# Rethinking Thresholds-based Rate Adaptation Algorithms: A Reverse Engineering Perspective

Yang Song[§], Xiaoyan Zhu[†], Yuguang Fang[§†] and Hailin Zhang[†]

[§] Department of Electrical and Computer Engineering
University of Florida
Gainesville, Florida 32611
Email: yangsong@ufl.edu; fang@ece.ufl.edu
[†]National Key Laboratory of Integrated Services Networks
Xidian University, Xi'an, China
Email: xyzhu@mail.xidian.edu.cn; hlzhang@xidian.edu.cn

*Abstract*— **Rate adaptation algorithms play a crucial role in IEEE 802.11 WLANs. While the network performance depends greatly on the rate adaptation algorithms, the detailed implementation is left to vendors. Due to its simplicity and practicality, the generic rate adaptation algorithm based on up/down thresholds is widely adopted in commercial IEEE 802.11 devices. Taking the popular ARF algorithm for example, the data rate is increased when ten consecutive transmissions are successful and a date rate downshift is triggered by two consecutive failed transmissions. Although widely deployed, disclosing the implicit objective function that the rate adaptation algorithm is dynamically maximizing, remains as an open problem in the literature. In this paper, we investigate the thresholds-based rate adaptation algorithm via a reverse engineering perspective where the implicit objective function is revealed. We consider this reverse engineering study of the thresholds-based rate adaptation algorithm as an important first step towards a comprehensive understanding on the rate adaptation mechanism designs and the complex interactions among multiple IEEE 802.11 stations.**

## I. INTRODUCTION

IEEE 802.11 WLAN technique has become the dominating technology for indoor wireless Internet access. While the original IEEE 802.11 DSSS only provides two physical data rates (1 Mbps and 2 Mbps), the current IEEE standard provides several available data rates based on different modulation and coding schemes. For example, IEEE 802.11b supports 1 Mbps, 2 Mbps, 5.5 Mbps and 11 Mbps and IEEE 802.11g provides 12 physical data rates up to 54 Mbps. In order to maximize the network throughput, IEEE 802.11 devices, i.e., stations, need to adaptively change the data rate to combat with the time-varying channel environments. For instance, when the channel is good, a high data rate which usually requires higher SNR can be utilized. On the contrary, a low data rate which is error-resilient might be favorable for a bad channel. The operation of dynamically selecting data rates in IEEE 802.11 WLANs is called *rate adaptation* in general.

The implementation of rate adaptation algorithms is not specified by the IEEE 802.11 standard. This intentional omission flourishes the studies on this active area where a variety of rate adaptation algorithms have been proposed [1]–[17]. The common challenge of rate adaptation algorithms is how to match the unknown channel condition optimally such that the network throughput is maximized. According to the methods of estimating channel conditions, rate adaptation algorithms can be divided into two major categories. The first one is called *closed-loop* rate adaptation. Most schemes in this approach enable the receiver to measure the channel quality and sends back this information explicitly to the transmitter where the rate adaptation decision is made. For example, the receiver records the SNR or RSSI value of the received packet and sends this information back to the transmitter via CTS or ACK packet. Consequently, the transmitter estimates the channel condition based on the feedback signal and adjusts the data rate accordingly. By utilizing additional feedback mechanisms, the close-loop rate adaptation algorithms can achieve a better performance than the open-loop counterparts. However, in practice, the close-loop rate adaptation algorithms are rarely used in commercial IEEE 802.11 devices. This is because that the extra feedback information needs to be conveyed reliably and hence an inevitable modification on the current IEEE 802.11 standard is needed. This non-compatibility hinders the close-loop rate adaptation algorithms from practical implementations in current off-the-shelf IEEE 802.11 products.

The second category of rate adaptation algorithms, which is predominantly adopted by the vendors, is labeled as *open-loop* algorithms. The widely utilized auto rate fallback algorithm, a.k.a., ARF, falls into this category. As many other open-loop rate adaptation algorithms, ARF adjusts the date rate solely based on the IEEE 802.11 ACK packets. For example, in Enterasys RoamAbout IEEE 802.11 card [18], two consecutive frame transmission failures, indicated by not receiving ACKs promptly, induces a rate downshift, while ten consecutive successful frame transmissions triggers a rate upshift [19]. Most commercialized IEEE products follow this up/down

scheme [4], [5]. In this paper, we focus on the open-loop rate adaptation algorithms due to the practical merits. More specifically, we consider a generic *thresholds-based rate adaptation algorithm* which works as follows. If there are $\theta_u$ consecutive successful transmissions, the data rate is upgraded to the next level. On the other hand, if $\theta_d$ consecutive transmissions failed, a rate downshift is triggered. Since ARF is merely a special case of this generic rate adaptation algorithm, our analysis can be applied to ARF and its variants straightforwardly.

As a tradeoff with simplicity, there are several challenges existing for the thresholds-based rate adaptation algorithm. First, due to the trial-and-error based up/down mechanism, it inherently lacks the capability of capturing short dynamics of the channel variations [5]. To tackle this issue, Qiao *et.al.* propose a fast responsive rate adaptation solution in [1]. By introducing a measure of "delay factor", the responsiveness of the thresholds-based rate adaptation algorithm can be guaranteed. The second challenge of the thresholds-based rate adaptation algorithm is the indifference to collision-induced failures and noise-induced failures. It is worth noting that in a multiuser WLAN, an ACK timeout can be ascribed to either an MAC layer collision, or an erroneous channel. However, the thresholds-based rate adaptation algorithm is unable to distinguish them effectively. Therefore, excessive collisions may introduce unnecessary rate degradations which significantly deteriorate the system performance. Attributing the actual reason of a transmission failure, or a packet loss, is named *loss diagnosis* [20] and has attracted tremendous attention from the community. Most of the solutions rely on the RTS/CTS signalling approach to differentiate the two causes of packet losses. However, the RTS/CTS mechanism is rarely turned on in practical IEEE 802.11 WLANs due to its costly overhead[1] [5]. Recently, Choi *et.al.* [22] propose an algorithmic solution, which is specific to the thresholds-based rate adaptation algorithm, to mitigate the collision effect in multiuser IEEE 802.11 WLANs. Therefore, the performance deterioration by the "indifference of collision" can be compensated effectively.

In this paper, we analytically investigate the behavior of the thresholds-based rate adaptation algorithm from a reverse engineering perspective. In other words, we answer the essential yet unresolved question, i.e., *"What is the thresholds-based rate adaptation algorithm actually optimizing?"*, by unveiling the implicit objective function. As a result, several intuitive observations of the thresholds-based rate adaptation algorithm can be explained straightforwardly by inspecting this objective function. Therefore, our reverse engineering model provides an alternative means to understand the thresholds-based rate adaptation algorithm. Our work is a complement to the recent trend of reverse engineering studies on existing heuristics-based networking protocols, such as TCP (transport layer) [23]–[25], BGP (network layer) [26] and random access MAC protocol (data link layer) [27]. Our results explicitly show that

the values of $\theta_u$ and $\theta_d$ play an important role in the objective function and thus the network performance hinges largely on the selection of the up/down thresholds.

The rest of this paper is organized as follows. The reverse engineering model of the thresholds-based rate adaptation algorithm is presented in Section II. More discussions on the disclosed implicit objective function are provided in Section III and Section IV concludes this paper.

## II. Reverse Engineering for the Thresholds-based Rate Adaptation Algorithm

We consider a station in a multi-rate IEEE 802.11 WLAN. There are $N$ stations in the WLAN where each station, say $i$, has a transmission probability of $p_i$. Note that the equivalence of the probability model and the IEEE 802.11 binary exponential backoff CSMA/CA model has been extensively studied in [28] and [27]. Throughout this paper, we assume that the transmission probability of each station is fixed. The interaction of the data rate and the transmission probability is left as future research. For notation brevity, we assume that the stations have a same transmission probability of $p$. We emphasize that this assumption induces no loss of generality.

In this work, we focus on the widely deployed open-loop thresholds-based rate adaptation algorithm. Not surprisingly, the speed of channel variations has a great impact on the performance of the rate adaptation algorithm. In addition, while the original intention of the thresholds-based rate adaptation algorithm is to maximize the average throughput, a natural question arises that whether this is *indeed* the case. In this section, to better understand the impact of $\theta_u$ and $\theta_d$ on the performance of the thresholds-based rate adaptation algorithm, we investigate the thresholds-based rate adaptation algorithm from a reverse engineering standpoint. We assume that the RTS/CTS signals are turned off due to the practical concerns. In a time slot, say $t$, we denote the channel state[2] as $s(t)$ and denote the successful transmission probability, given the current transmission rate $r(t)$ and the channel condition $s(t)$, as

$$P_S(s(t), r(t)) = p(1-p)^{N-1}(1 - e(s(t), r(t))) \quad (1)$$

where $e$ denotes the frame error rate (FER) and is given by

$$e(s(t), r(t)) = 1 - (1 - P_e(s(t), r(t)))^L \quad (2)$$

and $P_e(s(t), r(t))$ is the bit error rate (BER) which is determined by the current data rate, i.e., modulation scheme, and the current channel condition. $L$ is the frame length of the packet. Similarly, we define

$$P_F(s(t), r(t)) = 1 - P_S(s(t), r(t)) \quad (3)$$

as the transmission failure probability at time $t$. It is worth noting that both $P_S$ and $P_F$ are functions of the current data rate $r(t)$ as well as the instantaneous channel condition $s(t)$, which is a random process in nature. Particularly, we assume

---

[1]According to the reports from CAIDA [21], the Internet traffic is dominated by small size packets under 100 bytes. Therefore, the RTS/CTS signalling mechanism becomes burdensome in practice.

[2]Note that the number of feasible channel states can be potentially infinite.

that the thresholds-based rate adaption algorithm will increase the data rate by an amount of $\delta$ if there are $\theta_u$ consecutive successful transmissions and decrease it by $\delta$ if there are $\theta_d$ consecutive failures. Denote $u = \theta_u - 1$ and $d = \theta_d - 1$ for notation succinctness. We define a binary indicator function $\zeta(t)$ where $\zeta(t) = 1$ means that the transmission at time slot $t$ is successful and $\zeta(t) = 0$ otherwise, for this particular station. Mathematically, the updating rule of the thresholds-based rate adaptation algorithm can be written as

$$
\begin{aligned}
r(t+1) &= (r(t) + \delta)\Gamma_{\zeta(t)=1}\Gamma_{\zeta(t-1)=1}\cdots\Gamma_{\zeta(t-u)=1} \\
&+ (r(t) - \delta)\Gamma_{\zeta(t)=0}\Gamma_{\zeta(t-1)=0}\cdots\Gamma_{\zeta(t-d)=0} \\
&+ r(t)\Gamma_{o.w.}
\end{aligned} \tag{4}
$$

where

$$
\Gamma_x = \begin{cases} 1, & \text{if event x is true} \\ 0, & \text{if event x is false.} \end{cases} \tag{5}
$$

For example, $\Gamma_{\zeta(t)=1} = 1$ if the transmission at time $t$ is successful and $\Gamma_{\zeta(t)=1} = 0$ otherwise. The symbol of $o.w.$ denotes the event that neither $\theta_u$ consecutive successful nor $\theta_d$ consecutive failed transmissions happened. For simplicity, we assume that the maximum allowable data rate is sufficiently large and the minimum data rate is zero, i.e., not transmitting. Define

$$
\mathbf{h}(t) = [r(t), r(t-1), \cdots, r(1), e(s(t), r(t)), \cdots, e(s(1), r(1))] \tag{6}
$$

as the *history vector*. In addition, we define

$$
Z(t+1) = \mathbb{E}\{r(t+1)|\mathbf{h}(t)\} \tag{7}
$$

where $\mathbb{E}$ is the expectation operator.

*Condition 1:* (C.1) The channel states between two consecutive successful transmissions or two consecutive failed transmissions are independent random variables.

We emphasize that the restrictive condition (C.1) is not our general assumption. If (C.1) is satisfied, however, the derivation of the reverse engineering analysis can be presented in a more concise form, as will be shown shortly.

First, we obtain

$$
\mathbb{E}\{\Gamma_{\zeta(t)=1}\Gamma_{\zeta(t-1)=1}\cdots\Gamma_{\zeta(t-u)=1}|\mathbf{h}(t)\}
$$
$$
= Pr\{\Gamma_{\zeta(t)=1} = 1, \Gamma_{\zeta(t-1)=1} = 1, \cdots, \Gamma_{\zeta(t-u)=1} = 1|\mathbf{h}(t)\}. \tag{8}
$$

If condition (C.1) is satisfied, (8) can be further decomposed as

$$
\begin{aligned}
&\mathbb{E}\{\Gamma_{\zeta(t)=1}\Gamma_{\zeta(t-1)=1}\cdots\Gamma_{\zeta(t-u)=1}|\mathbf{h}(t)\} \\
&= Pr\{\Gamma_{\zeta(t)=1} = 1|\mathbf{h}(t)\} \times Pr\{\Gamma_{\zeta(t-1)=1} = 1|\mathbf{h}(t)\} \\
&\cdots \times Pr\{\Gamma_{\zeta(t-u)=1} = 1|\mathbf{h}(t)\} \\
&= \prod_{k=0}^{u} P_S(s(t-k), r(t-k)).
\end{aligned} \tag{9}
$$

Similarly, we have

$$
\begin{aligned}
&\mathbb{E}\{\Gamma_{\zeta(t)=0}\Gamma_{\zeta(t-1)=0}\cdots\Gamma_{\zeta(t-d)=0}|\mathbf{h}(t)\} \\
&= Pr\{\Gamma_{\zeta(t)=0} = 1, \Gamma_{\zeta(t-1)=0} = 1, \cdots, \Gamma_{\zeta(t-u)=0} = 1|\mathbf{h}(t)\}.
\end{aligned} \tag{10}
$$

If (C.1) is assumed to be valid, we can obtain

$$
\begin{aligned}
&\mathbb{E}\{\Gamma_{\zeta(t)=0}\Gamma_{\zeta(t-1)=0}\cdots\Gamma_{\zeta(t-d)=0}|\mathbf{h}(t)\} \\
&= \prod_{k=0}^{d}(1 - P_S(s(t-k), r(t-k))).
\end{aligned} \tag{11}
$$

For notation succinctness, we will temporarily assume that (C.1) is satisfied. The condition will be relaxed after the implicit objective function is revealed.

Therefore, we can rewrite (7) as

$$
\begin{aligned}
Z(t+1) &= \mathbb{E}\{r(t+1)|\mathbf{h}(t)\} \\
&= (r(t) + \delta)\prod_{k=0}^{u} P_S(s(t-k), r(t-k)) \\
&+ (r(t) - \delta)\prod_{k=0}^{d}(1 - P_S(s(t-k), r(t-k))) \\
&+ r(t)\left(1 - \prod_{k=0}^{u} P_S(s(t-k), r(t-k)) \right. \\
&\left. - \prod_{k=0}^{d}(1 - P_S(s(t-k), r(t-k)))\right) \\
&= r(t) + \delta\left\{\prod_{k=0}^{u} P_S(s(t-k), r(t-k)) \right. \\
&\left. - \prod_{k=0}^{d}(1 - P_S(s(t-k), r(t-k)))\right\}.
\end{aligned} \tag{12}
$$

Let us revisit (4), which can be rewritten as

$$
\begin{aligned}
r(t+1) &= r(t) + \delta\left[\frac{1}{\delta} \times \left\{ (r(t) + \delta)\Gamma_{\zeta(t)=1}\cdots\Gamma_{\zeta(t-u)=1}\right.\right. \\
&+ (r(t) - \delta)\Gamma_{\zeta(t)=0}\Gamma_{\zeta(t-1)=0}\cdots\Gamma_{\zeta(t-d)=0} \\
&\left.\left.+ r(t)\Gamma_{o.w.} - r(t) \right\}\right] \\
&= r(t) + \delta\xi(t)
\end{aligned} \tag{13}
$$

where

$$
\begin{aligned}
\xi(t) &= \frac{1}{\delta} \times \left\{ (r(t) + \delta)\Gamma_{\zeta(t)=1}\Gamma_{\zeta(t-1)=1}\cdots\Gamma_{\zeta(t-u)=1} \right. \\
&+ (r(t) - \delta)\Gamma_{\zeta(t)=0}\Gamma_{\zeta(t-1)=0}\cdots\Gamma_{\zeta(t-d)=0} \\
&\left.+ r(t)\Gamma_{o.w.} - r(t) \right\}.
\end{aligned} \tag{14}
$$

We observe that

$$
\begin{aligned}
\mathbb{E}\{\xi(t)|\mathbf{h}(t)\} &= \prod_{k=0}^{u} P_S(s(t-k), r(t-k)) \\
&- \prod_{k=0}^{d}(1 - P_S(s(t-k), r(t-k))).
\end{aligned} \tag{15}
$$

Next, we present the reverse engineering theorem for the thresholds-based rate adaptation algorithm.

*Theorem 1:* The thresholds-based rate adaptation algorithm of (4) is a stochastic approximation which solves an implicit objective function $U(t)$, with a constant step size of $\delta$, where $U(t)$ is in the form of

$$
\begin{aligned}
U(t) = \; & r(t)\left\{\prod_{k=0}^{u}P_S(s(t-k),r(t-k))\right. \\
& \left. -\prod_{k=0}^{d}(1-P_S(s(t-k),r(t-k)))\right\} + \mathcal{K} \quad (16)
\end{aligned}
$$

if condition (C.1) is satisfied and $\mathcal{K}$ is an invariant with respect to rate $r(t)$.

*Proof:* Theorem 1 follows directly from the previous analysis. Note that the thresholds-based rate adaptation algorithm can be written as

$$
r(t+1) = r(t) + \delta\xi(t) \quad (17)
$$

where $\xi(t)$ is the stochastic gradient and satisfies

$$
\mathbb{E}\{\xi(t)|\mathbf{h}(t)\} = \frac{\partial U}{\partial r(t)}\Big|_{\mathbf{h}(t)}. \quad (18)
$$

Hence Theorem 1 holds. ∎

## III. DISCUSSIONS

In this section, we provide some discussions on the disclosed utility function in (16).

REMARK 1:

If condition (C.1) does not hold, we can replace

$$
\prod_{k=0}^{u}P_S(s(t-k),r(t-k))
$$

and

$$
\prod_{k=0}^{d}(1-P_S(s(t-k),r(t-k)))
$$

in (16) with

$$
Pr\left\{\Gamma_{\zeta(t)=1}=1,\cdots,\Gamma_{\zeta(t-u)=1}=1|\mathbf{h}(t)\right\} \quad (19)
$$

and

$$
Pr\left\{\Gamma_{\zeta(t)=0}=1,\cdots,\Gamma_{\zeta(t-d)=0}=1|\mathbf{h}(t)\right\} \quad (20)
$$

respectively and the theorem remains valid.

REMARK 2:

It is worth noting that the objective function $U(t)$ is a time-varying function which is determined by the data rate as well as the channel conditions of the past $\tau$ time slots where $\tau = \max(\theta_u,\theta_d)$. Note that the data rate within the last $\tau$ time slots always remains unchanged. However, the channel fluctuations affect the successful transmission probability $P_S$ and thus alter the objective function $U(t)$.

REMARK 3:

The partial derivative of the objective of $U(t)$ given $\mathbf{h}(t)$, i.e.,

$$
\widetilde{\xi(t)} = \prod_{k=0}^{u}P_S(s(t-k),r(t-k)) - \prod_{k=0}^{d}(1-P_S(s(t-k),r(t-k))) \quad (21)
$$

is also time-varying. If the probability that the last $\theta_u$ transmissions are all successful is greater than the probability that the last $\theta_d$ transmissions are all failures, the station tends to increase the data rate and vice versa. The speed of rate increasing or deceasing is determined by the difference of these two probabilities. In other words, the partial derivative in (21) could be either positive or negative which corresponds to a rate upshift or a rate downshift. The absolute value of the instantaneous derivative determines the speed of rate changing.

REMARK 4:

A direct computation of the partial derivative in (21) is challenging, if not impossible, due to the uncertainty induced by the unpredictable stochastic channel. Therefore, the thresholds-based rate adaptation algorithm, described in (4), utilizes an alternative stochastic approximation approach with the unbiased estimation of $\widetilde{\xi(t)}$, i.e., $\xi(t)$ in (14), which significantly reduces the computational complexity since only local information based on ACKs is required. This nature of simplicity and practicability facilitates the popularity of the thresholds-based rate adaptation algorithm and its variants such as ARF.

To achieve a better understanding of (16), let us consider the following extreme cases from a reverse engineering standpoint.

- $(u=0,d=0) \Rightarrow (\theta_u=\theta_d=1)$: In this case, the thresholds-based rate adaptation algorithm increases the data rate if the current transmission is successful and decreases otherwise. From (16), we can observe that this aggressive algorithm merely compares the successful probability with the failure probability of the current time slot and expects that the next time slot will remain the current channel condition.

- $(u=+\infty,d=0) \Rightarrow (\theta_u=+\infty,\theta_d=1)$: From (16), we can see that the derivative is always negative since the first part of (21) is zero. Therefore, the thresholds-based rate adaptation will keep decreasing data rate until the minimum data rate is reached, i.e., zero, which collaborates the intuition.

- $(u=0,d=+\infty) \Rightarrow (\theta_u=1,\theta_d=+\infty)$: In this scenario, the second part of (21) is always zero and thus the algorithm will keep upgrading data rate until the maximum data rate is achieved.

- $(u=+\infty,d=+\infty) \Rightarrow (\theta_u=\infty,\theta_d=+\infty)$: The derivative of (21) is always zero and hence the data rate never changes.

Hence, by inspecting (16) and (21) directly, we provide an alternative means to understand the behavior of the thresholds-based rate adaptation algorithm. In (4), we have assumed a constant stepsize $\delta$ while in current off-the-shelf IEEE 802.11

devices, a discrete set of data rates are provided. However, continuous rate adaptation can be achieved by controlling the transmission power jointly or deploying Adaptive-Coding-and-Modulation (ACM) capable devices. Therefore, our reverse engineering model, while fits in continuous rate scenarios, provides an approximate model for discrete rate adaptation scenarios.

It is immediate to observe from (16) and (21) that the selection of $\theta_u$ and $\theta_d$ has a significant impact on the performance of the rate adaptation algorithm. Ideally, the rate adaptation algorithm attempts to find the data rate which maximizes the expected throughput in the next time slot, i.e.,

$$r' = \mathrm{argmax}_r \, r \times P_S(s(t+1), r). \tag{22}$$

Define

$$V = r \times P_S(s(t+1), r). \tag{23}$$

Therefore, if we can estimate the value of $\frac{\partial V}{\partial r}$ and relate it by

$$\widetilde{\xi(t)} \to \frac{\partial V}{\partial r}, \tag{24}$$

then the thresholds-based rate adaptation algorithm is indeed optimizing the expected throughput via the stochastic approximation approach. Therefore, we believe that the reverse engineering result in this paper provides a first step towards a comprehensive understanding on the optimal-yet-simple rate adaptation algorithm designs. In addition, based on the unveiled implicity objective function of (16), the interactions of rate adaptations among multiple IEEE 802.11 stations can be investigated in a game theoretical framework.

## IV. Conclusions and Future Work

In this paper, we investigate the thresholds-based rate adaptation algorithm which is predominantly utilized in practical IEEE 802.11 WLANs. Although widely deployed, the obscure objective function of this type of rate adaptation algorithms, commonly based on the heuristic up/down mechanism, is less comprehended. In this work, we study the thresholds-based rate adaptation algorithm from a reverse engineering perspective. The implicit objective function, which the rate adaption algorithm is actually maximizing, is unveiled. Our results provide, albeit approximate, an analytical model from which the heuristics-based thresholds-based rate adaptation algorithms, such as ARF, can be better understood.

In this paper, to emphasize on the impact of random channel variations, we restrict ourselves to the scenario where all stations have a fixed and known transmission probability $p$. The interaction of the transmission probabilities and the data rates seems interesting and needs further investigation. Due to the competitive nature of channel access, a stochastic game formulation seems appealing. Moreover, our work assumes a fixed length $L$ for every packet. A natural extension to the joint transmission rate and frame size adaptation scenario is our on-going research following [29] and [30].

## References

[1] D. Qiao and S. Choi, "Fast-responsive link adaptation for ieee 802.11 wlans," *IEEE ICC*, 2003.

[2] P. Chevillat, J. Jelitto, A. N. Batteto, and H. L. Truong, "A dynamic link adaptation algorithm for ieee 802.11a wireless lans," *IEEE ICC*, 2003.

[3] M. Lacage, M. H. Manshaei, and T. Turletti, "Ieee 802.11 rate adaptation: A practical approach," *ACM MSWiM*, 2004.

[4] J. Kim, S. Kim, S. Choi, and D. Qiao, "Cara: Collision-aware rate adaptation for ieee 802.11 wlans," *IEEE Infocom*, 2006.

[5] S. H. Y. Wong, H. Yang, S. Lu, and V. Bharghavan, "Robust rate adaptation for 802.11 wireless networks," *ACM MobiCom*, 2006.

[6] B. Sadeghi, V. Kanodia, A. Sabharwal, and E. Knightly, "Opportunistic media access for multirate ad hoc networks," *ACM MobiCom*, 2002.

[7] S. Pal, S. R. Kundu, K. Basu, and S. K. Das, "Ieee 802.11 rate control algorithms: Experimentation and performance evaluation in infrastructure mode," the University of Texas at Arlington, Tech. Rep., 2006.

[8] Q. Pang, V. C. M. Leung, and S. C. Liew, "A rate adaptation algorithm for ieee 802.11 wlans based on mac-layer loss differentiation," *IEEE BroadNets*, 2005.

[9] H. Jung, K. Cho, Y. Seok, T. Kwon, and Y. Choi, "Rara: Rate adaptation using rate-adaptive acknowledgement for ieee 802.11 wlans," *IEEE CCNC*, 2008.

[10] G. Judd, X. Wang, and P. Steenkiste, "Low-overhead channel-aware rate adaptation," *ACM MobiCom (Extended Abstract)*, 2007.

[11] J. C. Bicket, "Bit-rate selection in wireless networks," Master's thesis, Massachusetts Institute of Technology, 2005.

[12] G. Holland, N. Vaidya, and P. Bahl, "A rate-adaptive mac protocol for multi-hop wireless networks," *ACM MobiCom*, 2001.

[13] K. Ad and M. Leo, "Wavelan(c)-ii: a high-performance wireless lan for the unlicensed band," *Bell Labs Technical Journal*, vol. 2, pp. 118–133, 1997.

[14] I. Haratcherev, K. Langendoen, R. Lagendijk, and H. Sips, "Hybrid rate control for ieee 802.11," *ACM MobiWac*, 2004.

[15] [Online]. Available: http://madwifi.org/browser/madwifi/trunk/ath_rate/onoe

[16] J.-H. Yun and S.-W. Seo, "Novel collision detection scheme and its applications for ieee 802.11 wireless lans," *Computer Communications*, vol. 30, 2007.

[17] S.-C. Wang and A. Helmy, "Beware: Background traffic-aware rate adaptation for ieee 802.11 mac," *IEEE WoWMoM*, 2008.

[18] [Online]. Available: http://secure.enterasys.com/products/wireless/RBTBG-AW

[19] J. Choi, K. Park, and C. kwon Kim, "Cross-layer analysis of rate adaptation, dcf and tcp in multi-rate wlans," *IEEE Infocom*, 2007.

[20] S. Rayanchu, A. Mishra, D. Agrawal, S. Saha, and S. Banerjee, "Diagnosing wireless packet losses in 802.11: Separating collision from weak signal," *IEEE Infocom*, 2008.

[21] [Online]. Available: http://www.caida.org/data/

[22] J. Choi, J. Na, K. Park, and C. kwon Kim, "Adaptive optimization of rate adaptation algorithms in multi-rate wlans," *IEEE ICNP*, 2007.

[23] F. Kelly, A. Maulloo, and D. Tan, "Rate control in communication networks: shadow prices, proportional fairness and stability," *Journal of Operation Research Society*, vol. 49, pp. 237–252, 1998.

[24] S. H. Low and D. E. Lapsley, "Optimization flow control — I: basic algorithm and convergence," *IEEE/ACM Transactions on Networking*, vol. 7, no. 6, pp. 861–874, 1999.

[25] M. Chiang, S. H. Low, A. R. Calderbank, and J. C. Doyle, "Layering as optimization decomposition: A mathematical theory of network architectures," *Proceedings of the IEEE*, vol. 95, pp. 255–312, Mar. 2007.

[26] T. G. Griffin, F. B. Shepherd, and G. Wilfong, "The stable paths problem and interdomain routing," *IEEE/ACM Transactions on Networking*, vol. 10, pp. 232–243, 2002.

[27] J.-W. Lee, A. Tang, J. Huang, M. Chiang, and A. R. Calderbank, "Reverse-engineering mac: A non-cooperative game model," *IEEE Journal on Selected Areas in Communications*, Jun. 2007.

[28] F. Cali, M. Conti, and E. Gregori, "Dynamic tuning of the ieee 802.11 protocol to achieve a theoretical throughput limit," *IEEE/ACM Transactions on Networking*, vol. 8, 2000.

[29] C. cheng Chen, E. Seo, H. Luo, N. H. Vaidya, and X. Wang, "Rate-adaptive framing for interfered wireless networks," *IEEE Infocom*, 2007.

[30] D. Qiao and S. Choi, "Goodput enhancement of ieee 802.11a wireless lan via link adaptation," *IEEE ICC*, 2001.