



Li, L., Fan, Z., & Kaleshi, D. (2012). Using multiple metrics for rate adaptation algorithms in IEEE 802.11 WLANs. In *IEEE Wireless Communications and Networking Conference* (pp. 2807-2812). Institute of Electrical and Electronics Engineers (IEEE).
<https://doi.org/10.1109/WCNC.2012.6214279>

Link to published version (if available):
[10.1109/WCNC.2012.6214279](https://doi.org/10.1109/WCNC.2012.6214279)

[Link to publication record in Explore Bristol Research](#)
PDF-document

University of Bristol - Explore Bristol Research

General rights

This document is made available in accordance with publisher policies. Please cite only the published version using the reference above. Full terms of use are available:
<http://www.bristol.ac.uk/red/research-policy/pure/user-guides/ebr-terms/>

Using Multiple Metrics for Rate Adaptation Algorithms in IEEE 802.11 WLANs

Li Li

Department of Electrical and
Electronic Engineering
University of Bristol
Bristol, BS8 1UB, UK
Email: l.li@bristol.ac.uk

Zhong Fan

Toshiba Research Europe Ltd
Telecommunications Research Laboratory
32 Queen Square, Bristol, BS1 4ND, UK
Email: zhong.fan@toshiba-trel.com

Dritan Kaleshi

Department of Electrical and
Electronic Engineering
University of Bristol
Bristol, BS8 1UB, UK
Email: dritan.kaleshi@bristol.ac.uk

Abstract—IEEE 802.11 Wireless LANs (WLANs) use rate adaptation algorithms (RAAs) to dynamically switch data rates to accommodate the fluctuating wireless channel conditions. Classic RAAs such as ARF and ONOE suffer from rate poisoning and inability to distinguish between collision and packet losses caused by channel errors. Existing approaches in the literature are able to solve only one of the above two issues. In this paper we propose a novel rate adaptation protocol to address both issues for multi-rate wireless networks. The novelty of our approach is to combine the metrics of expected packet transmission time (ETT) and the average number of frozen slots (ANFS) to estimate the quality and level of contention for the current channel. A mathematical model to calculate ETT and ANFS on the fly is presented. Our protocol is simple and practical, which takes into consideration not only link quality and frame loss characteristics, but also impact of collisions during the design. Simulation results show that without the perfect knowledge of current channel condition and any signal strength information, our algorithm can achieve significant performance improvement in terms of end-to-end throughput for different network conditions compared with state-of-the-art link adaptation algorithms.

I. INTRODUCTION

The IEEE 802.11 wireless media standard supports multiple data bit rates at the physical layer (PHY), where the terminal may transmit at a higher rate than the base rate if channel conditions so permit [1]. In order to choose the most appropriate transmission rate, various link adaptation algorithms at the MAC layer have been proposed. The rate adaptation algorithms can be classified into two categories: SNR based or packet retransmission (loss) based [2][3][4][5]. In the SNR based rate adaptation algorithms, the received signal strength information (RSSI) is used as an indicator of link quality, and then a transmission rate is selected based on the average or instantaneous RSSI from a predetermined SNR-rate table. Receiver based auto rate (RBAR) [2] is a typical example of such algorithms. In the packet retransmission based rate adaptation algorithms, the transmitting terminal counts the outcome (either succeeded or failed) of each transmission attempt. Based on the packet transmissions history, the transmitting rate can be adaptively adjusted. Auto rate fallback (ARF) is the first documented bit rate selection algorithm in this category [3].

For aforementioned rate adaptation protocols, there are two common issues in their design, namely rate poisoning effect

and loss differentiation problem, which cause decrease of performance. For a fixed length data frame, a higher data bit rate means a shorter transmission time but a higher risk of channel error, as the frame transmission rate is inversely proportional to the transmission time and proportional to the bit error rate (BER). Thus the design of RAA must take into consideration both transmission time and BER, otherwise it may cause so-called rate poisoning effect. For example, statistics-based RAAs such as ARF [3] and ONOE [4] use past consecutive transmission successes/failures to estimate channel condition and hence select the data bit rate based on this estimation. Because of the nature of their algorithm design, the consecutive transmission successes are rewarded by the increase of data rate, while the consecutive transmission failures lead to the decrease of data rate. Instead of using higher data rate with higher BER, ARF and ONOE tend to use lower data rate to achieve lower BER, which normally results in a longer channel occupancy time. The longer channel occupancy time may in turn cause longer back-off delays and increase the chance of collision, hence resulting in an inefficient medium usage and a poor overall network throughput.

Classical RAAs such as ARF, RBAR, ONOE and SampleRate [6] also lack the ability to distinguish the packet errors due to poor channel quality from errors due to frame collisions. This inability of loss differentiation forces RAAs to misinterpret all packet losses for link errors caused by poor channel condition even the channel condition actually is good. Most of RAAs then lower their transmission rate hence resulting in a poor overall system performance.

There have been a few research work focusing on the solutions to one of the above two issues. For example, SampleRate attacks the rate poisoning effect by applying expected frame transmission time (ETT), while CARA [7] and WOOFF [8] provide approaches to fix the loss differentiation problem by either using RTS/CTS exchange or using the channel busy time (CBT) metric, which requires either specific hardware implementation in NICs (network interface cards) or modification to MAC frames.

Since aforementioned algorithms are only targeting one of the two issues, they may work well under some scenarios but perform poorly under other scenarios. In this paper, we

present a novel rate adaptation algorithm with the combination of transmission time estimation and channel contention estimation. The proposed algorithm provides the method to address the rate poisoning and loss differentiation problems in one go and the simulation results show that our algorithm achieves better performance than current state-of-art approaches in most cases. To achieve this objective, this paper makes the following two main contributions. First, we design a method to accurately estimate the channel state by using ETT and ANFS, which in turn provides a clear image of the network state and identifies network congestion. Secondly, we design a new algorithm called MMRA (Multiple Metrics based Rate Adaptation), which applies the estimated channel and network states to mitigate the negative impact of channel quality and collisions on the rate adaptation mechanism.

The advantages of our approaches are three-fold: first, our algorithm is a simple solution that makes use of locally available information to estimate the packet transmission time and level of contention for the current channel. Our algorithm requires neither precise channel information which is difficult or impossible to obtain in real world, nor any reliable signal strength estimation from the radio interfaces which again is not trivial. Second, our approaches are purely based on MAC layer measurements and do not need cross-layer information. Third, our protocols are practical and can be easily implemented in current IEEE 802.11 NICs without any modifications.

The rest of the paper is organized as follows. Section II describes the system model and the design of the protocol. Section III presents the analysis of simulation results. In Section IV, we conclude the paper.

II. SYSTEM DESIGN

A. System assumptions

We assume a single hop wireless LAN with fully connected topology, where all the nodes are in radio range of each other. In total N terminals are deployed in the network. All terminals are identical and stationary. Each terminal has saturated traffic to transmit to one of its neighbors. Due to the much smaller size of MAC control messages compared to the data packet, the error of the non-data packets is considered negligible. We assume single transceiver at each node and simultaneous transmissions from more than one node will result in collision. Once a source gains the channel access and starts transmitting, we assume that other sources will not transmit until the transmission is over.

In the following, we give details of the proposed rate adaptation protocol. In addition to ETT, we apply the metric of the average number of frozen slots in the back-off process (ANFS). The key function of our algorithms is to use both ETT and ANFS to estimate the quality and contention level for the current channel and identify different channel states so as to use the appropriate transmission rate and hence maximize the throughput. Our approach is based on two main functionalities: the method to calculate ETT and ANFS and to infer the channel state, and a sampling-based rate adaptation algorithm that maximizes the link throughput by choosing the highest

possible data rate. It is worth noting that we use ETT and ANFS proposed from [9] and [10] respectively to leverage their advantages by making them aid each other. Moreover, we adopt more careful and realistic calculation of the metrics and collection of the statistics, and design new rate selection algorithms to cope with more complex packet errors caused by the mixed effect from channel errors and collisions.

B. Motivation and challenges

The major motivation of our work is to design simple link adaptation algorithms with the capability to explicitly address the rate poisoning effect and loss differentiation problem. This requires that the algorithm must take into account not only link quality and frame loss characteristics but also impact of collisions during the design of rate adaptation. Previous work relying on one metric to estimate the channel state may only work well at some scenarios but suffer performance degradation at other scenarios. ETT takes into consideration the mixed effects from wireless channel condition and collisions. Therefore it is a suitable metric to describe the quality of channel condition and can partially indicate the congestion in the channel as well. ANFS is a simple metric, but it can efficiently estimate the contention level in the channel and hence is a lightweight indicator to identify the collision errors.

Although selecting data rate based on the combination of multiple metrics may seem straightforward, the task of selecting the right metrics and integrating multiple metrics in one rate adaption algorithm is not trivial. In addition, to our best knowledge, this concept is novel as no existing studies have used such a combination of metrics in rate adaptation algorithm design.

C. ETT and ANFS description

In this subsection, we will present the calculation of ETT and ANFS for the system model of IEEE 802.11 DCF (distributed coordination function). The system model is extended from the general model proposed in [11]. Using this model, the effects of rate adaptation and packet collision/corruption in the channel can also be taken into account. For simplicity of exposition and without loss of generality, we introduce a notion of virtual time slot and assume that system time is slotted with each time slot of t seconds. This enables us to assume that channels are separated in time slots and to use terms as slots or phases in the remaining of the paper.

In IEEE 802.11 DCF, the overall time duration required to complete a packet transmission is dictated by the back-off procedure. The authors of [9] propose a method in which the expected transmission time (ETT) can be calculated by carefully analyzing the duration and occurring probability of different events taking place at back-off stages. When the time slot is sensed as idle, the back-off timer is decremented by one. If the time slot is sensed as busy due to the channel occupied by other traffic, the back-off timer is frozen. We denote P_{idle} as the probability of a time slot being idle and t_{slot} as the duration of a time slot, while the occurring probability of a busy slot is

$(1 - P_{idle})$ and T_{busy} is the average channel occupation time by other traffic transmissions.

When the back-off timer expires (i.e. it reaches zero), the attempt of packet transmission either fails or succeeds. We denote $P_{fail,r}$ as the packet error probability at bit rate R_r and $P_{suc,r}$ is the probability of a successful packet transmission in a slot. Note P_{fail} represents the transmission failure events due to both packet error and collision. $T_{fail,r}$ represents the duration of a failed transmission in a time slot and $T_{suc,r}$ represents the duration of a successful transmission at bit rate R_r . For basic access scheme (i.e. without RTS/CTS), $T_{suc,r}$ and $T_{fail,r}$ can be computed by:

$$T_{suc,r} = DIFS + SIFS + HEADER + L_{data}/R_r + ACK, \quad (1)$$

$$T_{fail,r} = DIFS + HEADER + L_{data}/R_r + EIFS, \quad (2)$$

where L_{data} is the size of data packet in bits and $SIFS$, $DIFS$, $EIFS$, $HEADER$ and ACK are 802.11 parameters representing the duration of a SIFS, DIFS, EIFS, MAC/PHY header, and ACK frame, respectively. Let N_{retry} denote the maximum number of retry. Then the duration of i th retry with bit rate R_r can be calculated by

$$ETT_{ith,r} = P_{idle} * T_{idle} + (1 - P_{idle}) * T_{busy} + P_{suc,r} * T_{suc,r} + P_{fail,r} * T_{fail,r}.$$

Then the average frame ETT for a particular R_r can be computed by:

$$AETT_r = \alpha AETT_r + (1 - \alpha) ETT_r, \quad (3)$$

where α is chosen as 0.9 in [6]. The next step is to collect parameters such as P_{idle} , $P_{suc,r}$, $P_{fail,r}$, and T_{busy} . From (1) and (2), $T_{suc,r}$ and $T_{fail,r}$ can be calculated off-line once R_r is decided. Most of the parameters can be obtained through statistics collection. During one transmission, P_{idle} can be obtained by counting the number of idle slots during the back-off procedure. $P_{suc,r}$ and $P_{fail,r}$ can be obtained by counting the probability of successful or failed past transmission history. Although it is easy to obtain T_{busy} in simulations, it is almost impossible to collect the accurate channel busy time in real-world scenarios. To keep our algorithm practical, we decide to calculate T_{busy} indirectly by using the number of frozen slots. Therefore T_{busy} and T_{idle} can be calculated as

$$T_{idle} = t_{slot} * NIS, \quad (4)$$

$$T_{busy} = t_{slot} * NFS, \quad (5)$$

where NIS denotes the number of idle slots and NFS denotes the number of frozen slots in the current transmission. Both NIS and NFS can be obtained by keep tracking the number of idle slots and frozen slots during the transmissions.

In order to accurately predict if there is contention in the channel, for each node we calculate the average number of frozen slots ANFS for w frames transmitted. w is the sampling

window size and is defined as 7 in the simulation which is the same as the maximum number of retransmission attempts in 802.11. ANFS is given by:

$$ANFS = \frac{\sum_{i=1}^w NFS(i)}{w}. \quad (6)$$

D. The modules of MMRA

The MMRA protocol consists of the following four modules:

1) Statistics collection: after each frame transmission, frame delivery statistics, such as P_{idle} , $P_{suc,r}$, $P_{fail,r}$, T_{busy} , and T_{idle} are collected and processed as well as the number of successful/failed transmissions of each data rate. In addition, MMRA keeps tracking NFS for each node. Exponentially weighted moving average (EWMA) is used to filter out effects caused by sudden changes in current wireless channel and contention conditions.

2) ETT and ANFS calculation: once receiving the delivery statistics from the above module, this module uses the mathematical model described in the previous subsection to calculate ETT and ANFS for each node. The results are then passed on to the loss differentiation module.

3) Loss differentiation: for a node i finishing transmitting a frame, if $NFS_i > ANFS_i$, it infers that recently contention has increased and the transmission error is more likely due to collision than due to channel error. If $NFS_i \leq ANFS_i$, it indicates that the transmission is failed most likely because of the poor quality of the wireless channel.

4) Rate probing: let S_{smp} denote a set of candidate sampling bit rates which have a loss-free frame transmission time smaller than the current $AETT_r$. When the total number of transmissions is a multiple of 10, the algorithm will randomly choose a data rate in S_{smp} with less than 4 consecutive packet error failures and send the packet using that bit rate instead of the current one. Therefore higher data rates can potentially be used.

E. The MMRA algorithm

MMRA is a rate selection algorithm based on ETT and ANFS metrics and adopts a rate selection approach similar to that in [6]. The algorithm starts with the highest possible data bit rate. Upon the completion of a data frame with the bit rate R_r , ETT_r and $AETT_r$ can be calculated by (1) – (3), as well as NFS_r and $ANFS_r$ by (4) – (6) based on the collected statistics. After the calculation, if MMRA is still in the transmission mode, the current data rate is set to the one with the smallest AETT and S_{smp} is also updated accordingly. Otherwise, MMRA switches to the sampling mode and starts the rate probing procedure.

Note that the ability of loss differentiation enables enhancements to the rate selection and probing design. For example, the data rate with higher number of failures in transmission can still be chosen as probing rate or sending rate if ANFS indicates the errors are mostly contention related. With ETT and loss differentiation described in the above modules, MMRA has the ability to make correct estimation of

data packet size	2000 bytes
Number of nodes	5 - 35
Channel SNR	5 - 30 db
Initial data rate	54 Mbps
Data rate set	[6 12 24 36 54] Mbps
t_{slot}	9 μ s
t_{sifs}	16 μ s
t_{difs}	34 μ s
t_{header}	20 μ s
t_{ack}	42 μ s
t_{eifs}	92 μ s
N_{retry}	7

TABLE I
SIMULATION PARAMETERS

the current channel condition and congestion level. Therefore it can always find and hence switch to the best data rate which yields the highest throughput.

III. PERFORMANCE EVALUATION

In this section, using Matlab we validate our proposed protocol under various scenarios with different channel conditions and traffic loads. We consider a one hop wireless LAN which has various numbers of nodes. All the nodes are identical and static. The aggregated throughput is computed by dividing the total sum of successfully transmitted packet bits by the total duration of transmission time.

The configurations of the simulation and settings of IEEE 802.11a parameters are listed in Table I. To characterize the performance that the proposed algorithm acquires, we compare it with several well-known schemes in the literature in terms of aggregated throughput: SampleRate [6] represents the effect of ETT metric, AARF (adaptive ARF) with FREEZE [5] [10] represents the effect of frozen slot count metric, and CARA [7] represents RTS/CTS loss differentiation.

Due to the lack of space, we only present some typical performance results of our algorithm against various fixed channel conditions and traffic loads. The SNR value denotes the channel condition. As each source has saturated traffic, the level of contention in the network is changed by varying the number of source nodes. With the number of source nodes increasing, the likelihood of collision also increases. In the following figures, “MMRA” denotes the performance of our proposed scheme. “SamRate”, “CARA”, and “AARF-Fz” represent SampleRate, CARA, and AARF with FREEZE respectively.

Figure 1 shows the overall performance of AARF-Fz, SamRate and CARA against our proposed scheme when channel condition is relatively poor so that both channel loss and packet collisions have a mixed effect on packet transmission failures. In this scenario, the overall performance of MMRA is better than that of other protocols, and the improvement over SamRate and AARF-Fz increases with the number of source nodes. When the traffic load in the network increases, the packet errors caused by collision gradually grow. Protocols without the loss differentiation ability like SampleRate suffer

drastic performance degradation as shown in the figure, because it is forced to reduce the data rate to get better delivery results while the channel quality can actually support higher rate. It is also noticed that the throughput of AARF with FREEZE drops as well. This is due to two flaws in AARF and FREEZE design. First, as a lightweight loss differentiator, FREEZE still mistakes some collision errors for channel errors and hence lowers its data rates. Secondly, AARF tends to use lower data rates more often than higher rates as a consequence of rate poisoning. Our algorithm outperforms other schemes in this scenario mainly as a result of the combining efforts from ETT and ANFS. With ETT, our algorithm can use the highest possible data rate for transmissions. Aided by the ANFS scheme, our protocol can filter out packet errors caused by collisions from channel errors, which enables MMRA to stick to the higher data rate thus achieving higher throughput. CARA has a better performance than AARF-FREEZE and SampleRate, but is outperformed by MMRA in this particular scenario because of the heavy overhead introduced by RTS/CTS message exchange.

Figure 2 represents the scenario in which channel conditions are so good that collision errors become the dominating cause of the packet delivery failures. Therefore the performance of the rate adaptation schemes is mainly determined by the ability and efficiency of loss differentiation mechanisms used in their design. As shown in Figure 2, we can see that the performance of AARF-Fz and SampleRate decrease with the number of source nodes increasing. When the network is heavily loaded, CARA and MMRA have almost the same performance and outperform other protocols in terms of throughput. Having a closer look, we can see that MMRA has a slightly higher throughput than CARA. As discussed before, this is mainly because ETT used in MMRA has the advantage in bandwidth efficiency against RTS/CTS message exchange used in CARA. Compared with CARA, our algorithm using ANFS and ETT has almost the same ability of loss differentiation without RTS/CTS overhead.

Figure 3 depicts the performance of different schemes in a network with very poor channel quality such that channel loss contributes to most of the packet delivery errors. As so many frames failed in transmission because of poor channel quality, CARA is forced to turn on RTS/CTS for most of the frame transmissions, which consumes the already very limited bandwidth and leads to longer transmission times. As shown in Figure 3, MMRA outperforms CARA by roughly 20%, which further demonstrates that the combination of ETT and ANFS is more efficient in channel assessment than CARA under this particular scenario.

Finally, Figure 4 gives a view of performance comparison among rate adaptation schemes with different channel conditions and fixed number of nodes. It is observed that MMRA outperforms other three schemes due to the combination effect of ETT and ANFS. Since the network is heavily congested in this scenario, MMRA and CARA have the ability of identifying the packet error and collision error correctly, and effectively achieve better throughput than SampleRate and

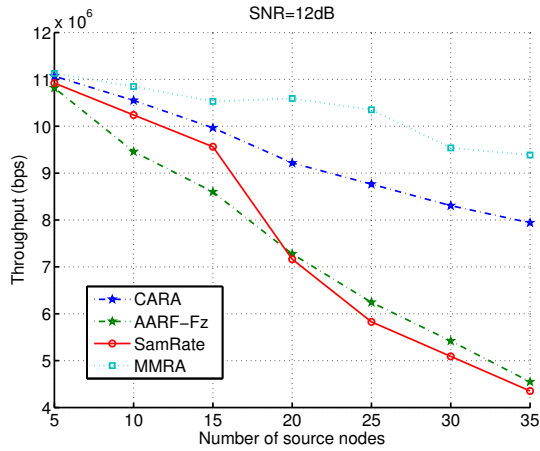


Fig. 1. Throughput vs. number of source nodes

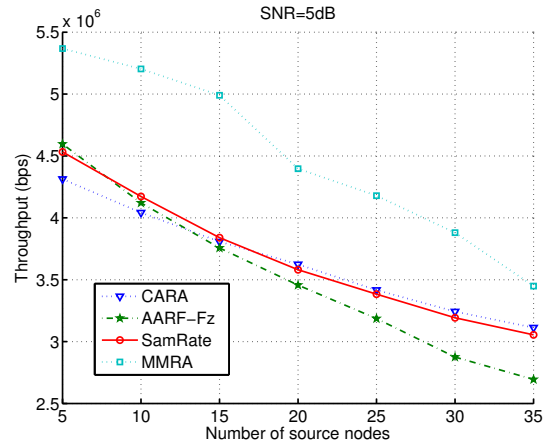


Fig. 3. Throughput vs. number of source nodes

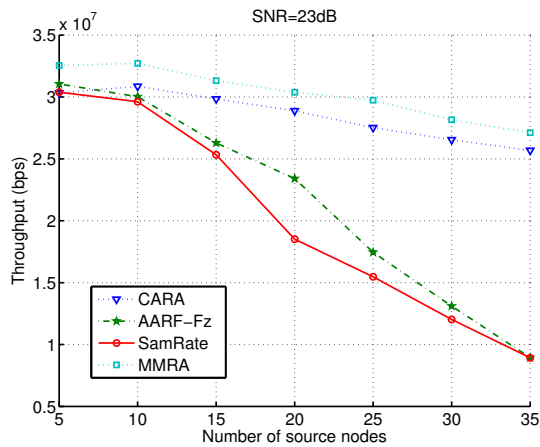


Fig. 2. Throughput vs. number of source nodes

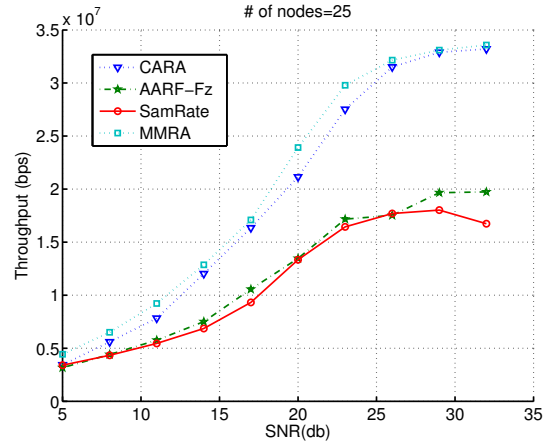


Fig. 4. Throughput vs. various SNR values

AARF-FREEZE. When the packet error is caused by channel error and collision, our scheme is also better than CARA because of reduced overhead. When there is no channel error in transmissions, both schemes have the same performance.

In summary, with the various channel conditions and different traffic load scenarios we have evaluated in this section, we observe that our algorithm always yields the best performance when channel is lossy and traffic loads are heavy. On average, MMRA increases the throughput by 10% than CARA and 100% than AARF-FREEZE and SampleRate in the scenario where channel noise and collision have a mixed effect on packet errors. In the collision dominated scenarios, MMRA has slightly better performance than CARA and both outperform AARF-FREEZE and SampleRate by more than 100% on average when the contention in the channel is severe.

IV. CONCLUSION

In this paper we have presented a rate adaptation algorithm with the ability to assess the channel quality and contention condition for IEEE 802.11 WLAN networks. The combination of multiple metrics such as ETT and ANFS enables nodes to adjust their transmission rates according to the conditions

of the underlying link in such an efficient way that no extra overhead is introduced. Simulation results have shown our protocol outperforms the other well-known schemes in terms of throughput in different channel quality and contention conditions. Furthermore, due to its simplicity of using only locally available information, it can be easily implemented without any change required to the current IEEE 802.11 standard.

ACKNOWLEDGMENT

We would like to thank A. Munro and J. He for helpful discussions. This research is supported by Toshiba TRL.

REFERENCES

- [1] IEEE Computer Society, "Part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications," *ANSI/IEEE Std 802.11*, 1999.
- [2] G. Holland, N. Vaidya, and P. Bahl, "A rate-adaptive MAC protocol for multi-hop wireless networks," in *ACM Mobicom*, 2001.
- [3] A. Kamerman and L. Monteban, "WaveLAN II: A high-performance wireless LAN for the unlicensed band," *Bell Labs Technical Journal*, vol. 2, no. 3, 1997.
- [4] Madwifi Project, *Madwifi*. <http://sourceforge.net/projects/madwifi>.
- [5] M. Lacage *et al.*, "IEEE 802.11 rate adaptation: A practical approach," in *ACM MSWIM*, 2004.

- [6] J. Bicket, "Bit-rate selection in wireless networks," MIT, 2005.
- [7] J. Kim *et al.*, "CARA: collision-aware rate adaptation for IEEE 802.11 WLANs," in *IEEE Infocom*, 2006.
- [8] A. Acharya *et al.*, "Congestion-aware rate adaptation in wireless networks: A measurement-driven approach," in *IEEE SECON*, 2008.
- [9] S. Wang and A. Helmy, "BEWARE: Background traffic-aware rate adaptation for IEEE 802.11 MAC," in *IEEE WoWMoM*, 2008.
- [10] A. Kulkarni *et al.*, "FREEZE: Rate adaptation in wireless LANs using channel contention estimates," in *IEEE COMSNETS*, 2011.
- [11] J. He *et al.*, "Modeling link adaptation algorithm for IEEE 802.11 wireless LAN networks," in *IEEE ISWCS*, 2006.