



**BEWARE: Background Traffic-Aware Rate Adaptation for
IEEE 802.11**

Journal:	<i>IEEE/ACM Transactions on Networking</i>
Manuscript ID:	TNET-00398-2009
Manuscript Type:	Original Article
Date Submitted by the Author:	25-Sep-2009
Complete List of Authors:	wang, shao-cheng; University of Florida, CISE Helmy, Ahmed; UFL, CISE
Keywords:	802.11, rate adaption

BEWARE: Background Traffic-Aware Rate Adaptation for IEEE 802.11

Shao-Cheng Wang and Ahmed Helmy, *Member, IEEE*

Abstract—IEEE 802.11-based devices employ rate adaptation algorithms to dynamically switch data rates to accommodate the fluctuating wireless channel conditions. Many studies observed that, when there are other stations transmitting in the network, existing rate adaptation performance degrades significantly due to its inability to differentiate losses between wireless noise and contention collisions. In this paper, we first conduct a systematic evaluation on the effectiveness of various rate adaptation protocols, which try to address this issue by exploiting optional RTS frames to isolate the wireless losses from collision losses. We observe that these existing schemes do not perform well in many background traffic scenarios, and can mislead the rate adaptation algorithms to persist on using similar data rate combinations regardless of background traffic level, thus result in performance penalty in certain scenarios.

The fundamental challenge is to dynamically adjust the rate selection decision objectives with respect to different background traffic levels, as well as fluctuating wireless conditions. In light of such observations, we design a new Background traffic aware rate adaptation algorithm (BEWARE) that addresses the above challenge. BEWARE uses a mathematical model to calculate on-the-fly the expected packet transmission time based on current wireless channel and background traffic conditions. We implement BEWARE design in Linux-based driver and the test-bed experiment results show that BEWARE outperforms other rate adaptation algorithms by up to 150% in various indoor and outdoor scenarios.

Index Terms—802.11, CSMA/CA, rate adaptation.

I. INTRODUCTION

WITH the multiple transmission data rates specified in the IEEE 802.11 standards, IEEE 802.11-based stations implement rate adaptation algorithm (RAA) to dynamically select the best data rate that yields the highest performance in the given wireless channel conditions. The key challenges are that RAA must not only accurately estimate the channel condition in order to infer the most suitable data rate, but also be very responsive to the rapidly fluctuating wireless channel dynamics. Several approaches have been proposed [1]-[8] to use various metrics such as received signal strength, local ACKs (Acknowledgements) information, and packet statistics to design a RAA in addressing the above challenges. The

effectiveness of RAAs has been extensively evaluated under various wireless channel conditions, when there is only one station in the network. On the other hand, in multiple-user environments, several studies [9][10] reported that the performance of some types of RAAs, e.g. Automatic Rate Fallback (ARF)[1], degrades drastically because the RAA mistakenly lowers its data rate when the consecutive frame losses are caused by collision losses and not by wireless channel losses.

There have been a few studies attempting to aid rate adaptation algorithms in dealing with the collision effects in multi-user environment. Their key idea is to provide RAAs the ability to “differentiate” between wireless channel losses and collision losses. For example, by assuming the only cause for the data frame transmission failure after a successful Request-To-Send/Clear-To-Send (RTS/CTS) exchange is due to channel error not collision, [9][10] use RTS/CTS to filter out collision losses from rate decision process. On the other hand, [10][11] suggest to add extra frames and fields to explicitly notify the sending station whether the transmission failure is due to collision or wireless channel errors.

While these proposals provide significant improvements compared to RAAs without loss differentiation capability, it is unclear whether loss differentiation is optimal to deal with all kinds of mixed wireless and collision loss scenarios. The fundamental problem is, as we will show later in this paper, that background traffic from other contending stations changes the throughput ranking of the operating data rates. In other words, under the same wireless condition, the data rate yielding the highest throughput in no background traffic scenarios is not necessarily the best one when background traffic exists. This is particularly problematic for existing loss differentiation schemes as they filter out all collision losses for RAA, the RAAs become insensitive to the throughput ranking changes caused jointly by wireless losses and collision losses, thus resulting in performance degradation.

In this paper, we design a new Background traffic aware Rate Adaptation Algorithm (BEWARE) that explicitly addresses the mixed effects from wireless and collision losses. The main contributions of this paper are: i. to systematically evaluate the performance of RTS-based loss differentiation in different mixed wireless and collision losses scenarios. We identify when and why RTS-based loss differentiation does not work in certain scenarios, ii. we use the insight in these systematic evaluations to identify a novel metric – the

1 expected packet transmission time – to explicitly address the
 2 mixed effects from wireless and collision losses on all
 3 available data rates. We propose an online algorithm to
 4 estimate this parameter for all data rates and embed this
 5 information into the RAA design as the key rate decision
 6 maker, and iii. to implement our design into a real-world
 7 driver and compare the performance of BEWARE with other
 8 RAAs with and without loss differentiation, and observe up to
 9 250% and 25% performance improvement, respectively.

10 The rest of the paper is organized as follows. Section II
 11 reviews the existing RAAs and related loss differentiation
 12 approaches. Section III evaluates the performance of existing
 13 RAAs and loss differentiation schemes. Section IV presents
 14 the design of our background traffic aware rate adaptation
 15 algorithm, and Section V discusses simulated performance
 16 under various background traffic scenarios. Section VI
 17 describes the test-bed implementation and performance
 18 comparisons in real-world scenarios. Section VII concludes.

21 II. RELATED WORK

22 In this section, we briefly review the existing rate adaptation
 23 algorithms (RAAs) and related loss differentiation schemes
 24 that help RAAs deal with collisions in multiple-user
 25 environment

26 A. Existing Rate Adaptation Algorithms

27 There have been quite a few RAAs proposed by academia
 28 and industry. They can be broadly classified into three
 29 categories based on the information they collect for rate
 30 selection decisions: 1) statistics based RAAs, 2) received
 31 signal strength (RSS) based RAAs, and 3) hybrid RAAs.

32 **1) Statistics-based RAAs:** Based on the statistics the RAA
 33 uses for rate decisions, we can further categorize this class of
 34 RAAs into three different approaches. i) Retry-based rate
 35 adaptation [1][2] uses consecutive transmission
 36 successes/losses (e.g. 10 and 2 in ARF [1], respectively) as the
 37 indicator of good/bad wireless condition. However, despite its
 38 easy design, previous study [4] has shown that, due to
 39 randomness of the wireless loss behavior, there is very weak
 40 correlation between past consecutive transmission
 41 successes/losses and future channel condition. Thus, this
 42 approach tends to yield pessimistic rate estimations. ii) Frame-
 43 Error-Rate(FER)-based rate adaptation [3][4] calculates FER
 44 by the ratio of the number of received ACK frames to the
 45 number of transmitted frames. The RAA decreases and
 46 increases the operation data rate if FER exceeds or falls below
 47 some pre-determined thresholds. However, as wireless
 48 channels are affected by many factors such as multi-path,
 49 channel fading, and obstructions, the major drawback for FER-
 50 based RAA is the inadequacy of using one set of pre-
 51 determined FER thresholds in all circumstances. iii)
 52 Throughput-based rate adaptation [5] calculates each data-
 53 rate's throughput based on the packet length, bit-rate, and the
 54 number of retries collected during a predefined decision
 55 window (~1 sec). The major drawback of this approach is that,

as the decision window has to be large enough to collect
 meaningful statistics, it causes the rate adaptation algorithm to
 be less responsive to sudden wireless condition changes.

2) Signal-strength-based RAAs: This class of RAAs [6][7]
 relies on wireless signal strength information, such as
 Received Signal Strength Indicator (RSSI) or Signal-to-Noise
 Ratio (SNR), to make the rate adjustment decisions. The
 RAAs pick the data rate by a pre-determined mapping between
 the received signal strength and throughput. However, in
 reality, such mapping is highly variable and a model
 established before-hand may not be valid in many
 environments later. Meanwhile, signal-strength-based RAAs
 have to overcome the communication issue of piggybacking
 the signal strength measurement taken at the receiver side to
 sender so that sender can adjust the data rate accordingly. One
 has to either use explicit signaling [7], which is incompatible
 with the IEEE 802.11 standard, or assume the channel is
 symmetric [6], which is clearly not the case in many real-world
 scenarios.

3) Hybrid RAAs: In this approach, RAA [8] collects both
 frame transmission statistics and received signal strength, and
 uses a statistics-based controller as the core rate adaptation
 engine. The rate decision can be overridden by signal strength
 based controller if it detects sudden changes in the received
 wireless signal strength. As hybrid RAA design still assumes
 symmetric wireless channel and pre-established RSSI-to-rate
 thresholds, this approach is not immune from the drawbacks
 we discussed in signal-strength-based RAAs section.

In summary, all types of RAAs strive to obtain accurate
 channel estimations from different kinds of loss characteristics
 and decide when to decrease or increase the rate. However, in
 multiple-user environments, packet collisions constitute new
 source of frame loss. None of these RAAs explicitly address
 this issue. In the next section, we review several proposals that
 attempt to aid RAAs in dealing with collision effects.

56 B. Loss differentiation for rate adaptation

57 Previous studies reported that, because ARF treats collision
 58 losses no different than wireless losses, ARF excessively
 59 decreases its rate upon contention collisions, even when
 60 wireless channel is close to perfect. As this effect causes
 severe performance degradation for ARF when background
 traffic exists, we refer this effect as "rate poisoning". There
 have been two approaches to aid rate adaptation algorithms in
 differentiating wireless losses from collision losses. i) Loss
 differentiation by RTS/CTS: With RTS/CTS exchanges
 preceding data transmissions, [9] and [10] assume that the only
 cause for the data frame transmission failure after a successful
 RTS/CTS exchange is due to channel error and not collision.
 Therefore, filtered by RTS/CTS, RAA rate decision process
 reacts only on wireless losses and is no longer affected by the
 collision effect. Kim et al. [9] further propose Collision-Aware
 Rate Adaptation (CARA) to reduce the extra RTS/CTS
 overhead by selectively turning on RTS/CTS after data frame
 transmissions fail at least once without RTS/CTS. The data
 rate is increased as the consecutive success count reaches 10,

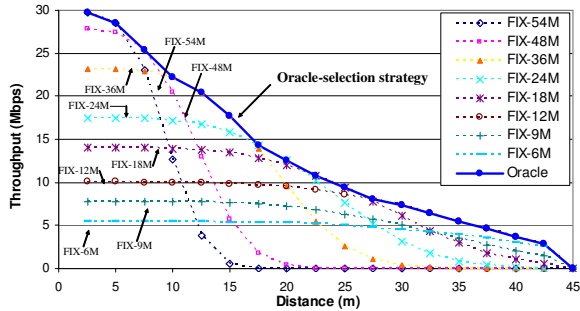


Fig. 1. Throughput versus distance for IEEE 802.11a data

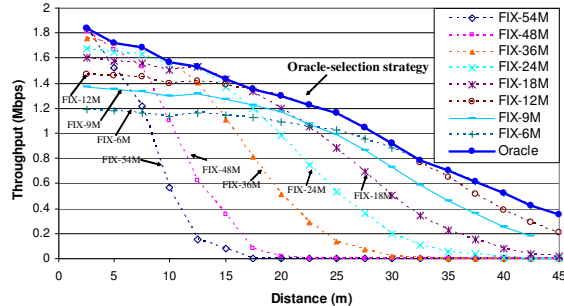


Fig. 2. Throughput versus distance for IEEE 802.11a data rates, with 12 background traffic stations

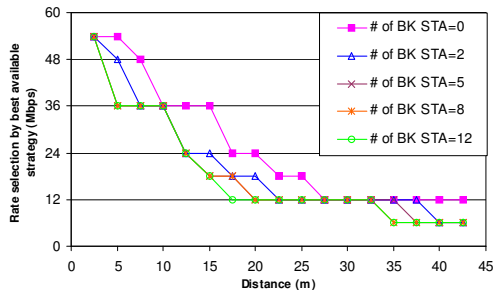


Fig. 3. Oracle-selected data rate under different number of background traffic stations

similar to ARF. ii) Loss differentiation by explicit notification: [10] and [11] propose to add extra frames and fields to explicitly notify the sending station of the source of losses. However, both proposals require incompatible changes to the IEEE 802.11 standard and thus are not favorable for real-world deployments.

In summary, loss differentiation is the dominating approach for RAAs in dealing with collision effects when there are other stations transmitting traffic in the network. However, it is not clear whether loss differentiation is sufficient to guide RAAs to perform well in various multiple-user environments with mixed wire-less loss and contention conditions. As we will show later in the paper, while RTS-based loss differentiation works in certain circumstances, we also find other scenarios that RTS-based loss differentiation performs poorly.

III. PERFORMANCE OF RATE ADAPTATION ALGORITHMS WITH BACKGROUND TRAFFIC

In this section, we first explain briefly how IEEE 802.11 rate adaptation works. In particular, we analyze how rate selection objective varies with the level of background traffic.

Furthermore, we systematically evaluate the performance of various RAAs with RTS loss differentiation schemes under different scenarios, including varying number of stations in the network and the distance between stations and access point. As we will show in this section, it is critical to examine how and why these RAAs do not perform well with background traffic. By such an investigation, we motivate the need for a new RAA that does takes background traffic into consideration, and we gain insight into how to design such a RAA

A. IEEE 802.11 rate adaptation with different level of background traffic.

To visualize the throughput-distance tradeoff among multiple data rates employed by IEEE 802.11 standard, in Fig. 1, we use ns-2 [12] to simulate an 802.11a station's maximum throughput as it moves away from the access point (AP) in Ricean fading environment [13]. As seen in Fig. 1, among the 8 data rates available in IEEE 802.11a, higher data rates can achieve higher throughput, but their transmission ranges are shorter. The crossing points of two adjacent data rates indicate that, at a given location, the error rate of the high data rate is becoming high enough such that it is more favorable to use the next lower data rate. Clearly, the rate adaptation mechanism should attempt to follow such transitions as close as possible to select the best data rate according to the current wireless channel condition experienced by the link. Ideally, if a rate adaptation mechanism has perfect knowledge of the current network condition, its data rate selections follow closely the outer envelope (plotted as thick solid line) of Fig 1. This way, the throughput obtained by the rate adaptation mechanism is always maximized given a particular channel condition. We will refer to this outer envelope concept as the "oracle-selection strategy" and its performance as the maximum throughput throughout the paper.

On the other hand, Fig. 2 plots the performance of the same data rate set under the same wireless channel condition, but with 12 other stations continuously transmitting background traffic in the network. We can see that, the shape of staircase like throughput-distance curves changes, and the rates selected by the oracle-selection strategy also change for the same location. It is because the data frames transmitted by any data rate are subject to both wireless losses and also collision losses caused by medium contentions with other stations. In other words, the extra backoff time spent in medium contentions and collisions changes the crossing points of two adjacent data rates, and thus the rate switching strategy. This combined effect also changes the performance ranking of data rates for a given location. Fig. 3 further illustrates this effect by plotting the rate selections by the oracle-selection strategy when operating with different number of saturated background traffic stations. The rate selected by the oracle-selection strategy varies widely with background traffic intensity. In other words, the rate adaptation strategy that works well in one background traffic scenario may not work in another background traffic scenario, hence the rate adaptation mechanism needs to explicitly address this phenomenon. With

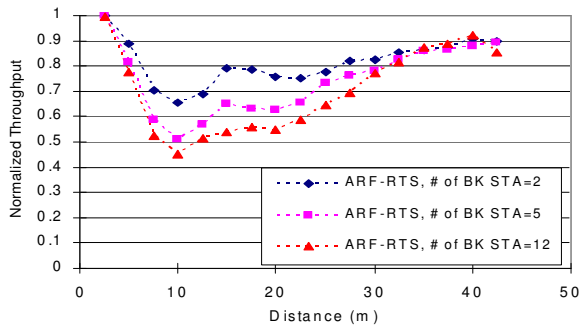


Fig. 4. Normalized throughput for ARF-RTS, with various number of background traffic stations in RTS access mode

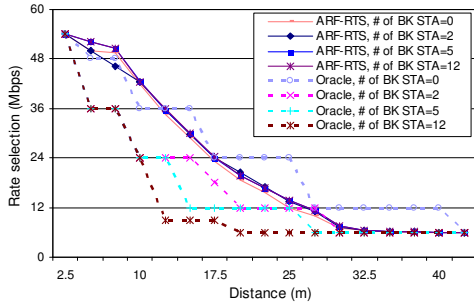


Fig. 5. Data rate selection for ARF-RTS and oracle-selection strategy, with various number of background traffic stations in RTS access mode

this observation, we argue that *it is very critical for rate adaptation designs to be aware of background traffic changes, in addition to fluctuating wireless conditions, and adjust their rate selection strategies accordingly; otherwise they will suffer from serious performance degradation.*

B. Performance of RAAs with RTS loss differentiation

Previous studies have reported superior performance of ARF with RTS on over that with RTS off. However, those studies did not provide systematic investigation into whether RAA with RTS differentiation achieves the optimal throughput in all situations and why. In this subsection, we compare the performance of RAAs with RTS differentiation to the *oracle-selection strategy*.

With the same simulation settings in the previous section, we first place all stations at 2.5m away from the AP and turn on RTS for all stations. We isolate the effects of RTS loss differentiation in performance comparisons by enabling only one station with RAA on, and all other background traffic stations with fixed data rate. In Fig. 4, we plot the RAA enabled station's throughput (normalized against the maximum throughput) as it moves away from the AP. We can observe from Fig. 4 that, when the RAA-enabled station is close to the access point, ARF with RTS (ARF-RTS) performs almost the same as the *oracle-selection strategy*, regardless of how many stations transmitting background traffic in the network. However, as the RAA-enabled station moves away from the access point, we can see that ARF-RTS starts to lose track of the best available rate. In addition, with more background traffic stations transmitting in the network, ARF-RTS station's

performance deviates further from the maximum throughput. Finally, ARF-RTS station's performance rises back to ~90% of maximum throughput when the station is reaching the transmission edge (>35m) of the AP.

To further explain such scenario, we plot Fig. 5 to illustrate the rate selection breakdowns of ARF-RTS along with that of *oracle-selection strategy*, as distance to access point increases. We can see that the rate selections of ARF-RTS remain almost the same regardless the number of background traffic stations in the network, as opposed to the rate selections of *oracle-selection strategy* that vary widely with background traffic level as we discussed above. This is because RTS frames isolate the wireless losses from collision losses and make the rate decisions solely on wireless losses. As a result, RAAs become insensitive to the throughput ranking changes caused jointly by wireless losses and collision losses, and persist in using the rate selections that is only suitable in no background traffic scenarios. On the other hand, when the RAA-enabled station is far away from AP, the only operable data rate is the lowest rate. In this case, ARF-RTS's rate selection coincides with the *oracle-selection strategy* and thus performs close to 100% of the maximum throughput. We further examine the rate selections of other statistics-based RAAs (e.g. ONOE [14], Sample-Rate [5], and RRAA-basic [4]) with RTS-on in background traffic scenarios, and find that the same phenomenon exists. It follows that *turning on RTS misleads RAAs into using rates only suitable for no-background-traffic in scenarios with background traffic, where these rates are not always optimal.* As a result, RTS loss differentiation only works well when the rate selections are similar for all other background traffic scenarios.

In summary, we have made the following two important observations in this section: *i) The oracle-selection strategy varies significantly with the level of background traffic. We argue that any rate adaptation mechanism should be aware of such change at the presence of background traffic, or it will suffer from serious performance degradation. ii) We show that none of the existing RAAs we have investigated perform well in every background traffic scenario. We see that even RTS loss differentiation can hurt the performance in some situations.*

IV. BEWARE DESIGN

From the observations we made in the previous section, we know that the key for RAA algorithm to perform well in background traffic scenarios is to incorporate not only wireless channel statistics but also background traffic conditions as indicators for the effectiveness of each available data rate. As a result, in this section, we present the design of BEWARE, a Background traffic aWaAre RatE adaptation algorithm for IEEE 802.11-based MAC. The center part to this design is to use a mathematical model to calculate the expected packet transmission time of each data rate that considers the combined costs of wireless channel errors and background traffic contentions as we discussed in Sec. III. The rate

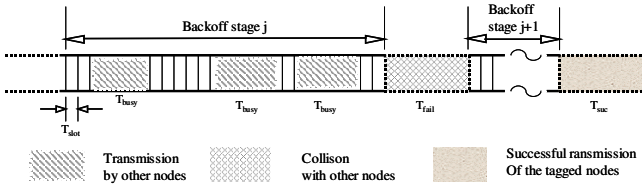


Fig. 6. Packet transmission and collision events during IEEE 802.11 MAC backoff

selection engine then uses this metric to find the data rate that yields the highest throughput in the given wireless channel and background traffic condition. The goals to design such rate selection strategy are two-fold: it has to be robust against any degree of background traffic; meanwhile, it also has to be responsive to random and even drastic wireless channel changes.

In Sec. IV.A, we describe the mathematical model for expected packet transmission time calculations and the rationale of why the expected packet transmission time can be a good rate selection metric. Then, in Sec. IV.B, we present the rate adaptation engine.

A. Packet transmission time estimation

The core of BEWARE design is the estimation for the expected packet transmission time of each data rate, with the consideration of mixed effects from wireless channel condition and collisions. In CSMA/CA-based 802.11 MAC, the overall time duration required to complete a packet transmission is dictated by the backoff procedure. Therefore, as shown in Fig. 6, we calculate the expected packet transmission time by carefully analyzing the duration and occurring probability of different events take place at backoff stages, as follows.

1) When the backoff timer decrements, the time slot is either sensed as idle (for T_{slot} , the length of one time slot) or as busy occupied by background traffic transmission (for T_{busy} , the average medium occupation time used by background traffic transmissions). We define P_{busy} to be the probability that, at a given time slot, the backoff timer is frozen due to busy medium in carrier sensing. It follows that the occurring probability of idle slot and busy slot is $(1 - P_{busy})$ and P_{busy} , respectively.

2) When the backoff timer expires (i.e. decrements to zero), the attempt of packet transmission either fails (after T_{fail}) or succeeds (after T_{succ}). We define P_{fail} to be the frame error probability. It follows that the occurring probability of packet failure and success is P_{fail} and $(1 - P_{fail})$, respectively. Note that P_{fail} represents the transmission failure events due to various packet failure sources such as collisions, channel fading, interference, and hidden terminals.

Once these parameters are collected, we can construct a mathematical model to calculate the occurring probability for combinations of all different backoff events throughout all backoff stages. We first define the occurring probability $F_{k,n-k}^j$ that, in any single backoff stage j with backoff timer selected from 0 to W_j (maximum number of backoff slots in stage j), there are exactly k busy time slots and $(n-k)$ idle slots:

$$F_{k,n-k}^j = \frac{1}{W_j} C_k^{i+k} P_{busy}^k (1 - P_{busy})^{n-k}, \quad 0 \leq k \leq n \leq W_j. \quad (1)$$

Moreover, we know that any combination of busy and idle slots can have a cumulative effect from successive backoff stages. Therefore, we then define $S_{k,n-k}^j$ for probability of backoff counter being frozen $(k-j)$ times and idle $(n-k)$ times that up to back off stage j (which implies packet transmission failed j times),

$$S_{k,n-k}^j = \begin{cases} (1 - P_{fail}) F_{k,n-k}^0, & \text{for } j = 0, \quad 0 \leq k \leq n \leq W_0 - 1. \\ (1 - P_{fail}) P_{fail} \sum_{i=0}^{j-1} \left(\sum_{l=0}^{n-k} S_{l,n-k}^{j-1} \times F_{k-l,n-k-i}^j \right), & \text{for } 1 \leq j \leq m-2, \quad j \leq k \leq n \leq \sum_{i=0}^j W_i - 1. \\ P_{fail} \sum_{i=0}^{j-1} \left(\sum_{l=0}^{n-k} S_{l,n-k}^{j-1} \times F_{k-l,n-k-i}^j \right), & \text{for } j = m-1, \quad m-1 \leq k \leq n \leq \sum_{i=0}^{m-1} W_i - 1. \end{cases} \quad (2)$$

Note that m in this equation is the maximum number of retries specified in the standard.

$S_{k,n-k}^j$ includes all possible cases, from combination of previous stage(s) to the current stage, which result in $(n-k)$ idle slots, $(k-j)$ busy slots, and j failed transmission periods. In addition, the time points when such combinations happen can be characterized by,

$$T_{k,n-k}^j = (k-j) * T_{busy} + j * T_{fail} + (n-k) * T_{slot} + T_{succ} \quad \text{for } 0 \leq j \leq m-1, \quad j \leq k \leq n \leq \sum_{i=0}^j W_i - 1 \quad (3)$$

As a result, the expected packet transmission time can be expressed by

$$T_{avg} = \sum_{k=0}^N \sum_{n=k}^N \sum_{j=0}^{m-1} (S_{k,n-k}^j * T_{k,n-k}^j), \quad \text{where } N = \sum_{i=0}^{m-1} (W_i - 1). \quad (4)$$

Once the expected packet transmission time is obtained, it is sent to the rate selection module for rate selection decisions as we shall describe in the next subsection. We note that the detailed derivations and evaluations of the accuracy of such model can be found in our previous work [17]. In addition, we've also shown that the throughput of the tagged node is inversely proportional to the expected packet transmission time¹. Therefore, we can expect the above mechanism to be a good metric for the rate selection decisions.

On the other hand, we can see that the average packet transmission time is a function of several parameters from the environment, i.e., P_{busy} , P_{fail} , and T_{busy} . In the following, we show how our model captures the mixed effects from

¹ $Throughput = \frac{Avg_packet_size}{Expected_packet_transmission_time}$ [17]

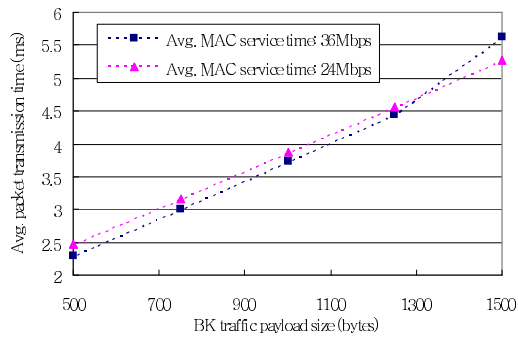


Fig. 7. Average packet transmission time of two adjacent data rates when changing background traffic payload size.

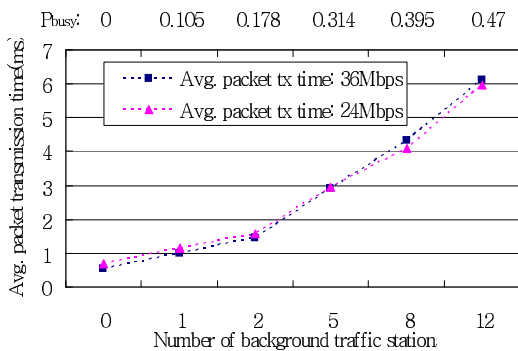


Fig. 8. Average packet transmission time of two adjacent data rates when changing number of background traffic stations.

background traffic and wireless channel losses and make rate switching decisions accordingly between two adjacent data rates.

In Fig. 7, we plot the average packet transmission time of two adjacent data rates in IEEE 802.11a standard, 36Mbps and 24Mbps, with changing the background traffic stations payload sizes but keeping all other parameters fixed. We can see that average packet transmission time of 36Mbps becomes larger than that of 24Mbps after background traffic payload size increases to more than 1300bytes. Recall from the derivations above (Eq. 3 and 4), change in background traffic payload size corresponds to change in T_{busy} , and in turn the length of busy medium slots in backoff stages. In other words, as T_{busy} increases, busy medium slots become longer in each backoff stage, and consequently the backoff stage length is longer. As a result, as the higher data rates are more vulnerable to have more backoff stages due to high wireless loss rates (P_{fail}), longer backoff stages cause the expected packet transmission time of the higher data rate to grow faster than that of the lower data rate. Therefore, there is a crossing point where the expected packet transmission time of 36Mbps becomes higher than that of 24Mbps as T_{busy} increases. It follows that beyond such point, the performance for 24Mbps packets is better than 36Mbps packets even the wireless conditions for both data rates remain unchanged. It is essential to note, for RAAs that only consider wireless loss effects in rate decisions, that they can not capture the above performance crossing point caused by background traffic changes and make rate switching

decisions accordingly.

Similarly, Fig. 8 plots the average packet transmission time of 36Mbps and 24Mbps with the scenarios that there are different numbers of background traffic stations in the network. Note that all other parameters including wireless loss conditions remain unchanged in these scenarios. We can also see a crossing point where average packet transmission time of 36Mbps becomes larger than that of 24Mbps when there are more than 5 background traffic stations in the network. The reason is similar to what was discussed above. More background traffic stations correspond to larger P_{busy} , which cause more number of busy medium slots in each backoff stage, and in turn longer overall backoff duration. This also explains the data rate performance ranking changes in different background traffic scenarios that we observe in Sec. III.

Although using the *expected-packet-transmission-time* as rate selection metric may seem straightforward, it became clear only after our thorough and systematic investigations (in Sec. III) on how and why various existing RAAs do not perform well with background traffic. In addition, this concept is novel as no existing studies, to the best of our knowledge, have used such a rigorous metric in RAA design.

B. Rate selection engine

In this section, we describe the high-level design of how BEWARE makes rate selection decisions. As shown in Fig. 9, the BEWARE design can be broken down into the following tasks:

1) Statistics collection/processing: After the packet transmission completes, transmission environment statistics, including T_{busy} , P_{busy} , and P_{fail} , are collected and processed by exponentially weighted moving average (EWMA) to smooth out the biases to the abrupt changes in current wireless channel and collision conditions. In addition, BEWARE keeps track other statistics such as number of successful/failed packets of different data rates.

2) Expected packet transmission time calculation: With the environmental parameters collected in the above module, this module use the mathematical model described in previous subsection to calculate the expected packet transmission time. The resultant expected packet transmission time are updated with recent history values by EWMA and fed into rate selection module for processing.

3) Rate probing: Periodically, BEWARE sends packets at a data rate other than the current one to update the expected transmission time of other data rates. In order to avoid the common rate-probing pitfalls reported in [4], BEWARE adopts various measures to ensure probing other data rates is not done very often and the cost is not too high. That is, BEWARE limits the frequency of packet probing to a fraction (~5%) of the total transmission time. BEWARE also limits the number of retries allowed for probing packets to 2 to save costly waiting time for unsuccessful probing. In addition, BEWARE does not probe data rates that suffer from excessive failures for most recent packet attempts (4 recent successive packets have been unacknowledged), and those whose

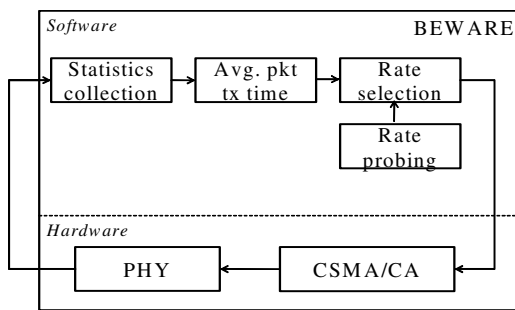


Fig. 9. Structure of BEWARE design

expected transmission time with no background traffic already exceed the expected transmission time of current operating data rate.

4) Rate selection decisions: The rate selection module constantly compares the expected packet transmission time of current data rate and that of others, and decides to change operating data rate whenever it finds a data rate yields the shorter transmission time (and thus highest throughput). BEWARE also implements a short-term frame loss reaction mechanism in case wireless channel conditions change too rapidly. That is, the rate selection module forces data rate to decrease one level when the packets exhaust all retries for three times consecutively.

C. Discussion

We note that most of the parameters required for calculating expected transmission time can be directly obtained from passive channel activity monitoring, which does not incur any extra overhead. To be more specific, we can determine P_{fail} by counting the ratio of failed packet transmission attempts and total packet transmission attempts. We also obtain P_{busy}/T_{busy} by keeping track of the number/duration of experienced busy medium slots, respectively. On the other hand, T_{fail} and T_{succ} are directly determined by the operating data rate and T_{slot} is specified in different version of IEEE 802.11 standard. In practice, it may be difficult to obtain some of these parameters accurately due to implementation complexity in real devices. We can consider alternative approaches [15][16] by using number of consecutive idle slots between two busy slots to estimate P_{busy} and P_{fail} .

On the other hand, one may argue that the expected packet transmission time can be directly obtained by keeping track of the medium access time of every packet without involving time-slot level channel monitoring. While this approach has been proposed in multi-hop wireless mesh network routing studies [18][19], it may not be suitable for MAC layer rate adaptation decisions for the sampling granularity it provides. In other words, the statistics averaged through potentially dozens of packets in the past may not be able to provide the most up-to-date channel information for the rate adaptation decisions that are made on a per-packet basis. In addition, one may argue that collecting time-slot level statistics (i.e. P_{busy} and T_{busy}) might prevent the stations from going into sleep mode, which is critical for energy savings. We can optimize

the energy consumptions by collecting the statistics only when the station has packets to send. Exploring the tradeoff between energy savings and collecting most up-to-date statistics is one of the topics in our future work.

V. PERFORMANCE EVALUATION

In this section, we use the network simulator ns-2 [12] to evaluate the performance of BEWARE and other RTS-based loss differentiation RAAs, including ARF with RTS/CTS (as referred to ARF-RTS) and CARA-1 under various mixed wireless and background traffic scenarios

A. Simulation setup

We enhance the ns-2 simulator to support 802.11a Physical layer (PHY) and port various RAAs from previous studies or the real-world driver implementations [14]. We simulate scenarios in an infrastructure-based network, which contains one Access Point (AP) and a number of static wireless stations spreading in the network. We consider realistic wireless channel conditions by using Ricean fading model. We first fix the K factor at 6 ($K=6$) and environment maximum velocity $v=10$ m/s. We later discuss the effects of different fading parameters in Sec. V.E. The traffic sources are UDP flows unless stated otherwise.

B. Performance of single station with varying distance

We first focus on RAAs' performance with varying distance under background traffic scenarios. We place 2~12 stations on a circle around the AP within 2.5 meter radius, and all stations transmit UDP background traffic with RTS access mode. The transmission data rate of background traffic stations is locked at 54Mbps because of their proximity to the AP. We then add one RAA-enabled station in the network and measure the RAA's performance by varying the distance between RAA-enabled station and the AP. We show results with 12 stations transmitting background traffic as an example in Fig. 10, while results with other number of background traffic stations show a similar trend. In all cases, the performance of BEWARE follows closely the *oracle-selection strategy* by less than 10% in throughput, and the performance of CARA-1 trails behind BEWARE by another 10%-15%. On the other hand, similar to what we discuss in Section III.B., the performance of ARF-RTS significantly deviates from the *oracle-selection strategy* when the distance from station to AP is close-by to intermediate (5m~35m). This is because, in this range, the rate selections for no background traffic deviate significantly from the optimal rate selections for this background traffic scenario. As we discussed in Section III, ARF with RTS loss differentiation suffers from performance degradations by continuing to use the rate selections only suitable for no background traffic.

C. Performance of single station with dynamically changing background traffic

In this subsection, we further investigate how different RAAs adapt with dynamically changing background traffic

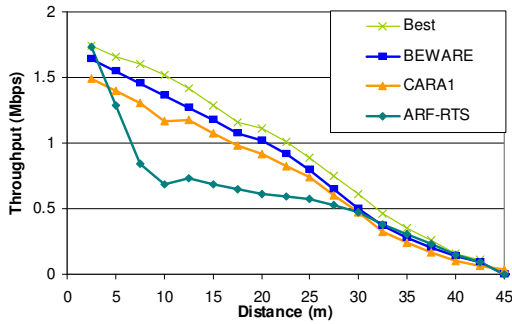


Fig. 10. Throughput comparison for Best (oracle-selection strategy), BEWARE, CARA1, and ARF with RTS/CTS, with 12 background traffic stations in RTS access mode

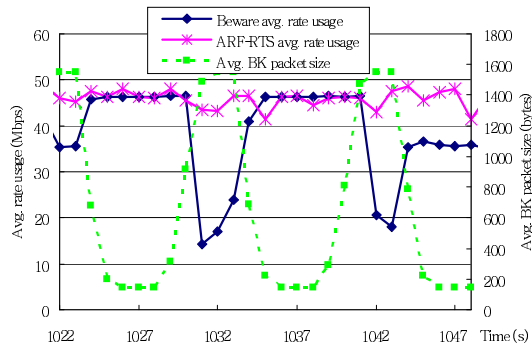


Fig. 11. Data rate selection for BEWARE and ARF-RTS with dynamically changing background traffic payload size

levels. We place 12 background traffic stations randomly scattered in the network and always use the lowest transmission rate (6Mbps) to guarantee high packet delivery rate. We synchronize the traffic patterns of the background traffic stations so that, for every 3~5 seconds, they all change packet payload size around the same time. We then add one RAA-enabled station in the network and measure the RAA's performance. We can see from Fig. 11, as the average packet size of background traffic changes, the average data rate used by ARF-RTS does not show noticeable changes. On the other hand, we can see that BEWARE tries to adapt its rate selections as background traffic packet size changes. Recall from Sec. IV, that the higher data rates the more backoff stages due to high wireless loss rates (P_{fail}). Thus, the longer backoff stages caused by increased background traffic payload sizes make the higher data rates less favorable to operate in situations with large background traffic payload size. Therefore, we can see that BEWARE adapts to the lower data rates when it senses such changes. On the other hand, when the background traffic payload sizes are small, BEWARE uses the highest possible data rate for optimal performance. We observe that BEWARE outperforms ARF-RTS by ~20% in throughput in this dynamically changing background traffic environment. We further investigate the effects of other background traffic changing patterns, such as increasing the frequency of background traffic payload size fluctuations and changing the number of simultaneously transmitting stations, and we observe that BEWARE consistently outperforms ARF-

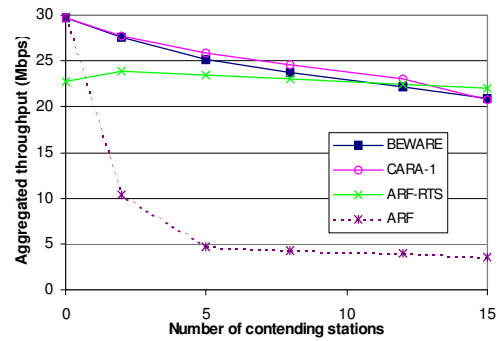


Fig. 12. Aggregate throughput comparison for BEWARE, CARA1, ARF-RTS, and ARF in close-by topology with various number of contending stations

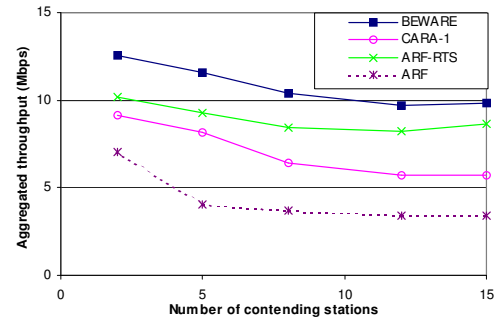


Fig. 13. Aggregate throughput comparison for BEWARE, CARA1, ARF-RTS, and ARF in random topology with various number of contending stations

RTS for 25%-50% in various dynamic changing background traffic scenarios

D. Aggregated performance in different topologies

We now evaluate aggregate performance when all stations turn on RAA and operate with the same RAA homogeneously. We first simulate a topology with minimum wireless losses, in which various numbers of stations are uniformly placed at 2.5m away from AP and each station transmits fixed size 1500-byte long UDP traffic. As shown in Fig. 12, ARF's aggregate performance degrades severely due to the "rate poisoning" effect we discussed in Sec. III. On the other hand, with the help from RTS loss differentiation, ARF-RTS performs well for any number of contending stations. Furthermore, BEWARE and CARA-1 perform closely and both outperform ARF-RTS in most cases, thanks to the overhead reduction design in CARA-1 and accurate background traffic effect estimation in BEWARE.

Secondly, we simulate a random topology with various numbers of stations randomly scattered in the network with maximum distance 45m away from AP to guarantee no hidden terminals. Each station transmits UDP traffic with random size. As shown in Fig. 13, the performance ranking differs from what we observe in Fig. 12. While ARF still suffers from rate poisoning and performs the worst, CARA-1 no longer outperforms ARF-RTS. This is because, as nodes spread at different distances from the AP, both wireless loss and contention losses are in effect, which cause CARA-1 stations to decrease the data rate aggressively. On the other hand,

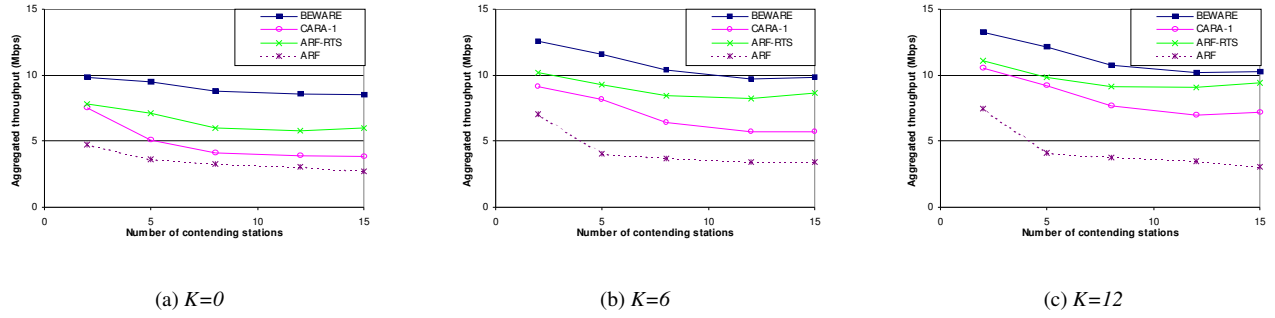


Fig. 14. Aggregate throughput comparison for BEWARE, CARA1, ARF-RTS, and ARF in random topology under different Ricean Parameter K

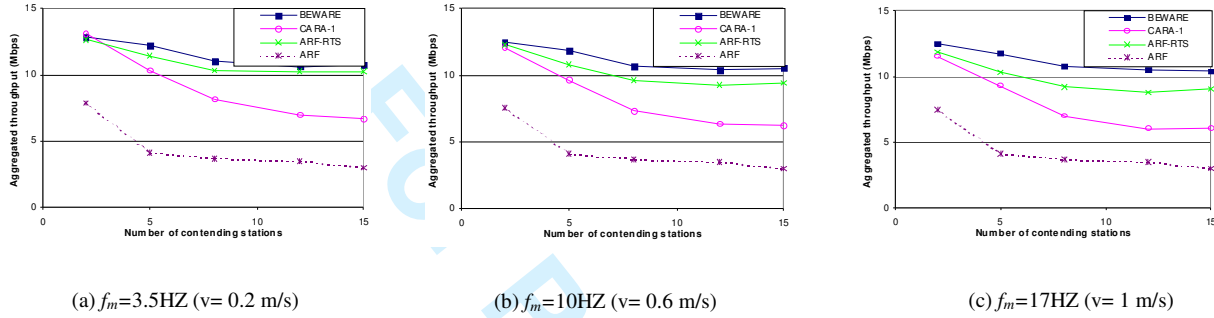


Fig. 15. Aggregate throughput comparison for BEWARE, CARA1, ARF-RTS, and ARF in random topology under different Doppler Spread f_m

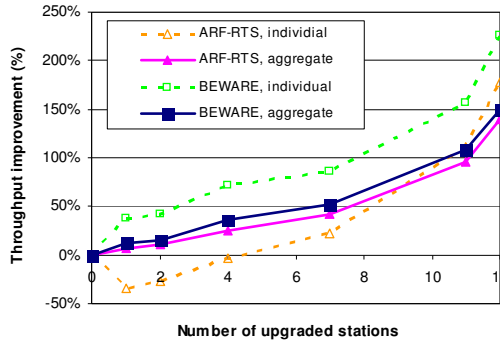


Figure 16. Individual and Aggregate throughput improvement of BEWARE and ARF-RTS with various number of contending stations in heterogeneous deployments

BEWARE still performs the best in random topology. On average, BEWARE outperforms ARF by 200%-250% and ARF-RTS, the best proposed by previous studies, by 20%-25% in aggregate performance.

E. Aggregated performance under various channel fading conditions

We now compare the performance of different RAAs under various channel fading conditions. We vary the Ricean parameter K and Doppler spread f_m . Note that, as K increases, the line-of-sight component is stronger and the overall channel SNR increases. On the other hand, as f_m increases, the channel condition changes more rapidly. Fig 14 plots the aggregate performance of different RAAs under different K in a random topology similar to what we used in Sec. V.D. We can see that,

as K increases, the overall throughput of all RAAs increases as expected. However, the ranking of RAA performance remains unchanged. BEWARE outperforms ARF-RTS, CARA-1, and ARF under all different K parameters we studied. We then plot Fig. 15 with the aggregate performance of different RAAs under different Doppler spread. We can see that, as f_m decreases, BEWARE still outperforms ARF-RTS in most cases, but the performance gap between BEWARE and ARF-RTS closes. To be more specific, BEWARE outperforms ARF-RTS by 25% when $f_m = 17\text{Hz}$. This advantage decreases to 5% when f_m decreases to 3.5Hz. Previous studies [11][20] reported that, as ARF is designed to increase its rate after several consecutive packet successes, ARF-based RAA tends to yield higher throughput by taking advantage of the slower changing channel environment. However, the performance of ARF degrades when the wireless channel condition changes rapidly. On the other hand, we can see that, as BEWARE yields comparable performance in different f_m environments. Thus, BEWARE is robust to both fast-changing and slow-changing wireless channel conditions.

F. Performance with heterogeneous RAA deployments (interoperability effects)

As rate adaptation is an option that is left open for wireless card vendors to implement, it is not uncommon that there are stations equipped with different RAAs in real world scenarios. Therefore, it is essential to evaluate the performance of different RAAs in heterogeneous scenarios. In this experiment, we evaluate how different RAAs improve the individual and aggregate performance with a gradual upgrade deployment. We consider a network with 12 stations randomly placed

1 within the transmission range of the AP, and transmit UDP
 2 traffic with random sizes. We start with the baseline scenario
 3 where all stations operate with ARF without RTS/CTS, in
 4 which all stations operate at the lowest data rate due to “rate
 5 poisoning” problem. We then gradually upgrade a number of
 6 stations with BEWARE or ARF-RTS, and evaluate the
 7 aggregate performance improvement over baseline scenario
 8 and individual performance improvement of the same station
 9 after upgrade. We can see from Fig. 16 that, as the aggregate
 10 performance of ARF-RTS improves when upgraded stations
 11 added to the network, the individual performance of ARF-RTS
 12 actually decreases when less than half of the stations in the
 13 network are upgraded. When there are just a few stations
 14 upgraded with ARF-RTS, individual performance of upgraded
 15 stations decrease due to excessive use of higher data rates as
 16 we discussed in Sec. III.B. Meanwhile, aggregate performance
 17 increases as other stations take advantage of the excess loss
 18 transmission opportunities incurred by upgraded stations. On
 19 the other hand, when there are more and more stations
 20 upgraded to ARF-RTS, those stations mutually take advantage
 21 of other upgraded stations’ loss transmission opportunities,
 22 and collectively result in higher aggregate throughput even the
 23 rate selections made by these stations are not the most suitable
 24 ones for the corresponding scenario. By contrast, both
 25 individual and aggregate performance of BEWARE start to
 26 improve when just 1 station is upgraded. In addition, as the
 27 stations upgraded to BEWARE start to use the optimal data
 28 rates for the given wireless and collision conditions, other
 29 stations benefit from the extra free transmission time spared by
 30 BEWARE stations, and yielding higher throughput for all
 31 (legacy and upgraded) stations. Note that these gradual
 32 deployment scenarios are essential to study. When introducing
 33 a new algorithm, it is vital to carefully design and analyze
 34 interoperability with existing schemes.

35 In summary, with the homogeneous and heterogeneous
 36 background traffic scenarios we evaluate in this section, we
 37 observe that, while the effectiveness of RTS-based loss
 38 differentiation RAAs differ in different scenarios, BEWARE
 39 always yields the best performance for most cases. In addition,
 40 even with only one station equipped with BEWARE in the
 41 network, both individual performance of BEWARE and
 42 aggregate network performance improve over the rate-
 43 poisoned all-ARF network.

44 VI. TEST-BED IMPLEMENTATION AND EXPERIMENTAL 45 RESULTS

46 In the previous section, we have seen BEWARE’s superior
 47 performance over other RAAs under various simulated
 48 background traffic and wireless scenarios. In this section, we
 49 implement our BEWARE algorithm in Atheros-based Linux
 50 device drivers. We then conduct a series of systematic
 51 experiments to evaluate and compare BEWARE’s
 52 performance in real-world scenarios, including indoor and
 53 outdoor environments, different number of background traffic
 54 stations and traffic patterns.

55 A. Implementation

56 We implement the BEWARE algorithm in the open source
 57 MADWIFI [21] driver based on Atheros chipsets. We also
 58 port the ARF rate adaptation algorithm for comparison
 59 purposes. Our implementation follows the system design
 60 structure we described in Ch. 4, in which we estimate the
 expected packet transmission time with the backoff procedure
 parameters (i.e. T_{busy} , P_{busy} , P_{fail} , T_{succ} , and T_{fail}). Following, we
 describe the different challenges that we face in implementing
 the system modules in real hardware and the necessary
 modifications to accommodate such challenges.

1) **Statistics collection and processing:** The first challenge
 we face is in obtaining some of the parameters needed for the
 algorithm; particularly the parameters depend on individual
 backoff stages, i.e. T_{busy} and P_{busy} . While MADWIFI may be
 the most accessible open source WLAN driver available in the
 community that implements many packet transmission details
 in the software, such as packet encapsulations, QoS settings,
 and transmission buffers, MADWIFI leaves the control and
 feedback of backoff procedure details in the firmware. In other
 words, it is not possible for us to control or even know exactly
 how many backoff counters used in a particular backoff stage.
 As a result, we are unable to keep track the number/duration of
 busy medium slots to obtain T_{busy} and P_{busy} , as we described in
 the simulations. While this information may ultimately be
 available from the 802.11 chipsets if we have the access to the
 firmware, we develop an alternative estimation-based
 approach to resolve this problem as follows.

Recall from Sec. IV, we know that T_{busy} and P_{busy} (as well as
 T_{slot}) determine the length of individual backoff stages. As a
 result, instead of collecting T_{busy} and P_{busy} and use them to
 characterize length of individual backoff stages, we keep track
 the length of individual backoff stages directly. Particularly,
 we can obtain the of 1st backoff stage by logging the length of
 all non-retransmitting successful transmissions, and subtract it
 from the actual packet transmission time (T_{succ}). We note that
 logging the length of transmissions that involve re-
 transmissions is not a good choice since they involve different
 backoff stages and potentially different packet transmission
 time (T_{fail} and T_{succ}).

It is important to note that keeping track individual backoff
 stage length takes longer time to provide the up-to-date
 channel information for the rate adaptation decisions,
 compared with our original approach by using time-slot level
 statistics (i.e. P_{busy} and T_{busy}). We will show later in the
 experiment results that this estimation-based approach does
 not seem to affect the overall performance of BEWARE
 algorithm in real-world scenarios. In addition, there is an
 interesting trade-off that collecting time-slot level statistics
 (i.e. P_{busy} and T_{busy}) might prevent the stations from going into
 sleep mode, which is critical for energy savings. In other
 words, using the new estimation-based approach might be
 advantageous from the power consumption standpoint.
 However, since power consumption issues are not the focus of
 this study, we leave the issues in exploring the tradeoff
 between energy savings and collecting most up-to-date

statistics as one of the topics in our future work.

2) **Expected packet transmission time calculation:** Since P_{busy} and T_{busy} are no longer available in our driver implementation, we need to modify the model for calculating the expected packet transmission time. Recall from the model in Ch. 4 (Eq. 1 ~ Eq. 4), we construct the derivation of overall backoff duration by the cumulative effects from the successive backoff stages. Therefore, as we get the 1st backoff stage length, $T_{1st-stage}$, in our new approach, we can calculate the overall backoff stage duration as

$$T_{avg} = \sum_{n=1}^m [(2^{(n-1)} * T_{1st-stage} + (n-1)T_{fail} + T_{succ}) * P_{fail}^{(n-1)} * (1 - P_{fail})]. \quad (5)$$

Note that, compared with the model in Ch. 4, this model also simplifies the computation complexity and makes it more suitable to be implemented in the real-world driver. On the other hand, since we have made several estimations to the environment dynamics in this new design, it is important, as we will show in the next subsection, to design a series of experiments to fully expose the new implementation approach to a rich set of real-world dynamics.

B. Test-bed Experiment Setup

Our experimental setup consists of one Cisco AP-1230 802.11a/b/g access point and laptops equipped with Proxim Orinoco Gold 802.11a/b/g combo PCMCIA cards. The laptops run Ubuntu Linux with kernel version 2.6.24.5 and modified MADWIFI driver based on version 0.9.4.

We conduct both indoor and outdoor experiments on the University of Florida campus. The indoor are conducted in an office/lab setting with concrete walls separating the rooms and many metal cubical partitions within the lab. For outdoor experiments, we choose an open area between two buildings on campus. We choose one Line-of-Sight (LOS) location and one non-LOS (NLOS) location.

We conduct each experiment with multiple runs (at least 3), and present the average over all runs. In order to provide fair comparisons among different RAAs, we choose channel 40 of 802.11a and conduct the experiments during late evenings or weekends to minimize impacts from uncontrollable external factors of interference.

We compare the performance of BEWARE with ARF and ARF-RTS, in order to understand how different rate adaptation algorithms perform in real-world scenarios with different wireless loss and background traffic environments.

C. Indoor Performance

The layout of the indoor experiments is shown in Fig. 17. We place up to 3 background traffic stations next to the AP. Each background traffic station is configured to transmit continuous UDP packets with payload size 500 bytes long, and uses the lowest data rate to ensure that the background traffic is detectable at the farthest range of the AP. We then place one RAA-enabled station in the three different indoor locations to investigate the RAAs' effectiveness under mixed wireless loss

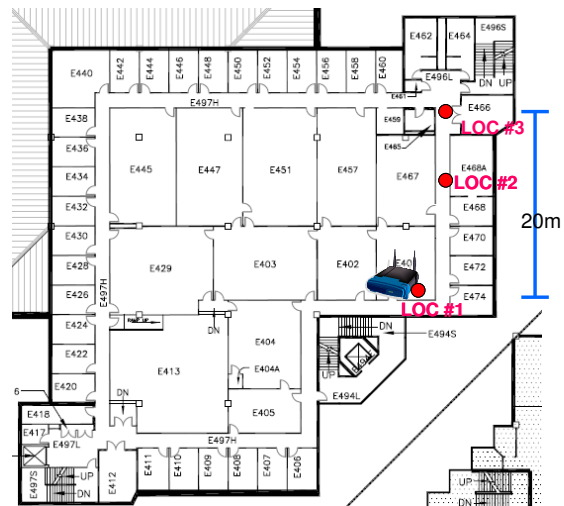


Fig. 17. Indoor experiment layout

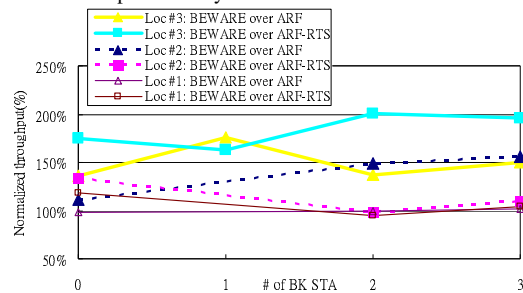


Fig. 18. Normalized throughput for BEWARE over ARF and BEWARE over ARF-RTS in indoor environment with number of background traffic stations

and contention conditions. Location #1 is within 1m of the AP so that we can examine the RAAs' performance when the wireless condition is almost perfect (SINR ~33db). Location #2 is about 12m away from the AP, with average SINR 26 to 24 db, and obstructed by 2 concrete walls in the line-of-sight from the AP location. Location #3 is further down with direct distance about 20m and is also obstructed by 2 concrete walls. The average SINR at this location is 16 to 18 db.

In Fig. 18, we plot the performance of BEWARE normalized by either ARF-RTS or ARF, at the three different locations and with different number of background traffic stations. The two thin solid lines show that, at location #1 where RAA-enabled station is just next to the AP, BEWARE does not provide significant performance improvement over ARF-RTS & ARF. On the other hand, when we move the RAA-enabled station to location #2 (dotted lines) and location #3 (thick solid lines), we can see from Fig. 18 that BEWARE consistently outperforms ARF-RTS, ARF, in all background traffic scenarios. At location #3, BEWARE's performance improvements over ARF-RTS are more significant, when compared with the performance at location. #2. In addition, BEWARE's performance improvement increases, up to 200%, with more background traffic in the network.

D. Outdoor Performance

In outdoor experiments, as shown in Figure 19, we place 2 background traffic stations next to the AP and one RAA-

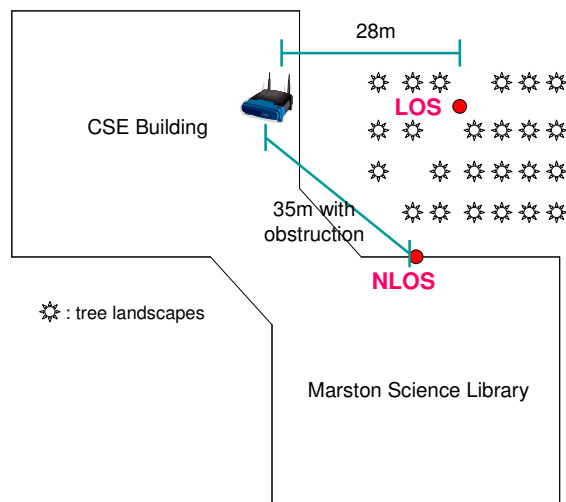


Fig. 19. Outdoor experiments layout.

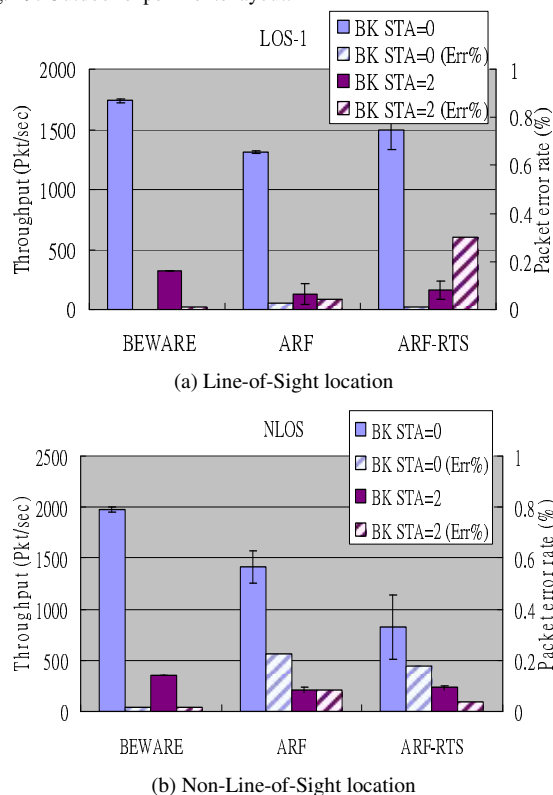


Fig. 20. Performance of BEWARE, ARF, and ARF-RTS at different locations in the outdoor environment.

enabled station in the LOS with direct distance about 28m and NLOS location at the side that is blocked by two building poles where the direct distance is ~ 35 m away from the AP. We compare the performance of BEWARE, ARF, and ARF-RTS at these two locations, with and without both background traffic stations turned on.

As we can see from Fig. 20, BEWARE consistently outperforms ARF and ARF-RTS, in both locations and in both background traffic levels, up to 250% in throughput. BEWARE's performance advantage is more significant when there are more background traffic in the network. In addition, BEWARE's packet loss rate is always $< 2\%$ in all scenarios

evaluated. On the other hand, in this outdoor experiment, both ARF and ARF-RTS suffer from substantial packet loss rate, up to 18% in no background traffic scenario and up to 35% in the two-background-traffic-station scenario.

VII. CONCLUSION

In this paper, we first identify why data rate selection strategies of 802.11-based stations should accommodate the different background traffic scenarios. This observation further helps us explain why RTS-based loss differentiation schemes, which are proposed by previous studies to aid rate adaptation algorithms in dealing with collision effects, do not perform well in background traffic scenarios. In particular, RTS-based loss differentiation hurts the performance by persistently using the same rate selections regardless of background traffic level. Therefore, these observations motivate us to design a rate adaptation algorithm that explicitly addresses wireless and contention factors in its design.

We proposed a novel background traffic-aware rate adaptation, BEWARE, that uses an accurate mathematical model to estimate the effectiveness of the data rates in given wireless and contention conditions. We show that the rate selections of BEWARE are close to what are selected by the *oracle-selection strategy* that has global knowledge of network conditions. Through extensive simulations and real-world test-bed experiments, we also show that, compared to other RTS-based loss differentiation schemes, BEWARE yields the best performance, up to 250% throughput improvement, in the scenarios we have investigated in the paper.

In the future, we plan to investigate the interactions between rate adaptation algorithms and upper-layer protocols such as TCP. We believe that, as the design of BEWARE fully addresses the wireless and contention factors in MAC layer, it should render the best performance when integrated with upper-layer protocols

REFERENCES

- [1] A. Kamerman, L. Monteban, "WaveLAN-II: A High-Performance Wireless LAN for the Unlicensed Band," Bell Labs Technical Journal, pp. 118–133, Summer '97.
- [2] M. Lacage, M. H. Manshaei, T. Turletti, "IEEE 802.11 rate adaptation: a practical approach," ACM MSWIM '04
- [3] B.E. Braswell, J.C. McEachen, "Modeling Data Rate Agility in the IEEE 802.11a WLAN Protocol," OPNETWORK '01.
- [4] S. Wong, H. Yang, S. Lu, V. Bharghavan, "Robust Rate Adaptation in 802.11 Wireless Networks," ACM MOBICOM '06.
- [5] J. Bicket. "Bit-rate Selection in Wireless Networks," MIT Master's Thesis, '05.
- [6] J. P. Pavon and S. Choi, "Link Adaptation Strategy for IEEE 802.11 WLAN via Received Signal Strength Measurement," IEEE ICC'03
- [7] Gavin Holland, Nitin Vaidya, and Paramvir Bahl, "A Rate-Adaptive MAC Protocol for Multi-Hop Wireless Networks," ACM MobiCom '01
- [8] I. Hatatcherev, K. Langendoen, R. Legendijk, H. Sips, "Hybrid rate control for IEEE 802.11," ACM MobiWac '04
- [9] J.S. Kim, S.K. Kim, S.H. Choi, D. Qiao, "CARA: collision-aware rate adaptation for IEEE 802.11 WLANs," IEEE INFOCOM '06.
- [10] 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 BRAODNETS '05.

- 1 [11] J.H. Yun and S.W. Seo, "Novel collision detection scheme and its
2 applications for IEEE 802.11 wireless LANs," Computer
3 Communications Volume 30, Issue 6, March '07.
- 4 [12] "The Network Simulator – ns-2," <http://www.isi.edu/nsnam/ns/>
- 5 [13] R. J. Punnoose, P. V. Nikitin, and D. D. Stancil, "Efficient Simulation
6 of Ricean Fading within a Packet Simulator," IEEE VTC'00-Fall.
- 7 [14] MADWIFI, <http://sourceforge.net/projects/madwifi/>
- 8 [15] M. Heusse, F. Rousseau, R. Guillier, and A. Duda, "Idle Sense: An
9 Optimal Access Method for High Throughput and Fairness in Rate
10 Diverse Wireless LANs," ACM SIGCOMM '05
- 11 [16] C. Hu and J. Hou, "A Novel Approach to Contention Control in IEEE
12 802.11e-Operated WLANs," IEEE INFOCOM '07.
- 13 [17] S.C. Wang and A. Helmy, "Performance Limits and Analysis of
14 Contention-based IEEE 802.11 MAC," IEEE LCN '06.
- 15 [18] R. Draves, J. Padhye, and B. Zill. "Routing in Multi-Radio, Multi-Hop,
16 Wireless Mesh Networks," ACM MobiCom '04.
- 17 [19] T. Salonidis, M. Garetto, A. Saha, and E. Knightly, "Identifying High
18 Throughput Paths in 802.11 Mesh Networks: a Model-based
19 Approach," IEEE ICNP '06.
- 20 [20] D. Qiao and S. Choi, "Fast-Responsive Link Adaptation for IEEE
21 802.11 WLANs," IEEE ICC'05.
- 22 [21] MADWIFI, <http://sourceforge.net/projects/madwifi/>
- 23
- 24
- 25
- 26
- 27
- 28
- 29
- 30
- 31
- 32
- 33
- 34
- 35
- 36
- 37
- 38
- 39
- 40
- 41
- 42
- 43
- 44
- 45
- 46
- 47
- 48
- 49
- 50
- 51
- 52
- 53
- 54
- 55
- 56
- 57
- 58
- 59
- 60

1
2
3 The link of the conference version of this paper:
4

5 http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4594830
6

7 The differences between the conference version and journal version are that:
8

9 1) journal version adds the test-bed experimentation part of the proposed
10 algorithm (Sec. VI), including the implementation in Linux driver and
11 extensive experimental results in analyzing the implemented algorithm and
12 other competitive algorithms.
13

14 2) The journal version extends the contents and integrates the feedbacks
15 from the conference paper.
16

17 Thanks,
18 Shao-Cheng Wang
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

For Peer Review