

A Novel Rate Adaptation Scheme for 802.11 Networks

Yanxia Rong, Amin Y. Teymorian, Liran Ma, Xiuzhen Cheng, and Hyeong-Ah Choi

Abstract—We develop a rate adaptation scheme for 802.11 networks termed SRC that is based on a novel combination of sequential hypothesis testing and short-term loss ratios. The underlying design of SRC represents a fundamental departure from previous rate adjustment strategies that employ fixed-size observation windows and seemingly-intuitive rate increase guidelines. Through extensive simulation results, SRC is shown to be robust to dynamic characteristics of the wireless channel and to significantly increase network throughput over existing mainstream rate adaptation solutions. SRC does not require any type of specialized hardware or modifications to the existing 802.11 standard.

Index Terms—Rate adaptation, IEEE 802.11 networks, sequential hypothesis testing.

I. INTRODUCTION

THE ubiquitous deployment of Wi-Fi network infrastructure has made 802.11 networks a significant part of today's Internet access network. In their underlying standard, multiple PHY layer transmission rates (e.g., 6 Mbps, 18 Mbps, 24 Mbps, and 54 Mbps) are provided. Although this multi-rate capability is critical to system performance, there is no dynamic rate adaptation (or "control") mechanism specified by the standard.

Existing proposals for rate adaptation mechanisms rely on strategies that can be broadly categorized as either physical layer measurement or frame-based estimation. In physical layer measurement, information from the PHY layer is used to directly measure the channel quality. Such information includes the received signal-to-noise ratio (SNR). The intuition is that a larger SNR implies a higher probability of receiving data with a low bit error rate (BER). This approach appears to have the potential to significantly improve the network throughput because the channel quality is directly represented in the rate selection process.

However, it has been recently reported that SNR measurements have little value in predicting loss rates [1]. Additionally, SNR variations have been demonstrated to notably degrade the accuracy of rate estimation [2]. Although long-term operations on PHY layer metrics have been proposed to alleviate this instability, this does not necessarily lead to higher throughput when compared to short-term estimation

methods. One explanation for this phenomenon is that long-term channel observation fails to accurately capture the short-term (at a scale of hundreds of milliseconds) opportunistic gain in the wireless channel.

Since adapting rates based on direct channel measurement is difficult to perform and in violation of the 802.11 standard, frame-based estimation becomes an attractive alternative. The frame-based estimation techniques make decisions on rate increase or decrease based on the success or failure, respectively, of packets within a fixed-size window. The window may consider back-to-back observations (e.g., ARF's consecutive successes [3]) or cumulative observations (e.g., RRAA's percentage of successful packets[2]) of data or probe packets. Nevertheless, frame-based estimation algorithms are plagued by an inherent limitation in their decision flexibility: too small of a window size results in over sensitivity to channel dynamics, while too large of a window lacks responsiveness to sudden changes in channel condition. The effect of this inflexibility includes, for example, unnecessary throughput degradation for small bursts of noise and slow rate decrease decisions in the presence of mobility induced error, respectively. Additionally, as illustrated in this paper, seemingly-intuitive guidelines such as increasing the transmission rate based on the number of successful transmissions can result in significant performance penalties in many practical scenarios. Further exacerbating these issues is the presence of channel dynamics such as signal fading, random channel errors, contention from hidden terminals, and mobility induced channel variations. Although recent work has mitigated some channel dynamics (e.g., interference from hidden terminals [4]), rate adaptation remains a non-trivial problem.

With the challenges associated with short and long-term measurement in mind, we design a Sequential hypothesis testing based Rate Control scheme termed SRC that maintains robustness to channel dynamics. It adjusts transmission rates adaptively and increases network throughput significantly. SRC decides whether to switch rates based on test results of the hypothesis that the throughput at another rate is greater than the throughput at the current rate, without relying on any fixed-size windows. It also employs a self-adaptive feedback mechanism that assists with rate increase decisions in a variety of a common scenarios. Extensive simulation results attest to SRC's agility in exploiting evanescent opportunistic gains in the wireless channel, and its ability to outperform existing mainstream rate control solutions by as much as 59% in some scenarios. The remainder of this paper is organized as follows. In Section II, we review the literature and related research efforts that we build upon in this paper. A brief

Manuscript received October 26, 2007; revised March 6, 2008 and July 25, 2008; accepted August 3, 2008. The associate editor coordinating the review of this paper and approving it for publication was J. Zhang.

Y. Rong, A. Y. Teymorian, X. Cheng, and H.-A. Choi are with the Department of Computer Science, The George Washington University, Washington DC, 20052 USA (e-mail: {yxrong, amin, cheng, hchoi}@gwu.edu).

L. Ma is with School of Technology, Michigan Technological University, Houghton, MI 49931 USA (e-mail:lrma@mtu.edu).

Digital Object Identifier 10.1109/TWC.2009.071196

introduction to SRC's underlying statistical theory, sequential hypothesis testing, follows in Section III. Our proposed rate adaptation scheme is elaborated in Section IV, and we evaluate its performance using extensive OPNET [5] simulations in Section V. Finally, some concluding remarks and future research directions are discussed in Section VI.

II. RELATED WORK

One of the earliest rate control algorithms is Auto Rate Fallback (ARF) [3]. In this scheme, a probe packet is sent upon either 10 consecutive transmission successes or expiration of a timer. If the probe packet succeeds, ARF increases the transmission rate. Conversely, the rate is decreased upon two consecutive transmission failures. Experimental results indicate that ARF works well against signal noise due to location (e.g., near a periodic jamming source). However, the performance of ARF suffers severely when multiple nodes are contending for the wireless channel [6]. More importantly, there lacks a statistical basis to justify the fixed-size window of consecutive transmissions relied on by the algorithm. Not only does this prevent the capture of short-term channel dynamics, but recent performance evaluations suggest that this strategy cannot reliably predict transmission outcomes[2].

In order to improve ARF's short-term and long-term adaptation capabilities, the Adaptive ARF (AARF) algorithm is proposed in [7]. It adapts the previously fixed threshold in ARF (i.e., the number of consecutive transmission successes) according to a binary exponential backoff mechanism from up to 50 observations. Although flexibility is increased, the scheme is still fundamentally based on fixed windows, and as noted by Wong *et al.* [2], it lacks the ability to opportunistically capture the short-term dynamics of the channel.

Arguably the best rate control algorithm in the literature is the Robust Rate Adaptation Algorithm (RRAA) [2], which greatly improves throughput over the previously mentioned schemes. Motivated by fallacies in rate control design guidelines (e.g., using consecutive transmission failures to dictate rate decreases), short-term loss ratios are employed to direct rate changes. Thresholds based on expected throughput are initially computed for each rate, and a fixed-size window of observations is made to determine whether the relevant threshold has been exceeded. The rate is decreased or increased accordingly. Although simulation results validate the parameters of their rate adaptation strategy, RRAA still uses a fixed window to gauge the current channel condition and adjust the transmission rate. As illustrated in this paper, a fixed-size window cannot respond to channel dynamics fast enough.

We propose a rate control scheme termed SRC that makes a fundamental departure from the intuitions behind previously proposed schemes. SRC employs sequential hypothesis testing to flexibly adjust transmission rates in real-time without relying on fixed-size observation windows. It also includes a self-adaptive feedback mechanism to assist with rate increase decisions in a variety of common scenarios. Extensive simulation results show that SRC significantly increases network throughput and quickly adapts to dynamic channel conditions. SRC does not require any type of specialized hardware or

modification to the existing 802.11 standard, and it is aptly suited for implementation in the drivers of wireless network cards. Before elaborating the design of SRC, we briefly discuss its underlying statistical theory.

III. SEQUENTIAL HYPOTHESIS TESTING

In this section, we briefly introduce sequential hypothesis testing, the statistical theory that underlies SRC. A more in-depth discussion on the subject is available in [8].

Assume that there are two hypotheses H_0 and H_1 , and their corresponding probability distribution functions $P(x|H_0)$ and $P(x|H_1)$, respectively. To decide whether H_0 or H_1 is true, a sequence of observations x_1, x_2, \dots needs to be made. Given x_1, x_2, \dots, x_n , we can compute the ratio

$$\rho(n) = \frac{P[x_1, x_2, \dots, x_n|H_1]}{P[x_1, x_2, \dots, x_n|H_0]}. \quad (1)$$

If any two observations are independent from each other, Eq. (1) can be rewritten as follows

$$\rho(n) = \rho(n-1) \frac{P[x_n|H_1]}{P[x_n|H_0]}, \quad (2)$$

with $\rho(1)$ defined as

$$\rho(1) = \frac{P[x_1|H_1]}{P[x_1|H_0]}. \quad (3)$$

Now, when $\rho(n)$ is larger than a predetermined threshold, it implies that the likelihood that x_1, x_2, \dots, x_n happened under H_1 is larger than that under H_0 . On the other hand, if $\rho(n)$ is smaller than another threshold, it implies that the likelihood that x_1, x_2, \dots, x_n happened under H_0 is larger than that under H_1 . As a result, H_0 is accepted to be true instead of H_1 . If $\rho(n)$ is neither large enough nor small enough to make a decision, additional observations must be made, and a new probability ratio is obtained by accumulating the difference of the likelihoods using Eq. (2).

Even though a test accepts a hypothesis as being true, it is possible that a wrong decision is made. There are two possible cases when such errors occur. The first type of error is accepting H_1 when H_0 is actually true, which occurs with probability α . The other type of error occurs when we accept H_0 even though H_1 is true, and this is committed with probability β . (These two types of errors can be regarded as false positive and false negative.) Therefore, in order for a sequential hypothesis test to terminate, there must be enough confidence in its decision. That is, α must be small if H_1 is accepted, and β must be small when H_0 is accepted. The values of α and β are application-specific.

Given the required values of α and β , two thresholds A and B are next computed. These values must be chosen such that the two types of error are guaranteed to be less than α and β , respectively. For example, consider a sample sequence x_1, x_2, \dots, x_n that leads to H_1 being accepted. This implies that both $P[x_1, x_2, \dots, x_n|H_1] \geq (1 - \beta)$ and $P[x_1, x_2, \dots, x_n|H_0] \leq \alpha$ hold, and consequently

$$\rho(n) = \frac{P[x_1, x_2, \dots, x_n|H_1]}{P[x_1, x_2, \dots, x_n|H_0]} \geq \frac{1 - \beta}{\alpha}, \quad (4)$$

which gives an upper bound for A , $A \leq \frac{1-\beta}{\alpha}$. A similar derivation gives $B \geq \frac{\beta}{1-\alpha}$ as a lower bound for B .

As shown by Wald [8], a test with values of A and B corresponding to their upper and lower bounds, respectively, can provide at least the same precision as a test that employs the exact values of A and B . With these thresholds, a sequential hypothesis test can: i) accept H_1 if $\rho(n) \geq A$; ii) accept H_0 if $\rho(n) \leq B$; or iii) conduct additional observations while $B < \rho(n) < A$.

It is worth mentioning the difference between sequential hypothesis testing and the well-established Neyman-Pearson (NP) test. To test between two point hypotheses $H_0 : \theta = \theta_0$ and $H_1 : \theta = \theta_1$, the NP test is most powerful, i.e., with one error probability fixed, the other error probability is minimized, while a test using sequential hypothesis testing only requires the two error probabilities to be bounded. Another difference is that the NP test is in the context of a fixed sampling size, while the number of samples needed for the test with sequential hypothesis testing is variable. Additionally, if restricted to the two error probabilities being bounded, the test in sequential hypothesis testing consumes a smaller sampling size on average than the NP test [8]. Thus, sequential hypothesis testing is more suitable for rate adaptation as the channel condition has to be inferred accurately as well as promptly so as to opportunistically exploit fluctuations in channel quality.

SRC utilizes sequential hypothesis testing to infer the packet loss ratio when a station transmits with a specific data rate. After the two hypotheses H_0 (packet loss ratio is moderate) and H_1 (packet loss ratio is large) are constructed, whether or not a packet transmission is successful is treated as a sample x_i . SRC terminates the test as soon as the probabilities of committing a false positive and a false negative are bounded below the thresholds α and β . With the inferred packet loss ratio, SRC decides to decrease or increase data rate based on whether a different rate can provide greater throughput. With the above background information, we now proceed with a detailed description of the design and motivation of SRC.

IV. SRC: SEQUENTIAL HYPOTHESIS TESTING BASED RATE CONTROL

In this section, we elaborate the design of SRC and discuss its motivation. The presentation of SRC's pseudocode is deferred to the end of this section, after its functionality has been adequately described.

A. Motivation

The design of SRC is motivated by two key observations made during our study of the rate adaptation problem. First, the common technique employed by existing rate control schemes (such as RRAA) of directing rate changes is based on a fixed window of observations. The fixed sampling size makes schemes either too aggressive or too conservative in controlling rate changes. Hence, it is inherently limited in its ability to capture dynamic features of the wireless channel (e.g., mobility induced error).

Secondly, most well-known rate adaptation schemes follow a superficially effective but incorrect guideline: the transmission rate should be increased upon transmission successes. Our

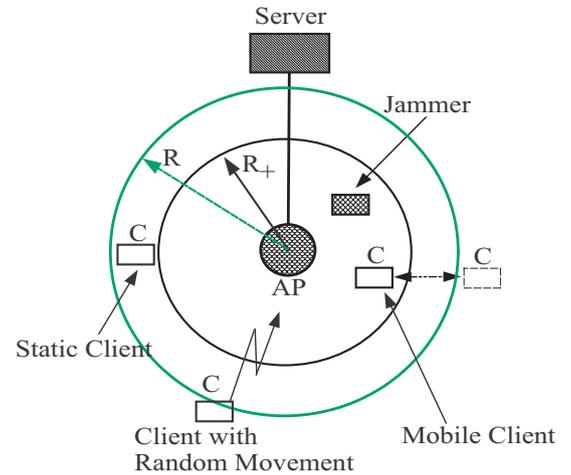


Fig. 1. The network topology.

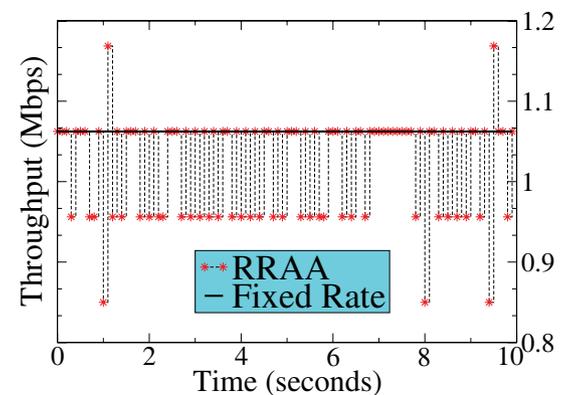


Fig. 2. A motivating scenario of RRAA compared to a fixed rate.

results indicate that this intuition can be misleading in practical scenarios and can incur significant performance penalties. For example, consider the static client shown in Fig. 1.¹ The AP and client communicate through 802.11a that supports data rates 6, 9, 12, 18, 24, 36, and 54 Mbps. The AP sends packets of size 1024 bytes with an inter-arrival time of 0.01 seconds. The client is located within the transmission range of the AP using rate R of 18 Mbps, but outside of the AP's transmission range at the next higher rate R_+ of 24 Mbps. We compare the performance of the client employing a fixed data rate of 18 Mbps to that of the previously discussed RRAA algorithm. Additionally, as discussed in Section V, all simulations are performed using OPNET Modeler 11.5 [5].

As shown in Fig. 2, the client's throughput (in Mbps) fluctuates dramatically when RRAA is selected, while it remains relatively constant and becomes greater with the fixed rate. In particular, when RRAA is applied, the client experiences near perfect channel conditions (i.e., the packet loss ratio is $< 10\%$) with the 18 Mbps rate and subsequently increases its transmission rate to 24 Mbps. However, the client is outside of the AP's transmission range at that rate and receives very few packets from the AP (i.e., the packet loss ratio is $> 85\%$). This situation causes a decrease to the 18 Mbps rate, where the

¹The figure represents multiple scenarios. Now we consider only the static client and the AP. The mobile client, the client with random movement and the jammer are employed for the evaluation performed in Section V.

TABLE I
RATE AND TRANSMISSION TIME PAIRS.

Data Rate(Mbps): R	Transmission Time(s): $Tx(R)$
6	0.001936
9	0.001331
12	0.001011
18	0.000707
24	0.000554
36	0.000402
48	0.000326
54	0.000302

channel quality is improved and the higher rate is eventually switched to again. This is the beginning of a costly oscillation between the two transmission rates. Indeed, about one third of the overall transmission attempts result in failure due to this oscillation. We also note that the results of this simple experiment corroborate the oscillatory behavior of SampleRate [9], whose rate increase guideline follows a similar intuition, as reported in the MadWifi Forum [10].

B. Design

Based on the above observations, we now elucidate the design of SRC. In SRC, the current rate is increased or decreased by one level at a time in a stepwise manner. The decision to increase or decrease the rate is based on whether or not the higher rate or the lower rate, respectively, can provide greater throughput. The throughput is determined using the sending rate and the loss ratio at that rate. However, this computation is not as straightforward as simply applying the nominal rates specified by the 802.11 standard because different portions of the MAC frame are sent at different rates (i.e., the header is always sent at the lowest rate, while the payload is sent at the current nominal rate). Therefore, we convert the nominal sending rate to the actual time required to transmit a frame. As an example, the transmission time $Tx(R)$ of each rate in 802.11a for sending a packet with a payload of 1300 bytes is shown in Table I. Note that the time spent for the backoff procedure is not included in the example, and RTS/CTS control frames are disabled. The throughput hence becomes a function of the transmission time and the loss ratio. More formally, given a transmission rate R with transmission time $Tx(R)$ and loss ratio $p(R)$, the throughput at R is proportional to

$$\frac{1 - p(R)}{Tx(R)}. \quad (5)$$

With this measure of throughput, it is possible to perform an in-depth study of the main functionalities of SRC: rate decrease, rate increase, and dynamic λ adjustment.

1) *Rate Decrease*: Given a current transmission rate R and the next lower rate R_- , the criteria for decreasing the rate from R to R_- is

$$\frac{1 - p(R_-)}{Tx(R_-)} > \frac{1 - p(R)}{Tx(R)}. \quad (6)$$

After isolating the loss ratio $p(R)$, we can equivalently conclude the amount of packet loss necessary for the throughput

at R_- to be greater than that at R . This relationship allows us to compute the *critical value* of $p(R)$ for each rate R , denoted as $P_-^*(R)$ in Eq. (7), for switching the rate from R to R_- .

$$P_-^*(R) = 1 - \frac{(1 - p(R_-))Tx(R)}{Tx(R_-)} \quad (7)$$

Observe that knowledge of $p(R_-)$ is necessary for computing $P_-^*(R)$. One strategy to estimate the loss ratio is to send probe packets separately at rate R_- . Example probing techniques can be found in [11] and [12]. However, without a considerable number of probing packets, $p(R_-)$ will not be estimated accurately. To avoid the overhead associated with this strategy, we employ the station's historical knowledge of the loss ratio at R_- .

Using the available values of $p(R_-)$, the corresponding critical values are computed for each rate. SRC then proceeds to test whether or not the current loss ratio $p(R)$ is larger than the critical value $P_-^*(R)$. That is, two hypotheses are tested on an observation-by-observation basis to decide if the current rate should be maintained or if the next lower rate R_- should be switched to. This decision should be made as soon as possible, while the accepted hypothesis is true with high probability (e.g., larger than 95%).²

The aforementioned hypotheses are tested by two sequential hypothesis tests, denoted as T_{basic} and T_{fast} , that run in parallel with each other. Although each of the tests are trying to decide whether to stay at the current rate (denoted as hypothesis H_0) or switch to the next lower rate (denoted as hypothesis H_1), they are defined differently depending on the particular test and its objectives.

In T_{basic} , the hypotheses are defined as $H_0 : p(R) < P_-^*(R)$ and $H_1 : p(R) \geq \lambda P_-^*(R)$, and the test terminates whenever a decision is made to accept H_0 or H_1 . Note that λ is a design parameter that can be dynamically adjusted in order to compensate the inaccuracy or obsolescence of P_-^* with the varying channel conditions. (How to choose the value of λ will be discussed in Section IV-B3.) If H_0 is accepted, the throughput at the lower rate has been weighed to be less than that at the current rate. Therefore, no decrease in rate is necessary, and the station updates the loss ratio for rate R and restarts T_{basic} with the next transmission attempt. On the other hand, if H_1 is accepted, the loss ratio at the current rate has been concluded to be high enough that greater throughput can be achieved at the next lower rate. In this case, the station decreases its rate to the R_- and restarts both T_{basic} and T_{fast} on the next transmission attempt.

Complementing T_{basic} , the test T_{fast} targets situations where the condition of the channel degrades dramatically in a short period of time (e.g., environments with high mobility). Under these circumstances, the loss ratio becomes large very quickly, and a station is therefore interested in decreasing its rate as soon as possible in order to reduce packet loss. Accordingly, T_{fast} weighs the likelihood of hypothesis $H_0 : p(R) < P_-^*(R)$ versus hypothesis $H_1 : p(R) \geq P_\Delta$. Here, P_Δ is selected based on the maximum tolerable packet loss percentage for a station before it decreases its rate. In the

²The level of confidence in the decision is dependent on the values of α and β that are selected.

case of H_0 being accepted, the station remains at the current rate, updates its value of $p(R)$, and begins a new T_{fast} test. Alternatively, if H_1 is decided to be true, there is considerable packet loss, and the station computes $p(R)$ and switches to the next lower rate R_- , hence requiring the ongoing T_{basic} test to restart at the new rate.

2) *Rate Increase*: Based on updated loss ratios, when a subsequent test accepts H_0 , i.e., it does not decide to reduce the rate, the station may choose to increase its transmission rate because relatively good throughput and low packet loss have been inferred. However, as illustrated in Fig. 2, successful transmissions and a corresponding low loss ratio at the current rate are not necessarily a good indicator of channel quality at a higher rate. Indeed, costly rate oscillations and throughput degradation can occur when such guidelines are used. Although probe packets may be sent at the higher rate, there still lacks a sound intuition for the number of probes required to accurately estimate the current packet loss ratio.

To address the above issues, SRC employs a self-adapting feedback mechanism for rate increase decisions. If the loss ratio, as updated after the acceptance of H_0 by either T_{basic} or T_{fast} , is less than an opportunistic threshold $P_\delta(R)$ (initially set to a maximum value of $P_{max}(R)$), the current rate R is increased one level to R_+ . Now if either of the tests terminates with H_1 being accepted while at R_+ , i.e., there is too large of a loss ratio at the higher rate, the station decreases its rate to R , and halves the value of $P_\delta(R)$. This reduction will occur whenever the station switches from R_+ back to R until it reaches a minimum value of $P_{min}(R) = \frac{P_{max}(R)}{2^m}$, for some integer m , at which point a comparison between the loss ratio $p(R)$ and the minimum value of $P_\delta(R)$ is made to decide on increasing the rate. When at R_+ , if H_0 is accepted by either test, the value of the opportunistic threshold for R , $P_\delta(R)$, is reset to $P_{max}(R)$.

The rationale behind decreasing $P_\delta(R)$ when the station reduces its rate from R_+ to R is that there is an implied bad channel quality at the higher rate and it should be more difficult for a station to immediately switch back to R_+ . If the station continues to oscillate between R and R_+ , the value of $P_\delta(R)$ is decreased exponentially in order to further prevent the station from quickly attempting R_+ again. Once the station stabilizes at R_+ , the value of $P_\delta(R)$ is reset so that less inhibited opportunistic increases can be made in the future.

3) *Dynamic λ Adjustment*: Recall that the historical value of $p(R_-)$ may not reflect the actual loss ratio of rate R_- due to factors such as obsolescence and dramatic variations in channel condition. In order to compensate for such inaccuracies without introducing overhead from large quantities of probe packets, SRC adopts a design parameter λ that is dynamically adjusted according to a simple yet effective heuristic. If T_{fast} determines that the rate should be decreased, λ is set to a value denoted λ_{max} . On the other hand, if either test determines that the current rate should be maintained, the value of λ is reset to λ_{min} . It is also set to λ_{min} after a rate increase. Although the only theoretical constraints on these values are $\lambda > 1$ (to have an effect) and $\lambda P_-^*(R) < 1$ (to make rate decrease possible), in practice, the value of λ should not make T_{basic} obviate T_{fast} .

The intuition behind the real-time adjustments to λ can be

understood as follows. If the channel condition is relatively stable or the station has just increased to a higher rate (indicating up-to-date historical knowledge about the loss ratio), little compensation is required and a relatively small value of λ should be adopted. Conversely, if a rate decrease has just been decided by T_{fast} , the inferred loss ratio at the current rate is quite large (i.e., $p(R) \geq P_\Delta$) and the channel has likely deteriorated significantly within a short period of time. Thus, the historical knowledge of the loss ratio significantly underestimates the current channel condition at the lower rate, and a larger value of λ should be selected.

Given the above functionalities, we now present the pseudocode for SRC. The hypothesis tests T_{basic} and T_{fast} are represented as separate functions from SRC for the sake of clarity.

Algorithm SRC

```

1:  $i = 1, \rho_b = 1, \rho_f = 1; R =$  the current rate;
2: loop
3:   Make the  $i$ th observation,  $x_i$ ;
4:   if  $T_{basic}(x_i)$  or  $T_{fast}(x_i)$  accept  $H_1$  then
5:      $p(R)$  is updated;
6:      $R = R_-$ ;
7:      $\rho_b = 1, \rho_f = 1$ ;
8:     If  $R_\delta(R) > R_{min}(R)$ , then  $R_\delta(R) = R_\delta(R)/2$ ;
9:     If it is  $T_{fast}(x_i)$  that terminates,  $\lambda = \lambda_{max}$ ;
10:  else
11:    if  $T_{basic}(x_i)$  or  $T_{fast}(x_i)$  accept  $H_0$  then
12:       $p(R)$  is updated;
13:      If  $T_{basic}(x_r)$  accepts  $H_0$ ,  $\rho_b = 1$ ;
14:      If  $T_{fast}(x_i)$  accepts  $H_0$ ,  $\rho_f = 1$ ;
15:      For  $R_-$ ,  $P_\delta(R_-) = P_{max}(R_-)$ ;
16:       $\lambda = \lambda_{min}$ ;
17:      if  $p(R) < P_\delta(R)$  then
18:         $R = R_+$ ;
19:         $\rho_b = 1, \rho_f = 1$ ;
20:      end if
21:    end if
22:  end if
23:   $i++$ ;
24: end loop

```

Algorithm T_{basic}

```

1: if  $x_i$  is a successful transmission then
2:    $\rho_b = \rho_b(1 - \lambda P_-^*(R))/(1 - P_-^*(R))$ ;
3: else
4:    $\rho_b = \rho_b(\lambda P_-^*(R))/P_-^*(R) = \rho_b \lambda$ ;
5: end if
6: If  $\rho_b \geq A$ , return  $H_1$  and terminate  $T_{basic}$ ;
7: If  $\rho_b \leq B$ , return  $H_0$  and terminate  $T_{basic}$ ;

```

Algorithm T_{fast}

```

1: if  $x_i$  is a successful transmission then
2:    $\rho_f = \rho_f(1 - P_\Delta)/(1 - P_-^*(R))$ ;
3: else
4:    $\rho_f = \rho_f P_\Delta/P_-^*(R)$ ;
5: end if

```

- 6: If $\rho_f \geq A$, return H_1 and terminate T_{basic} ;
- 7: If $\rho_f \leq B$, return H_0 and terminate T_{basic} ;

Note that the historical loss ratio $p(R_-)$ of the lower rate R_- is required each time $P_-^*(R)$ is calculated using Eq. (7). If the lower rate R_- has not been previously selected, the initial value of $p(R_-)$ is set to 0. The loss ratio $p(R)$ of the current rate R is updated each time T_{basic} or T_{fast} terminates by dividing the number of observed lost frames during the just-terminated test by the total number of transmission attempts that were made during the test. Therefore, when a station switches to the higher rate R_+ , the updated loss ratio $p(R)$ becomes the historical loss ratio used to calculate $P_-^*(R_+)$ when deciding whether to switch back to R .

In the above SRC pseudocode, we implicitly assume that the packet transmissions observed at the MAC layer are independent. Modeling transmission dependencies accurately is very difficult given the time-varying and location-dependent environmental noise, variable effects of multipath fading due to stationary and mobile obstructions, and almost unpredictable changes in the location of a wireless station. It has also been pointed out that the guidelines that predict the next transmission based on past transmissions are misleading [2]. Since SRC aims at exploiting the short-term opportunistic transmission gains, each packet transmission is regarded as Bernoulli process, with the transmission loss ratio varying over time. Such frame error models that assume independent transmission losses are common in the literature [13], [14]. In our future work, we will investigate the impact of the correlation among transmissions on our rate adaptation algorithm.

V. EVALUATION

Mainstream rate adaptation solutions [2], [3], [7], [9] all employ some incarnation of the properties investigated in our study (e.g., fixed-windows and specious rate increase guidelines). Therefore, we evaluate SRC with respect to RRAA since it has been shown to outperform the other schemes in practical settings consistent with our simulation environment.

Our evaluation is conducted using OPNET Modeler 11.5 [5]. Specifically, the MAC layer source code is augmented so that either RRAA or SRC can be used by the wireless stations for 802.11 standard-compliant rate adaptation. Non-standard compliant schemes such as RBAR [15] and OAR [16] are not studied. We compare the performance of the algorithms in the context of the network topology shown in Fig. 1. This topology represents simple yet comprehensive scenarios. The specific scenarios under study are as follows: i) a static client in an environment that is conducive to rate oscillation; ii) a mobile client that moves away from and toward the AP; iii) a client that experiences noisy channel conditions; and iv) a client with random movement. The simulation time for each scenario is 200 seconds.

In all of the aforementioned scenarios, we measure performance according to the ratio of bits received by SRC to those of RRAA, denoted as “relative throughput.” Since relative throughput is measured, the curve representing RRAA corresponds to the x -axis and is hence omitted from the figures below. Therefore, a curve above the x -axis represents a throughput improvement for SRC, while a curve below the x -axis depicts a performance decrease.

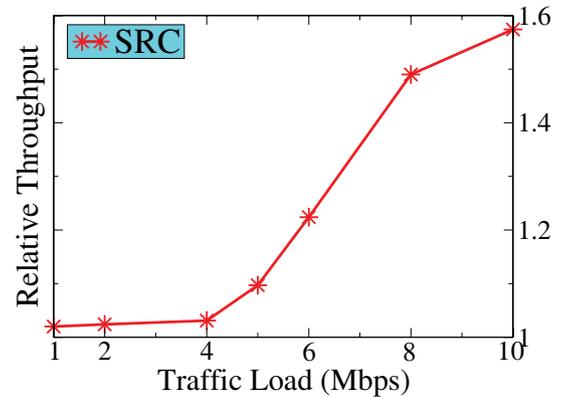


Fig. 3. Performance comparison with a static client in a setting conducive to rate oscillation under TCP traffic loads.

Additionally, the parameters α and β that are used by T_{basic} and T_{fast} are set to a common statistical analysis error value of 5%. The value of P_Δ is used by T_{fast} to decide if the channel deteriorates more than a tolerable level. Here P_Δ is set to 0.4 since losing 40% of transmission should represent a significantly deteriorated channel. However, in general, P_Δ is a tunable system parameter that can be adjusted according to the aggressiveness of a specific wireless policy. The system parameter λ used in T_{basic} is dynamically adjusted between λ_{min} and λ_{max} . Note that to make λ effective, the value of λ should be at least 1. The minimum value of λ , λ_{min} , is set to 1.1. In order to make T_{basic} not obviate T_{fast} , the maximum value of λ , λ_{max} , should not be too large. In our simulations, λ_{max} is set to 1.5. It is very challenging to derive the opportunistic threshold $P_\delta(R)$. Moreover, as indicated in Section IV-A, a small loss ratio at the current rate doesn't necessarily imply good channel condition at the higher rate. The effectiveness of rate increase relies more on the feedback mechanism, i.e., a failed rate increase will halve the value of $P_\delta(R)$. Here, we adopt the values of Opportunistic Rate Increase threshold (ORI) listed in [2] as the initial values of $P_\delta(R)$, i.e., $P_{max}(R)$. The minimum value of $P_\delta(R)$, $P_{min}(R)$, is set to $\frac{P_{max}(R)}{2^4}$ (with $m = 4$). We executed simulations with different m values of 3, 4, and 5, but there was no difference discernable. We focus our evaluations on WLAN networks, i.e., each station communicates through the AP.

A. Static Client

In this scenario, we examine SRC's ability to make stable range changes (i.e., avoid oscillation) and improve throughput in stationary settings. As depicted in Fig. 1, a static client is placed within range of the AP using the 18 Mbps rate, but out of range with the 24 Mbps rate. This enables the channel quality to be near perfect at 18 Mbps and very poor at 24 Mbps. SRC's performance compared to RRAA is shown for both TCP and UDP traffic in Fig. 3 and Fig. 4, respectively.

We see that SRC provides increasingly better throughput as the traffic load increases, reaching about 57% improvement at the 10 Mbps traffic load. This trend is primarily due to SRC's avoidance of (increasingly) costly rate oscillations between

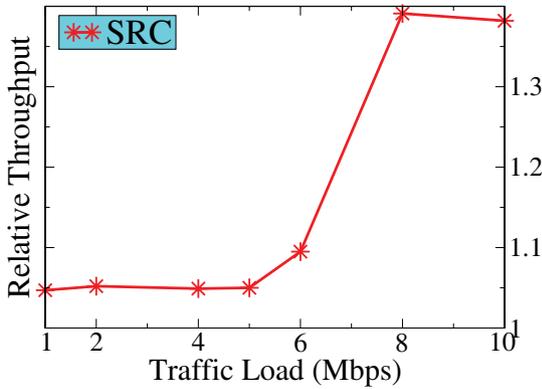


Fig. 4. Performance comparison with a static client in a setting conducive to rate oscillation under UDP traffic loads.

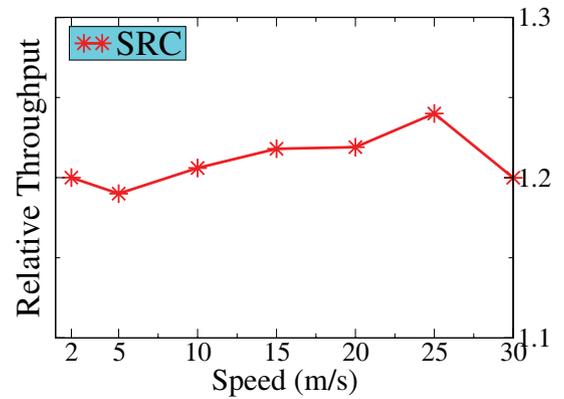


Fig. 6. Performance comparison with a mobile client that is moving away from the AP at different velocities under UDP traffic loads.

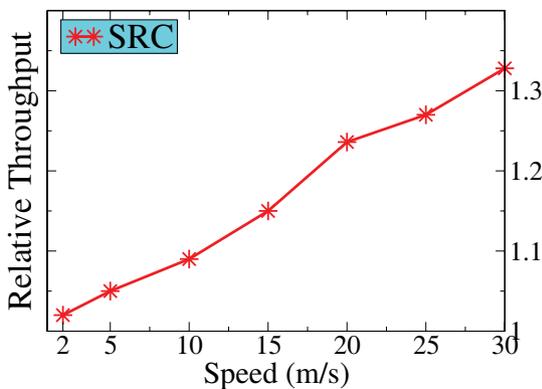


Fig. 5. Performance comparison with a mobile client that is moving away from the AP at different velocities under TCP traffic loads.

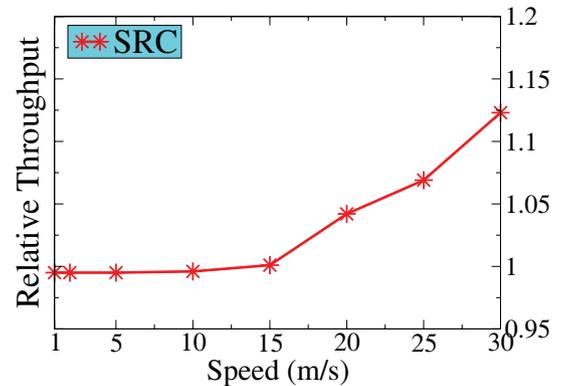


Fig. 7. Performance comparison with a mobile client that is moving toward the AP at different velocities under TCP traffic loads.

the two rates. In particular, the self-adjusting threshold $P_\delta(R)$ enables stable rate changes where oscillation becomes exponentially more difficult. On the other hand, schemes such as RRAA that follow the plausible intuition that rates should be increased based on transmission success ratios are penalized for the frequent (unnecessary) rate switches that are made.

It is necessary to point out that with TCP, the increase of the throughput improvement is more smooth than that with UDP (there is a sudden jump as the traffic load reaches 8 Mbps). The built-in congestion control mechanism of TCP tunes the traffic load according to the transmissions of the MAC layer. With UDP, the MAC layer is fed with traffic without respect to whether the packets are delivered. If the traffic load is low, the packet loss is not critical because the MAC layer will automatically retransmit the failed packet (up to a certain number of times, e.g., 7). It is likely that the packet can be delivered eventually before the next packet arrives at the MAC layer. When the traffic load becomes high, the queue of the MAC layer always has packets. The time wasted on retransmissions can instead be spent on delivering next packet if a more efficient rate adaptation algorithm is applied. The turning point in Fig. 4, i.e., 8 Mbps, implies that the average packet inter-arrival time to the MAC layer meets the average time spent on retransmissions.

B. Mobile Client

Here we evaluate the ability of SRC to handle channel dynamics such as mobility induced channel degradation and mobility induced channel improvement. A mobile client initially positioned within close proximity of the AP moves to the edge of the AP's transmission range at constant speeds of 2, 5, 10, 15, 20, 25 and 30 m/s, and a data rate of 4 Mbps. The simulation topology is illustrated in Fig. 1, and the performance comparison under TCP and UDP traffic is shown in Fig. 5 and 6, respectively. Additionally, after moving away from the AP, the client moves back toward the AP with constant speeds. The performance comparison result is demonstrated in Fig. 7 and 8 for TCP and UDP traffic, respectively.

When the channel condition deteriorates (the client moving away from the AP), as the client velocity increases, the throughput improvement obtained from SRC becomes greater. This reflects SRC's rate adjustment agility, and its ability to dynamically adjust rates without relying on a fixed-size observation window. Indeed, a greater velocity degrades the channel faster, and penalizes schemes that cannot react fast enough such as RRAA because of its fixed decision threshold. We note that the throughput improvement with TCP is greater than that with UDP because TCP's built-in congestion control mechanism throttles the traffic that is sent with RRAA after it fails to respond to the rapid channel degradation.

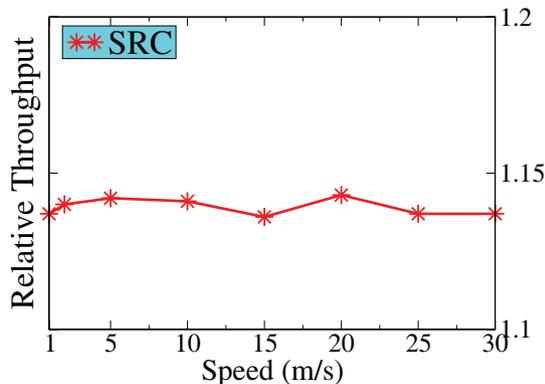


Fig. 8. Performance comparison with a mobile client that is moving toward the AP at different velocities under UDP traffic loads.

When the channel condition keeps improving (the client moving toward the AP), noticeable throughput improvement obtained by SRC can be observed for both TCP and UDP traffic. It is interesting to see that under TCP traffic loads, the throughput improvement has a steady increase as the velocity gets faster, while it remains approximately the same under UDP traffic loads. This can also be explained by TCP’s built-in traffic control mechanism. Since SRC can react faster than RRAA, the client with SRC arrives at the higher rate earlier so that more traffic is transmitted successfully with a higher rate, which allows TCP to unleash (i.e., throttle less traffic) more traffic.

C. Noisy Channel

We also consider the common real-life situation of a client experiencing a noisy (or jammed) channel. As shown in Fig. 1, a *jammer* is placed between the AP and client to simulate a noisy communication channel. The channel degrades as the periods of inter-jamming silence get smaller. Note that this scenario can also represent a signal under variable fading.

The result under TCP traffic loads is shown in Fig. 9, which indicates that SRC consistently improves throughput over RRAA, especially under poor channel conditions. For example, when the silence width is near 0.2 seconds, SRC achieves approximately 59% more throughput than RRAA. We also point out that even as the periods of inter-jamming silence get longer (i.e., as the channel quality improves), SRC’s flexible decision making still allows it to outperform RRAA by as much as 24%. These examples illustrate SRC’s ability to adapt to dynamic fluctuations in channel quality and increase network throughput. For the limit of space, the result of UDP is not presented, in which SRC also outperforms RRAA from our observation.

D. Random Movement

In the last scenario, we test the performance of SRC when the movement of the client is random. We use the random waypoint as the mobility model for the client. The AP is placed in the center of a predefined region, a 500 meters by 500 meters square. Inside the region, the client picks a destination randomly and moves towards it at a speed chosen

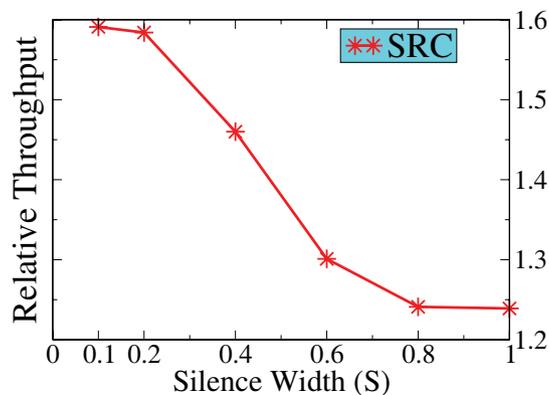


Fig. 9. Performance comparison with a client utilizing a channel exposed to different levels of noise under TCP traffic loads.

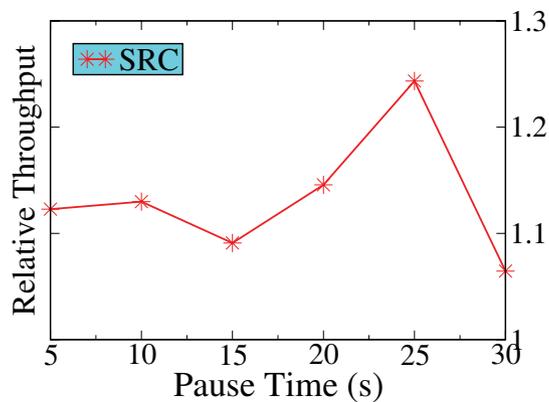


Fig. 10. Performance comparison with a client with random movement under UDP traffic loads

uniformly at random from the range of (0, 10) m/s. After it arrives at the destination, it pauses for a constant period of time. And then it picks a second random destination and moves towards it before pausing for another constant period of time. This process is repeated throughout the simulation. The UDP traffic load is used and fixed at 10Mbps. We vary the constant pause time 5 seconds to 30 seconds (increase the pause time by 5 each test run).

With the client under random mobility, the channel condition exposed to the client doesn’t change monotonically, which is more likely to occur in reality. The result is shown in Fig.10. Since, with each constant pause time, the client moves randomly, i.e., towards random destinations and at random speeds, a clear trend of throughput improvement may not be observed from the case of 5 seconds pause time to the case of 30 seconds pause time. Nevertheless, the result clearly demonstrates the better performance of SRC compared with RRAA. The throughput improvement can be as much as 25% and the least throughput improvement is still about 6% .

E. Remarks

SRC’s consistent improvement in throughput is attributed to its response agility that is empowered by sequential hypothesis testing. To give a closer view of the response time, we

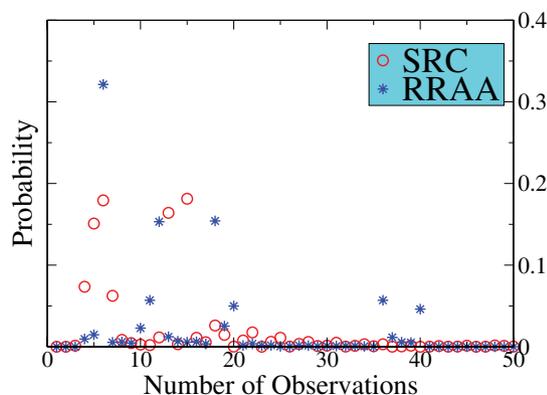


Fig. 11. PMF of number of observations with a mobile client that is moving away from the AP at the speed of 5 m/s.

plot the probability mass functions (PMFs) of the number of observations used by SRC and RRAA to make a decision. An example scenario is shown in Fig. 11, where the client moves away from the AP at the speed of 5 m/s. As it is demonstrated in the figure, the number of observations required by SRC concentrates in small values. On the contrary, most decisions made by RRAA consumes larger number of observations. Note that similar results are also found in other scenarios. Thus, it further proves that SRC responds to channel condition degradation (through the packet loss ratio) earlier than RRAA, which results in a higher throughput.

Another important observation is that SRC better supports applications that are sensitive to packet loss. For example, TCP based applications will suffer from data sending rate reduction when there is packet loss. As it is shown in all of the TCP scenarios, the throughput improvement of SRC over RRAA grows with the degradation in the channel quality. Specifically, as the client velocity increases or jammer silence width decreases, SRC allows TCP based applications to unleash more traffic than that of RRAA, which in turn helps SRC achieve a higher throughput.

VI. CONCLUSION

In this paper, we developed a sequential hypothesis testing based rate control algorithm for 802.11 networks termed SRC that challenges the well-entrenched intuitions of previous rate adaptation studies. The underlying design of SRC avoids both fixed-size transmission windows and deceptively attractively rate increase guidelines, while robustly increasing network throughput over other mainstream rate adaptation solutions by as much as 59%. SRC does not require any type of specialized hardware or modification to the 802.11 standard, and it is well-suited for implementation in WLAN device drivers.

As part of our future work, we plan to explore the tradeoff between decision flexibility and throughput as it relates to the test parameters α and β , and modify the design of SRC accordingly so that the throughput can be further improved. Additionally, we anticipate the evaluation of our design in the context of wireless mesh networks.

REFERENCES

- [1] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris, "Link-level measurements from an 802.11b mesh network," in *SIGCOMM '04: Proceedings of the 2004 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, 2004, pp. 121–132.
- [2] S. H. Y. Wong, S. Lu, H. Yang, and V. Bharghavan, "Robust rate adaptation for 802.11 wireless networks," in *MobiCom '06: Proceedings of the 12th Annual International Conference on Mobile Computing and Networking*, New York, 2006, pp. 146–157.
- [3] A. Kamerman and L. Monteban, "Wavelan(c)-ii: a high-performance wireless lan for the unlicensed band," *Bell Labs Techn. J.*, vol. 2, no. 3, pp. 118–133, 1997. [Online]. Available: <http://dx.doi.org/10.1002/bltj.2069>
- [4] F. Y. Li, A. Kristensen, and P. Engelstad, "Hidden terminal detection in 802.11-based wireless ad hoc networks," in *IST Summit '06: Proceedings of the 15th IST Mobile & Wireless Communication Summit*, 2006.
- [5] "Opnet network modeler and simulator." [Online]. Available: <http://www.opnet.com/>
- [6] N. Kim, "IEEE 802.11 mac performance with variable transmission rates," *IEICE Trans. Commun.*, special edition on Advances in Ad Hoc Mobile Communications and Networking, vol. 88, no. 9, pp. 3524–3531, 2005.
- [7] M. Lacage, M. H. Manshaei, and T. Turletti, "IEEE 802.11 rate adaptation: a practical approach," in *MSWiM '04: Proceedings of the 7th ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, New York, 2004, pp. 126–134.
- [8] A. Wald, *Sequential Analysis*. J. Wiley & Sons, 1947.
- [9] J. Bicket, "Bit-rate selection in wireless networks," Master's thesis, Massachusetts Institute of Technology, Feb. 2005.
- [10] "Madwifi: a linux kernel device driver for wireless lan chipsets." [Online]. Available: <http://madwifi.org/>
- [11] D. S. J. D. Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," in *MobiCom '03: Proceedings of the 9th Annual International Conference on Mobile Computing and Networking*, New York, 2003, pp. 134–146.
- [12] R. Draves, J. Padhye, and B. Zill, "Routing in multi-radio, multi-hop wireless mesh networks," in *MobiCom '04: Proceedings of the 10th Annual International Conference on Mobile Computing and Networking*, New York, 2004, pp. 114–128.
- [13] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE J. Select. Areas Commun.*, vol. 18, no. 3, pp. 535–547, 2000.
- [14] J. Jun, P. Peddabachagari, and M. Sichitiu, "Theoretical maximum throughput of IEEE 802.11 and its application," in *NCA'03: Second IEEE International Symposium on Network Computing and Application*, Cambridge, USA, 2003.
- [15] G. Holland, N. Vaidya, and P. Bahl, "A rate-adaptive MAC protocol for multi-hop wireless networks," in *MobiCom '01: Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, New York, 2001, pp. 236–251.
- [16] B. Sadeghi, V. Kanodia, A. Sabharwal, and E. Knightly, "Opportunistic media access for multirate ad hoc networks," in *MobiCom '02: Proceedings of the 8th Annual International Conference on Mobile Computing and Networking*, New York, 2002, pp. 24–35.