (c)10 Seconds

Figure 13: Efficacy in avoiding receiver side interference, which is introduced for 1-, 5-, and 10-second durations at different times. Utility of RTS and/or a decrease in rate trigger an increase in transmit power.

aged over ten runs. The graph shows that when operating at maximum power, link z gets significantly lower throughput than link x. When running x and z with Symphony, both links achieve greater throughput because they are able to operate independently at the lower transmit powers, thereby also demonstrating increased spatial reuse due to power control. *In both sets of experiments,* Symphony *increases the throughput of asymmetry-affected links by three times.*

Avoiding Receiver Side Interference: To show that Symphony is effective in addressing receiver-side interference in mobile environments, we consider links p and q in Figure 8(b). Link q operates at maximum power, whereas link p operates with Symphony. The setup is such that link p operates at 0 dBm when run individually, HS1 and HS2 share the channel at maximum power, and HS2 does not perceive transmission on link p if p operates at 0 dBm and hence destroys packets at HR1 (i.e. causes receiver side interference). In each run, we start a 20 Mbps UDP transfer on link p for 3 minutes, and start a 5 Mbps transfer on link q for a short periods (1 second, 5 seconds and 10 seconds) of time at different times during the 3 minutes.

The bottom graphs in Figure 13 show that in response to link q's entry and exit, Symphony on link p increases and decreases transmit power respectively to avoid the adverse effect of receiver side interference. The graph shows that Symphony *is responsive to receiver side interference even at short timescales of 1 second.* The top two graphs in each of the cases (a), (b) and (c) show the rate and RTS triggers that identify the condition that triggered increased power as in line 1 of Algorithm 3. We note that the utility of RTS is not always sufficient to detect receiver side interference, primarily because even RTS is sent at the chosen (lower) transmit power with Symphony, which may not be perceived by HS2. In such a case, rate drops and leads to increased transmit power, thereby letting RTS reach HS2. Recall that if RTS is useful, it reduces unnecessary reduction in rate.

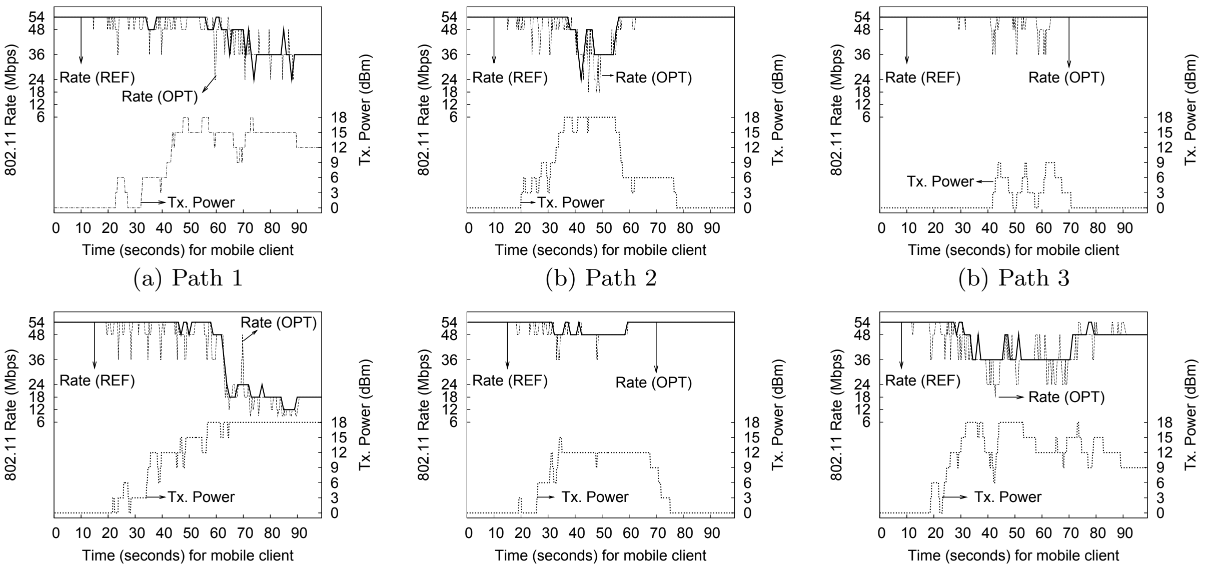
4.2.3 Client Mobility

To demonstrate Symphony's agility to client mobility, we consider three AP locations and six client paths as shown in Figure 8(c). On each of the paths, the client is mobile at a speed of 0.75m/s. We again setup VOIP calls from the AP to the client. Figure 14(a) shows for path p1 that moving the client away starts affecting the bitrate in the OPT phase, and hence Symphony increases transmit power to maintain the bitrate to the same level as in the REF phase. As the client moves farther, even the rate in REF phase falls. Figure 14(b) shows that moving the client away on path p2 makes Symphony increase transmit power, but it also reduces the power when the client returns to the AP location. *Overall,* Symphony *opportunistically enables a link to operate up to 18dB lower than the default.* Similar observations can be made on other paths.

For the same experiment, Figure 15 shows the application level loss rate for Symphony in comparison to using default maximum transmit power, and the difference in R-score. R-score is a popular performance metric for the quality of voice calls [14]. An R-score of 70 or more is considered good voice quality. While R-score depends on several factors [14], in brief, the difference in R-score can be simplified to $40 \times (log(1 + 10e_m) - log(1 + 10e_s))$, where e_m and e_s represent the loss rate with maximum power and Symphony respectively. The graph shows that Symphony incurs little extra impact on application level loss on all the paths. Further, *in the worse case (path 4), the R-score using* Symphony *deteriorates only by 3.4 and the average R-score deterioration using* Symphony *is 2.* While the actual R-score also depends on the end-to-end delay, we note that with even the maximum loss rate (as in path 4) using Symphony, in order to achieve R-score as low as 70, the acceptable end-to-end delay is over 300ms. Further, in-depth analysis of the losses shows that significant part of the loss occurs when the client is far away from the AP. In a real mobility enabled WLAN, mobile clients handoff to closer APs for better quality, and hence we believe that even this application-level loss will not occur in practice.

4.2.4 Experiments with non-compliant nodes

We now consider the case when Symphony is incrementally deployed in a setting where it has to operate in conjunction with non-compliant nodes such as other legacy WLANs. We first investigate whether transmit power reduction can be achieved in this situation. We consider the setup (Figure 8(a)) with s3 as the AP location, and a,b,c,y as four client locations, with s3 running Symphony. We switch to



(a) Path 1 (b) Path 2 (b) Path 3

(a) Path 4 (b) Path 5 (b) Path 6

Figure 14: Adaptation to mobility: Symphony's behavior of rate and power in different paths.

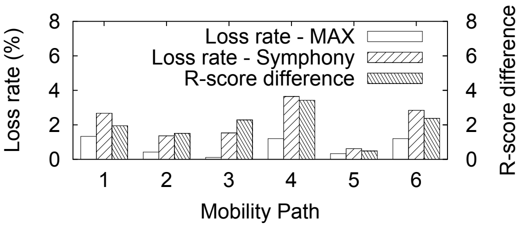


Figure 15: Application loss rate with mobility.

channel 6 in 802.11g, on which there are other active APs and clients from our office WLAN. By observing beacons at the location s3, we determine that there are at least 13 non-compliant APs on channel 6.

For each client, the Symphony AP makes a 2 Mbps transfer to every client every half an hour, to estimate the power level appropriate for the clients. We perform this experiment over a period of 12 hours mostly during regular office hours when the network is active. Figure 16(a) shows the CDF of the average transmit power per client in each run, across several runs, for four clients. The graph shows that Symphony *opportunistically reduces the transmit power on all links even when operating in conjunction with a non-*Symphony*-compliant network.* For instance, links (s3, c) and (s3, y) operate 6dB lower than the default transmit power 85% of the time.

We also study the effect of Symphony on non-compliant nodes in a controlled environment by conducting experiments on the ORBIT testbed. In this setup, we consider 2 APs and 11 clients using Symphony, and 2 APs and 13 clients are in legacy mode (using maximum transmit power and SampleRate for rate adaptation). For the legacy clients, we emulate realistic WLAN traffic (as described in [16]), while using two-way VOIP for the Symphony clients. We compare the average packet error rate (PER) for legacy clients with and without transmit power control on the Symphony part of the network. Figure 16(b) shows the CDF of average PER per client. The graph shows that Symphony nodes have a positive effect of reducing PER for the legacy clients. Since the legacy clients use higher transmit power than the Symphony nodes, when power control is enabled, their transmissions more likely benefit from the *capture effect*, which in turn reduces the PER. Reduced PER results in the usage of higher MAC bit-rates, as shown in Figure 16(c). These results demonstrate Symphony's incremental deployability.

4.2.5 Potential Battery Life Improvements

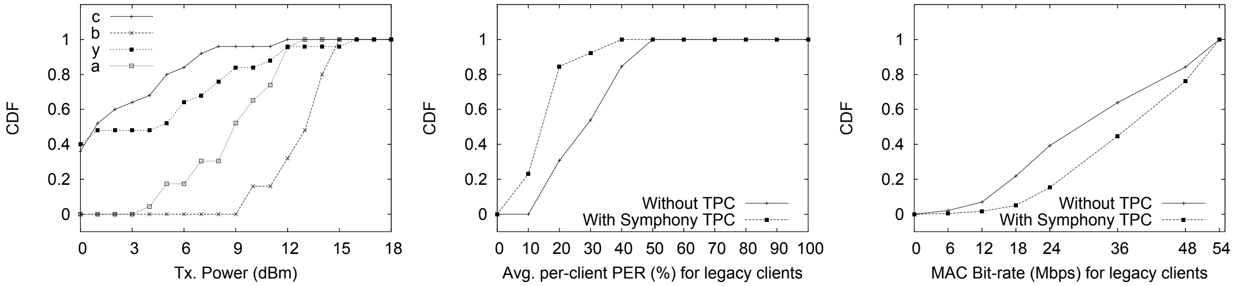
Due to lack of hardware support for measuring the exact energy savings, we choose the analytical model in [38] to demonstrate the potential energy benefits of Symphony. According to the model given by Equations 17 and 41 in [38], the power dissipated during transmission (P_d) can be represented in terms of the transmit power (P_t) and the common power (P_c) that is consumed by the circuitry independent of the transmit power, as

$$P_d = P_c + \frac{P_t}{0.02 \times 5^{[\frac{2}{3} \times log_{10}(P_t)]}} \tag{1}$$

where all quantities are in milliwatts. Using the above equation and power consumption values from Broadcom's BCM 4328 card [1], we obtain $P_c = 305mW$. Using this value and the above equation, we can calculate the power dissipated at any transmit power.

Now, as an example, in the experiment corresponding to Figure 11(b), using the CDF and the power dissipated by clients at each transmit power level, we can calculate the average power dissipated by Symphony clients as $P_d^S = 429mW$ for the ORBIT setup and $P_d^S = 497mW$ for the indoor office setup. In contrast, transmitting at the maximum transmit power of 18dBm would lead to a power dissipation of $P_d^M = 766mW$.

Since battery life is inversely proportional to the power dissipated, assuming that a card transmits and receives for equal amounts of time (and ignoring MAC protocol effects for simplicity and illustrative purposes), the active mode battery life improvement can be approximated as follows.



(a) Transmit power reduction (b) PER reduction (c) MAC bitrate increase

Figure 16: Interaction with an operational network that does not run Symphony.

At maximum transmit power, if $\mathcal{L}_M$ is the battery life, and P_r is the power dissipated during reception, then

$$\mathcal{L}_M = \frac{K}{P_r + P_d^M + C} \tag{2}$$

where K is a proportionality constant and C represents the power consumed by the other components of the mobile (such as the processor, the graphics display, etc.) respectively. Similarly, if $\mathcal{L}_S$ is the battery life with Symphony, then, $\mathcal{L}_S = \frac{K}{P_r + P_d^S + C}$. Then, the active mode battery life improvement is $\frac{\mathcal{L}_M - \mathcal{L}_S}{\mathcal{L}_M}$. Substituting the P_d values calculated earlier and using the value for $P_r = 295mW$ for the BCM 4328 card [1], we determine that the active mode battery lifetime improvement with Symphony over using default transmit power can be up to 46% for the ORBIT setup and up to 33% for the office setup (assuming that C is negligible). Compared to a per-cell power control solution [31], for Figure 10, the improvement for each client can range from 0 to up to 26% with average being up to 17%. The higher the value of C, the lower will be the benefits of transmit power control. However, we envision that technology improvements will only reduce the value of C [43].

5. DISCUSSION AND LIMITATIONS

While Symphony's per-link power control improves network throughput, its greedy approach may not achieve *maximum* network throughput; a network wide optimization problem considering traffic profiles, locations of users, channel conditions for all users, etc. may lead to higher network throughput, while compromizing some users. However, such network-wide optimizations are also much harder to realize in practice, especially with user mobility.

In this paper, we assume that clear channel assessment threshold (CCA) is fixed, mainly because commercial hardware does not openly allow modifying it to avoid its potential abuse. However, Symphony's transmit power control is complementary to CCA tuning and can be easily extended to incorporate CCA tuning. In general, Kim et al. [27] argue that fixing one parameter and tuning the other gives the same effect, although fixing CCA threshold and varying transmit power has added benefits.

While Symphony does not need modifications to the MAC protocols, it does require synchronization that is currently not a common feature in WLANs. However, Symphony only requires loose synchronization that can be easily achieved, as we do in our implementation. On the implementation side, our AP-client synchronization is currently susceptible to packet losses. If the broadcast packet from the AP for switching phase is lost, a client will not switch phase and affect Symphony functionality. However, this can be easily fixed by utilizing the client's local clock to switch phase at the appropriate time if the broadcast message is not received within a tolerance, much like how NTP [4] functions.

Finally, the success of Symphony depends on the prevalance of per-packet transmit power control feature in wireless cards. While this feature, to our knowledge, is only supported by Atheros cards today, we believe that with the increasing popularity of mobile devices and the growing need to improve battery life, other manufacturers will also eventually provide the above feature.

6. CONCLUSION

Per-link adaptive transmit power control in 802.11 WLANs faces the following challenges: (1) receiver-side interference and asymmetric channel access, (2) incorrect data rate selection, and (3) mobility-induced channel variations at short timescales. In this paper, we propose Symphony—a novel synchronous two-phase rate and power control system, whose agility in adaptation enables us to systematically address the three challenges, while increasing 802.11 network capacity and battery life of mobile devices. Through a detailed prototype study, we conclude that Symphony is (1) effective in achieving the goals set-forth, (2) easy to realize in a WLAN, (3) readily deployable even in the presence of non-compliant nodes, while increasingly providing power control benefits as more nodes adhere to Symphony's strategy.

7. ACKNOWLEDGMENTS

We thank our shepherd, Ranveer Chandra, and the anonymous MobiSys reviewers for their feedback on this paper. We also thank Mesut Ergin for sharing his setup to emulate realistic WLAN traffic on the ORBIT testbed.

8. REFERENCES

[1] http://www.broadcom.com/products/Wireless-LAN/802.11-Wireless-LAN-Solutions/BCM4328.

[2] Cisco Aironet 350 Series Access Points. http://www.cisco.com/univercd/cc/td/doc/product/wireless/airo_350/accsspts/ap350hig/ap350ch2.htm.

[3] Infonetics research. http://www.infonetics.com/.

[4] Network time protocol. http://www.ntp.org.

[5] Vo-Fi and dynamic power control: A possible mismatch? http://tinyurl.com/323at6.

[6] Where are the Wi-Fi phone power-save capabilities? http://tinyurl.com/3bcn4x.

[7] Atheros breaks the power barrier with new wi-fi solution for mobile devices. http://atheros.com/news/AR6002.htm, October 2007.

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

[9] W. Alliance. Wmm power save for mobile and portable wi-fi certified devices. Technical report, 2005.

[10] S. Avallone, S. Guadagno, D. Emma, A. Pescape, and G. Ventre. D-ITG distributed internet traffic generator. In *Proc. QEST 2004*, pages 316–317, September 2004.

[11] J. C. Bicket. Bit-rate selection in wireless networks. Master's thesis, M.I.T., Cambridge, MA, February 2005.

[12] P. Chevillat, J. Jelitto, and H. L. Truong. Dynamic Data Rate and Transmit Power Adjustment in IEEE 802.11 Wireless LANs. *Intl. Journal of Wireless Information Networks*, 2005.

[13] M. Chiang and J. Bell. Balancing supply and demand of wireless bandwidth: joint rate allocation and power control. In *IEEE INFOCOM*, March 2004.

[14] R. Cle and J. Rosenbluth. Voice over IP performance monitoring. *ACM Comp. Commn. Rev.*, 31(2), April 2001.

[15] R. Draves, J. Padhye, and B. Zill. Routing in multi-radio, multi-hop wireless mesh networks. In *MobiCom*, 2004.

[16] M. A. Ergin, K. Ramachandran, and M. Gruteser. Extended Abstract: Understanding the Effect of Access Point Density on Wireless LAN Performance. In *MOBICOM*, 2007.

[17] G. J. Foschini and Z. Miljanic. A simple distributed autonomous power control algorithm and its convergence. *IEEE Trans. on Vehicular Technology*, April 1993.

[18] P. Hande, S. Rangan, and M. Chiang. Distributed uplink power control for optimal sir assignment in cellular data networks. In *INFOCOM*, 2006.

[19] I. Haratcherev, K. Langendoen, R. Lagendijk, and H. Sips. Hybrid rate control for IEEE 802.11. In *Intl. Workshop on Mobility Management and Wireless Access*, 2004.

[20] T. He, S. Krishnamurthy, J. A. Stankovic, T. Abdelzaher, L. Luo, R. Stoleru, T. Yan, L. Gu, J. Hui, and B. Krogh. Energy-efficient surveillance system using wireless sensor networks. In *MobiSys*, 2004.

[21] G. Holland, N. Vaidya, and P. Bahl. A rate-adaptive MAC protocol for multi-hop wireless networks. In *Proc. of ACM MOBICOM*, Rome, Italy, 2001.

[22] J. Jeong, D. E. Culler, and J.-H. Oh. Empirical analysis of transmission power control algorithms for wireless sensor networks. In *IEEE INSS*, 2007.

[23] E.-S. Jung and N. H. Vaidya. A power control mac protocol for ad hoc networks. In *MobiCom*, 2002.

[24] A. Kamerman and L. Monteban. WaveLAN-II: A high-performance wireless LAN for the unlicensed band. *Bell Labs Technical Journal*, pages 118–133, Summer 1997.

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

[26] J. Kim, S. Kim, S. Choi, and D. Qiao. CARA: Collision-aware rate adaptation for IEEE 802.11 WLANs. In *INFOCOM*, 2006.

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

[28] M. Lacage, M. H. Manshaei, and T. Turletti. IEEE 802.11 rate adaptation: a practical approach. In *MSWiM*, 2004.

[29] S. Lin, J. Zhang, G. Zhou, L. Gu, J. A. Stankovic, and T. He. ATPC: adaptive transmission power control for wireless sensor networks. In *SenSys*, 2006.

[30] MadWifi. Multiband Atheros Driver for WiFi. http://www.madwifi.org/.

[31] V. Mhatre, K. Papagiannaki, and F. Baccelli. Interference mitigation through power control in high density 802.11 wlans. In *Proc. of IEEE INFOCOM*, 2007.

[32] J. Monks, V. Bharghavan, and W.-M. Hwu. A power controlled multiple access protocol for wireless packet networks. In *INFOCOM*, 2001.

[33] A. Muqattash and M. Krunz. Power controlled dual channel (pcdc) medium access protocol for wireless ad hoc networks. In *Proc. of IEEE INFOCOM*, pages 470–480, 2003.

[34] A. Muqattash and M. Krunz. A single-channel solution for transmission power control in wireless ad hoc networks. *IEEE JSAC*, 2005.

[35] S. Narayanaswamy, V. Kawadia, R. S. Sreenivas, and P. R. Kumar. Power control in ad hoc networks: Theory, architecture, algorithm and implementation of the compow protocol. In *Proc. of European Wireless Conference*, 2002.

[36] V. Navda, R. Kokku, S. Ganguly, and S. Das. Slotted Symmetric Power Control in managed WLANs, http://www.nec-labs.com/∼ravik/RESEARCH/contour.pdf. Technical report, NEC Laboratories America.

[37] J. Pavon and S. Choi. Link Adaptation Strategy for IEEE 802.11 WLAN via Received Signal Strength Measurement. In *Proc. of IEEE ICC*, May 2003.

[38] D. Qiao, S. Choi, A. Jain, and K. G. Shin. Miser: an optimal low-energy transmission strategy for ieee 802.11a/h. In *ACM MobiCom*, pages 161–175, 2003.

[39] D. Qiao, S. C. A. Jain, and K. G. Shin. Adaptive transmit power control in IEEE 802.11a wireless LANs. In *IEEE VTC (Spring)*, volume 1, pages 433–437, April 2003.

[40] K. Ramachandran, R. Kokku, H. Zhang, and M. Gruteser. Synchronous Two-phase Rate and Power Control in 802.11 WLANs. http://www.nec-labs.com/∼ravik/RESEARCH/symphony-tech.pdf.

[41] K. Ramachandran, H. Kremo, M. Gruteser, P. Spasojevic, and I. Seskar. Scalability analysis of rate adaptation techniques in congested ieee 802.11 networks: An orbit testbed comparative study. In *WoWMoM*, 2007.

[42] D. Raychaudhuri, I. Seskar, M. Ott, S. Ganu, K. Ramachandran, H. Kremo, R. Siracusa, H. Liu, and M. Singh. Overview of the ORBIT radio grid testbed for evaluation of next-generation wireless network protocols. In *Proc. of IEEE WCNC*, March 2005.

[43] C. W. S. Gaudin. Intel unveils new low-power chip/building block. http://tinyurl.com/4wkxdr.

[44] V. Shah, E. Gelal, and S. V. Krishnamurthy. Handling asymmetry in power heterogeneous ad hoc networks. *Comput. Networks*, 51(10):2594–2615, 2007.

[45] A. Sheth and R. Han. A Mobility-Aware Adaptive Power Control Algorithm For Wireless LANs. In *IEEE CAS Low Power Workshop*, 2002.

[46] A. Sheth and R. Han. An Implementation of Transmit Power Control in 802.11b Wireless Networks. Technical report, Dept. of Comp. Science, Univ. of Colorado, 2002.

[47] A. Sheth and R. Han. Shush: Reactive transmit power control for wireless mac protocols. In *WICON*, 2005.

[48] V. Shrivastava, D. Agarwal, A. Mishra, S. Banerjee, and T. Nadeem. Understanding the limitations of transmit power control for indoor wlans. In *IMC*, 2007.

[49] D. Son, B. Krishnamachari, and J. Heidemann. Experimental study of the effects of transmission power control and blacklisting in wireless sensor networks. In *Proc. of IEEE SECON*, 2004.

[50] S. H. Y. Wong, S. Lu, H. Yang, and V. Bharghavan. Robust rate adaptation for 802.11 wireless networks. In *ACM MOBICOM*, 2006.

[51] M. Xiao, N. B. Shroff, and E. Chong. A utility-based power control scheme in wireless cellular systems. *IEEE/ACM Trans. on Networking*, 11(2):210–221, 2003.