

Applying Active Measurement in the Rate Adaptation for 802.11 Wireless Networks

Shaoh Lv
National Laboratory of Parallel
and Distributed Processing
National University of Defense
Technology,
ChangSha, Hunan, 410073,
China
shaohelv@nudt.edu.cn

Xiaodong Wang
National Laboratory of Parallel
and Distributed Processing
National University of Defense
Technology,
ChangSha, Hunan, 410073,
China
xdwang@nudt.edu.cn

Xingming Zhou
National Laboratory of Parallel
and Distributed Processing
National University of Defense
Technology,
ChangSha, Hunan, 410073,
China
xmzhou@nudt.edu.cn

ABSTRACT

Rate Adaptation (RA) is a mechanism to choose transmission rate based on the dynamic channel quality in wireless networks. The adaptation algorithm run solely at the sender side in 802.11 networks is studied. The key insight is the *inference discrepancy* in inferring the relative order of the expected performance of candidate rate, which argues that one can not always reach the correct order based on the channel state information collected exclusively by the sender itself. The consequence is wrong rate decision and significant performance loss. Therefore, a new RA structure is proposed to mitigate such effect by a novel component, *rate testing*. By embracing the active measurement, a lightweight and effective testing mechanism SFB, *short frame burst*, is proposed to detect and filter out the unsuitable transmission rate. Finally, an active measurement-based rate adaptation mechanism (AMRA) is designed and implemented. The experiments show that AMRA outperforms many other well-known RA solutions in most scenarios.

Categories and Subject Descriptors

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

General Terms

Design, Experimentation, Performance

Keywords

Rate adaptation, MAC, 802.11

1. INTRODUCTION

To utilize the resources efficiently in time-varying wireless channel, one should choose the proper transmission parameters based on the dynamic condition. The WLAN standard, IEEE 802.11 [1], also

provides the multi-rate capability to balance the tradeoff between communication efficiency and reliably. For example, high transmission rate is adopted in channels of good quality while lower rate is preferred to yield more resilience in those of poor quality. Rate adaptation (RA), e.g., to choose transmission rate based on the dynamic channel condition, is thus critical to the system performance. The standard however did not standardize the RA algorithm, which recently has been a hot topic.

The adaptation mechanisms can be categorized as sender-side and receiver-aware. This paper focuses on the sender-side adaptation, where all used information is collected by the sender without any additional feedback or help provided by the receiver except ACK. In opposite, the receiver-aware RAs make use of the channel state information only available at receiver, thus requires support and modification at the both sides, leading to the difficulty in deployment.

In general, there are two tasks to perform RA: first, information collection—collect sufficient information about the (expected) performance of all candidate rates. Second, rate selection—determine the relative performance order of all candidate rates, i.e., *rate ranking*, and choose the best one, i.e., *rate preference*.

One can rank the candidate rates directly or indirectly. The first way means the sender employs every rate in turn to get direct knowledge about the performance, and then order the rates. The second way means the sender makes use of one transmission rate constantly during one evaluation period, and then based on the observed transmission status, assess the current rate and *infers* the expected performance of other candidate rate and finally orders all rates. The latter is the dominant way in state-of-the-art RA designs.

Unfortunately, as shown in section 3, there is severe inference discrepancy in inferring the relative order for sender-side RA. Such discrepancy argues that, in fact, one can not always reach the correct order according to the information collected exclusively by the sender itself. The consequence is wrong rate decision and significant performance loss.

It is the partial understanding about the channel condition that leads to the discrepancy. For a transmission, the default feedback from receiver to sender in 802.11 standards is just a binary decision, e.g. success or failure (indicated by the presence or absence of ACK frame). Such coarse information is insufficient to evaluate the performance of the rate being used, i.e., no way to determine the cause of a packet loss to collision versus weak signal [19]. It is thus not surprising to get an inaccurate estimate about the performance of the transmission rates not being used, let alone the relative order of all candidate rates.

How to mitigate the effect of inference discrepancy? The very

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

WICON '08, November 17-19, 2008, Maui, Hawaii, USA
Copyright 2008 ICST 978-963-9799-36-3 ...\$5.00.

direct way, i.e., to modify the standards to enable more feedbacks, seems less feasible. From the other aspect, a novel but nature way is adopted in this paper, which tries to detect and avoid the discrepancy, than to remove it. More particularly, in the advanced RA, a new process, *rate testing*, is run immediately after a new transmission rate is preferred. Such process is charged for evaluating quickly the transmission rate just chosen and further filtering it out if unsuitable.

Up to now, there is no RA design to take rate testing into account explicitly. For example, as in RRAA [6] or sampleRate [15], after rate preference, the preferred rate is directly applied in data transmission until the next decision point. ARF, the first public RA algorithm, touches the idea implicitly but the single transmission based decision is quite problematic.

It is certainly challenging to design an effective testing mechanism. On one hand, accurate assessment should be provided to identify and retain only the optimal rate. On the other hand, the permitted measure period is strictly limited. The data transmission must be delayed until the rate test is accomplished. The longer we stay in testing, the less time we have for data transmission.

To enable effective rate testing, active measurement is employed, which is flexible and powerful in obtaining the network characteristics with the help of customized probes. In general, as compared to passive fashion, the active estimation is much faster and more accurate in getting the characteristics of specific network path or link though some measurement cost is required.

The rest of the paper is organized as follow. The inference discrepancy is at first explored in depth in section 3 and then in section 4 a new RA structure is discussed to mitigate the negative impact. The corresponding mechanism, AMRA (active measurement-based rate adaptation), is also given. Moreover, in section 5, the rate testing method, SFB (*short frame burst*) is presented in detail. Finally, section 6 describes the experiment setup and results and section 7 concludes this paper.

2. RELATED WORK

The state-of-the-art rate adaptation algorithms are categorized as sender-side and receiver-aware. RAF [2] and RBAR [12] are the representative receiver-aware RA mechanisms. In addition, OAR [13] and MAD [14], though originally built upon RBAR, can be adopted widely over many RA algorithms.

Starting from the first proposal ARF [10], the sender-side RA solutions are studied extensively. SampleRate, the unique one to derive directly the performance order, periodically switches among all candidate rates and prefers the best-performing one. AARF [11] strives to mitigate the oscillation of ARF under stable channel. RRAA [6] makes use of short-term loss ratio to trigger the rate adjustment. In addition, both CARA [5] and RRAA adopt RTS/CTS (in different fashion) to isolate the loss due to collision. Finally, in [16], RA is performed by using jointly the instantaneous signal strength and the long-term statistics.

No special consideration is given to the rate testing in previous work. ARF is an exception to touch the idea implicitly. ARF regards as probe the very next transmission after up-scaling the transmission rate and rolls back if the probe is failed. Such single transmission-based policy is however misled easily due to the random behavior of wireless channels. To our knowledge, this paper is the first to recognize the importance of rate testing and embrace active measurement in 802.11 rate adaptation.

3. THE INFERENCE DISCREPANCY

We now explore in depth the inference discrepancy. After the

two major ways are reviewed in brief to get the performance order, the inference discrepancy is described with further illustration via experimental observations. Finally the implication to the design of rate adaptation is discussed.

3.1 Ordering the candidate rates

To do rate preference, one must get the relative performance of all candidate rates. Two issues should be considered. The first is to choose a proper metric to indicate the channel state. In general, a channel state indicator can be either rate dependent metric (RDM) or rate independent metric (RIM). The former, like packet loss ratio or throughput, depends on the transmission rate being used. The latter, on the other hand, like RSSI or SNR, has no correlation with the transmission rate. The practical channel indicator usually comprise several simple metrics. For simplicity, the indicator consisting of only RIMs is referred to as channel quality indicator while the one containing RDMs as link performance indicator.

The second issue is to order all candidate rates with respect to the expected performance. Two possible methods can be applied:

- *Direct method*: employs each candidate rate in turn directly in data transmission, and then estimate the specified link performance indicator, and finally order all rates in terms of the estimated performance. As an example, sampleRate chooses the average transmission time (e.g., throughput) as the indicator to assess and prefer the transmission rate.

One of the vital drawbacks of the direct method is: it consumes long time to accomplish the entire estimate when there are many candidate rates. More importantly, such estimate impairs the system performance. Among the candidates, only one is the optimal so that the performance must be sub-optimal when the non-optimal rate is used. In particular, lots of losses can be expected when the transmission rate is too high to be supported by the channel. Such performance loss is demonstrated in the evaluation of sampleRate (see section 6).

- *Indirect method*: In a measure period, just one transmission rate is in use, during which the specified metrics are measured. The performance of the current transmission rate can be measured directly. The problem is how to assess other candidate rates that are not used recently. The unique feasible way is to *infer* the expected performance implicitly. Then the candidate rates can be ranked according to the information from measurement and inference.

This indirect method is widely applied but little attention is paid to its effectiveness in the two steps, measurement and inference. Several RA solutions, e.g., [5][11], are built upon the original ARF algorithm without any fundamental improvement on the two steps. Recently, a mentionable enhancement on the measurement is realized in RRAA. Instead of the number of successive success (or failure) in ARF, short-term loss ratio is used as link performance indicator. To our knowledge, however, no consideration is given to the inference step. So far almost all RA solutions are based on some heuristic assumptions. For example, when the current rate performs very well, it is believed commonly that the immediate higher one also does. As a foundation of inference, these assumptions should be examined carefully in practice.

3.2 Description of the Inference Discrepancy

The inference discrepancy can be described through the following three claims. Though these claims are mentioned partially in some previous work, their interaction and the induced effect on the system performance are less understood.

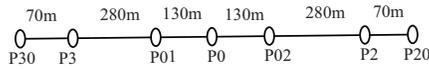
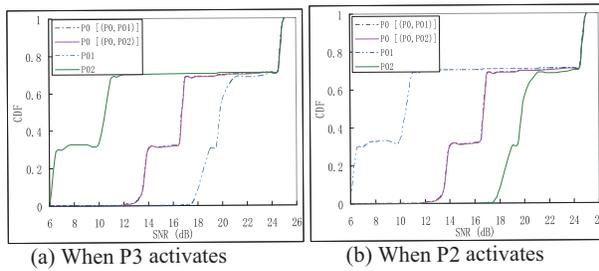


Figure 1: Linear topology: seven node



Inference discrepancy sheds light on the well-known performance issue of the threshold-based RA. As an example, let us focus on RRAA [6], a recent proposal that adjusts the transmission rate based on the short-term loss ratio. It upscales the transmission rate when the loss ratio is very low and down-scales the rate upon high loss ratio. RRAA suffers the performance oscillation even in stable channel. In particular, consider when the node pair is separated at 160m where the throughput at 36Mbps is already very low so that 36Mbps is no longer an optimal choice. RRAA, however, still tries to use such rate periodically. Something like:

- Work at 24Mbps: OK, the failure is very rare. So:
- Try 36Mbps: Oh, large number of losses! So:
- Work at 24Mbps again: OK, still very few failures. So:
- Try 36Mbps again: Oh, too many losses!

Along with the rate switch, the performance oscillates, e.g., the optimal point is reached with 24Mbps while so poor is when 36Mbps is used. This is out of question very harmful since no stable performance is enjoyed due to the pathologic behavior of RA even with no channel variation.

Now the problem is how to mitigate the effect of interference discrepancy? Intuitively, the origin of such discrepancy is the lack of knowledge about the receiver-side condition such that the sender cannot capture the time-varying channel characteristics timely. To enable more feedback, however, though may be effective, is infeasible in practice since which requires nontrivial modification of the 802.11 standards. The solution presented here follows a novel aspect. A new component, *rate testing*, is introduced which strives to discern the sub-optimal rate that is preferred due to the inference discrepancy. Next, a general structure that takes rate testing as a critical component is at first described and an effective testing mechanism by use of active measurement is then discussed.

4. THE GENERAL STRUCTURE AND AMRA

In this section, the importance to handle the interference is at first analyzed and then the new RA structure is discussed along with the concrete AMRA algorithm. The details about the parameter setting and some practical issues are also given.

4.1 Handling the interference

Interference plays an increasingly important role in modern wireless networks. For simplicity, sender/receiver/interferer is termed as S/R/I. Generally speaking, there are two types of interference.

- *Collision*: collision is the overlap of S's and I's transmission at R when R is within the communication range of I. The effect of collision on RA has been already considered, e.g., in CARA and RRAA. AMRA has already incorporated with the adaptive RTS proposed in [6] to protect the RA operation from the collision-induced losses.
- *OCR interference*: general speaking, the interference range of a node is larger than its communication range [18] so that even when R is out of the communication range of I, the reception of S's packet is still possibly distorted by I. OCR interference occurs when I is out of the communication range of both S and R. At this time, RTS/CTS is helpless. To our knowledge, there has no attention on the effect of OCR interference.

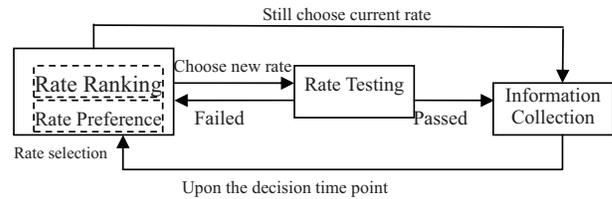


Figure 5: Component view of the new RA structure

Rate (Mbps)	RIT	RDT	M
6	50.00	N/A	22
9	14.34	45.32	30
18	13.25	45.22	40
36	11.50	38.63	50
48	9.7	31.00	50
54	N/A	24.40	50

Figure 6: The parameters of AMRA

OCR interference is different from collision though both of them impair the system performance. OCR interference is not so strong that can always lead to reception failure. It is suitable to lower the transmission rate to combat the OCR interference in some cases. On the other hand, OCR interference is also not the same as poor channel quality in that the failure occurs not randomly but only when the interferer is active. In summary, one should take into account both the strength and frequency in understanding the effect of OCR interference. The proposed algorithm, AMRA, fulfills this task with the help of rate testing. We finally note that AMRA can integrate the mechanism as proposed in [6] easily to handle collision, which however is left for future.

4.2 The general structure and AMRA

Fig.5 gives a component view of the general structure. In particular, there are three components—information collection, rate selection and rate testing. At first, in information collection phase, sender transmits data packets and tracks the transmission status. When the decision time arrives, RA then switches to the rate selection phase to rank the candidate rates and choose the transmission rate with best expected performance. If nothing is changed, RA enters the next information collection. Otherwise, RA launches rate testing to evaluate whether the new rate is really optimal. If not, the adjustment decision is aborted and RA goes back to choose another rate. Otherwise, the new rate will be used in data transmission.

Compared to the previous arts, the novelty of such structure is the addition of rate testing, which serves to mitigate the effect of inference discrepancy. It is an independent phase to interconnect partially the rate selection and information collection. A new chosen rate can be used further only when it passes the test.

Now the functions of the three processes in AMRA is given.

- *Information collection*: AMRA computes the short-term loss ratio during the latest M transmissions (include retries). The loss ratio is then $(M-S)/M$ where S is the number of successful transmissions. After M records are collected, the next process (i.e., rate selection) is started.

- Rate selection: AMRA decides to raise the transmission rate when the loss ratio is below than given threshold, *rate increase threshold*(RIT); and similarly, lower the transmission rate when such ratio exceeds another threshold, *rate decrease threshold* (RDT). Otherwise, AMRA does not change the transmission rate. No matter the rate is changed or not, all collected information is discarded after this step and the collection process starts again.
- Rate testing: AMRA enables rate testing when the next higher rate is chosen. If the SFB process prefers to reject the new rate, AMRA simply rolls back to the old one.

Before the run of AMRA, the parameters such as the estimate window size and the thresholds must be determined. AMRA evaluate the current rate by leveraging the latest transmissions within the window M . How to determine M ? If M is too small, the evaluation may be overwhelmed by temporary random behaviors, especially, leading to the misunderstanding of the transient behavior for the nearly stable channel. To reach a dependable estimate, therefore, AMRA employs sufficient large window to minimize the effect of the random behaviors. The process to determine these parameters is similar to RRAA. We omit the detail here and refer the interesting reader to [6].

As an illustrative example, Fig.6 gives the setting for 802.11g and 1500bytes, one representative packet size in Internet. From Fig.4, one can see that, during our experiments, 24Mbps and 12Mbps since they always perform not as well as their immediate lower neighbor, 18Mbps and 9Mbps respectively so that AMRA never prefers them. In practice, different setting may be required for different packet size but the method is the same.

4.3 Miscellaneous Issues

Two practical issues should be addressed: (1) too less records in information collection and (2) handling rate downscaling after rate selection.

By use of an estimate window M , AMRA already filters out many outdated transmissions. But if too long time is spent in waiting M transmissions, the loss ratio still becomes obsolete. Therefore, AMRA includes a timeout mechanism in information collection. The estimation window is flushed and a new estimation is started if less than M frames are observed after certain period of time (currently one second).

Not as the general structure, AMRA does not perform rate testing upon downscaling the rate. The reason is two-fold. First, given the sufficient large window, AMRA provides dependable estimate about the performance of current rate. If the loss ratio exceeds RDT (thus triggering rate reduction), we believe, the ability to yield good performance is essentially lost so that it is unnecessary to retain the current rate. Second, if the immediate lower rate is still beyond the support of channel, the rate is reduced again at the next decision point until be suitable for the channel quality. Though this may introduce some delay to respond to the drastic decrease of channel quality, the overall loss is limited since the frequency of such drastic decrease is low in our targeting scenario. In opposite, high overhead is introduced if one performs rate testing every time when lowering the rate. We found that this choice is effective in at least our experiments.

5. THE SFB MECHANISM

Active policy is a powerful and flexible way to estimate the network characteristics. The major issue in applying the active policy is to reach a good tradeoff between accuracy and cost. In particular,

```

Just after a new and higher rate  $R$  is chosen:
Modify the fragmentation threshold;
Generate the large probe packet;
While (not all probes are sent) do {
    Transmit a probe at rate  $R$ ;
    If (tx succeeds) then fosc++;
    If ((fosc >= a_t) or (tx is failed)) then break;
}
Reset the fragmentation threshold;
Discard all remaining probe frames;
If (fosc >= a_t) then accept the rate  $R$ 
else reject the rate  $R$ ;

```

Figure 7: The pseudo-code of SFB

from the view of application, all active probes contend for the channel access and reduce the resource utilization. Our focus is thus to design an effective probing mechanism and customize the probe to evaluate the interesting transmission rate as quickly and accurately as possible at the expense of reasonable cost.

5.1 The basic process

The name of SFB comes from the particular probe, *short frame burst*, e.g., a sequence of MAC frames of small size that been issued back-to-back.

When the loss ratio of the burst is high, SFB rejects the transmission rate. In general, the loss ratio, under the same channel, grows up when the packet size increases. Thus, when the loss probability of short frame is not low, it is reasonable to believe that the performance of large packet can not be very well.

In practice, SFB adopts two important methods to reduce the overhead. First, SFB stops the probe process upon any failure. That is, in a run, SFB does not contend for the channel again after missing an ACK. Second, SFB counts the number of successful transmission, FOSC, before the first failure. If such number reaches given threshold, denoted as a_t (see Fig.8), SFB stops the testing and accepts the rate being used. How to choose parameter to guarantee the effectiveness of the two optimizations is discussed in next.

Implementation: We make use of the fragmentation mechanism to implement the SFB. In short, a sufficient large MAC packet is generated and then the fragmentation threshold is modified to the size of probe frame, enabling the consecutive transmission of every fragment to emulate nearly the short frame burst. However, it may need more than one packets when a number of probes is required. Though the above implementation is just an approximation, the accuracy is sufficiently high. More importantly, it is fully compatible with the standards and can be easily deployed. Fig.7 shows the pseudo-code of SFB.

5.2 The Parameters of SFB

There are three critical parameters in SFB, e.g. the probe length, f_size , the number of probes, f_count , and the acceptance threshold, a_t . The used notations are enumerated in Fig.8. The analysis is conducted via two steps: first, we target on the basic function of SFB, e.g. always issue all probes and reject the rate when at least specific number of failures occur. Then f_size and f_count is determined. Second, we go into the optimized SFB and pinpoint that

Notation	Meaning
BER(e, R)	the bit error ratio of rate R under channel condition e
P(e, R, L)	the packet loss ratio of size L at rate R under condition e.
e(R)	the lower bound of condition to use rate R for 1500byte packet
f_size(R)	the length of probe in SFB to test the rate R
f_count(R)	the total number of probes in a run of SFB for rate R
f_error(R)	the min number of failures in a run of SFB to reject rate R
a_t(R)	the acceptance threshold FOSC should reach to accept rate R

Figure 8: The used notations

through proper setting, the basic process can be approximated well with less overhead. The threshold a_t is then determined.

We assume: (1) both the receptions of different probes and of different bits in the same probe are independent. (2) The effects of the errors in PHY preamble, PHY header and ACK are ignored. The conservative transmission policy makes them much more robust than the data payload. In fact, one can easily extend our analysis to include them.

At first, the packet loss ratio is derived as (1) in terms of bit error ratio, where 28 attributes to the length of MAC header.

$$P(e, R, L) = 1 - [1 - BER(e, R)]^{8(L+28)} \quad (1)$$

SFB with probe of size L rejects rate R if the loss ratio is no less than $P(e(R), R, L)$. In other words, SFB should reject the rate if the number of transmissions is still less than f_count when f_error failures occur. The relation is denoted in (2) where the function $Upp_Int(x)$ outputs the immediate integer no less than x .

$$f_error(R) = Upp_Int(f_count(R) * P(e(R), R, L)) \quad (2)$$

General speaking, larger number of probes results in more robust decision but at the expense of higher cost. To meet a reasonable tradeoff, f_error is taken as 2. Then f_count , based on (2), is derived by (3).

$$f_count(R) = Upp_Int(2/P(e(R), R, L)) \quad (3)$$

5.2.1 Choosing the probe size

Given the number of tolerable failure, e.g., f_error , the probe size, f_size , is chosen to minimize the overhead. In particular, at the turning condition $e(R)$ of rate R , the average cost includes: (1) contention time before the first transmission; (2) time for all successful frames; (3) time for all failed ones. The first part is however ignored since which is independent with SFB.

The total overhead, T_O , is expressed as follow.

$$T_O(R, L) = \sum_{i=1}^{f_s} (T_P + T_M + L/R + SIFS + T_A + T_w(i)) + \sum_{j=1}^{f_error} (T_P + T_M + L/R + SIFS + T_{AT} + T'_w(j)) \quad (4)$$

where $f_error=2$, $f_s=f_count-2$, $T_w(i)=SIFS$ and $T'_w(j)=EIFS+T_d(j)$. Therefore, The follow two declarations are considered. If both declarations are valid with high probability, it is certainly effective to stop upon the first error or at least a_t successive successes instead of issuing all probes.

$$T_O(R, L) = 2 * f(R, L) + f_part - 2C_g \quad (5)$$

where $ber=BER(e, R)$ and

$$f(R, L) = (L/R + C_g) / [1 - (1 - ber)^{8(L+28)}] \quad (6)$$

$$C_g = 2 * SIFS + T_P + T_M + T_A \quad (7)$$

$$f_part = \sum_{j=2}^2 (T'_w(j) + 2(T_P + T_M + T_{AT})) \quad (8)$$

Both C_g and f_part can be approximately regarded as invariant. To verify this, just note that T_P for transmitting PHY preamble and header, T_M for transmitting MAC header and SIFS are specified by the standard. Though both T_A , time for transmitting ACK and T_{AT} , time for handling ACK timeout depend on the transmission rate of preceding data packet, the difference is very little due to the small size of ACK. T_P , T_M , T_A and T_{AT} are all constants so that C_g and f_part are also invariant.

Therefore, to minimize the total overhead, is equal to minimize $f(R, L)$. The derivative, $f'_L(R, L)$ is:

$$f'_L(R, L) = \frac{1 - g(R, L)}{R * [1 - (1 - ber)^{8(L+28)}]^2} \quad (9)$$

where $g(R, L) = (1 - ber)^{8(L+28)} * (1 - 8Ln(1 - ber) * (L + R * C_g))$.

One can easily verify that the right part of the function $g(R, L)$ is positive for $ber \leq 10^{-3}$, $L \leq 2000$ and all transmission rates in 802.11g. To enable error-free reception, the ber in wireless network is usually much lower than 10^{-3} . So $g(R, L) < 1$ and further $f'_L(R, L) > 0$. In consequence, the average overhead grows up monotonically with the increase of probe size. This is why SFB prefers the short frame.

Then what probe size should be chosen? The use of too short frame may increase the difficulty in implementation. Especially, the maximum number of fragmentation is generally limited (10 in our testbed). Given this constraint, (10) shows how to obtain the number of probe packets, T_N , where f_m is the multiple of f_size that is the closest to MTU.

$$T_N(R) = \begin{cases} \lceil f_count(R)/10 \rceil & \text{if } \lfloor MTU/f_size \rfloor \geq 10 \\ \lceil f_count(R) * f_size(R)/f_m \rceil & \text{otherwise} \end{cases} \quad (10)$$

Fig.9 plots the required number of probes for different transmission rate under multiple probe size. Two observations are drawn. First, when the probe size is too small, the amount of probe packets is relatively large. For example, the required number of packets for 54Mbps is 10 when the probe size is 100byte, which will introduce high measurement cost. Furthermore, Such large number of transmitted packets also violates the implementation approximation since the transmission interval between two packets is at least as large as DIFS, not back-to-back. Second, the number of packets becomes invariant for all rates after the probe size is up to 200byte. The probe size thus should be close to 200byte. In SFB, the probe size is 235byte.

5.2.2 Choosing the acceptance threshold

Now consider the two optimizations given the probe size 235byte. Remind that basically, SFB rejects the rate if two failures occur. Therefore, The follow two declarations are considered. If both declarations are valid with high probability, it is certainly effective to stop upon the first error or at least a_t successive successes instead of issuing all probes.

- A1: given the first failure at the n th transmission ($n < a_t$), there will be at least one failure in the remaining.

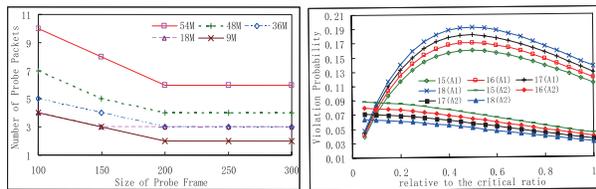


Figure 9: The required number of probe packets at diff. rate of A1 and A2 for some a_t with diff. probe size (15~18) at 48Mbps.

Rate	54	48	36	18	9
a_t	23	17	15	13	12
f_{count}	50	35	25	20	17
f_{size}	235byte				

Figure 11: The SFB parameter setting

- A2: given the successes at all the first a_t probes, there will be no more than one failures in the remaining.

The impact of violation of the two declarations is quite different. The violation of A1 means that: one failure occurs at any given n th transmission ($n < a_t$) but all other ($a_t - 1$) transmissions are successful. In fact, A1 violation is false positive, that is, we have a good transmission rate but SFB rejects it. Similarly, when A2 violates: two failures occur during the latter ($f_{count} - a_t$) transmissions while all the others are successful. Therefore, A2 violation is false negative, i.e., accepts the too high transmission rate. In particular, the violation of A2 is much more harmful: when A1 violates, we just do things less efficiently while when A2 violates, the performance must be very low since many transmissions are failed by using the too high rate. Following shows the violation probability of A1 and A2.

$$P_e^1 = [a_t * P(e, R, L)] * [1 - P(e, R, L)^{f_{count}-1}] \quad (11)$$

$$P_e^2 = [C_{f_{count}-a_t}^2 * P(e, R, L)^2] * (1 - P(e, R, L))^{f_{count}-2} \quad (12)$$

The design goal here is, thus, to lower the false negative probability (A2) as much as possible while keeping the false positive probability (A1) low. Fig.10 plots the violation probability at 48Mbps. Here the critical ratio (cr) refers to the loss ratio at turning condition, e.g. $P(e(R), R, 235)$. For A1, the violation probability is shown when the loss ratio is less than the critical ratio, e.g., $p < cr$. p is shown as the x-axis in the figure. For A2, the violation probability is shown when the loss ratio is larger than the critical ratio, e.g., $(1+p) > cr$. As one can see, the less a_t is, the lower the violation probability of A1 is. The case is however opposite for A2. The results of other a_t are omitted since the trend is similar. And furthermore, the results at other rates are also qualitatively similar. Therefore, relatively large a_t is preferred to reduce false negative while maintain the false positive at reasonable low level.

Summary: The final settings of SFB are summarized in Fig.11. Note that no more than a_t probes are issued in one run no matter how large the f_{count} is. Given the length of MTU as 2304byte, one thus needs at most three packets one time.

6. EXPERIMENTAL RESULTS

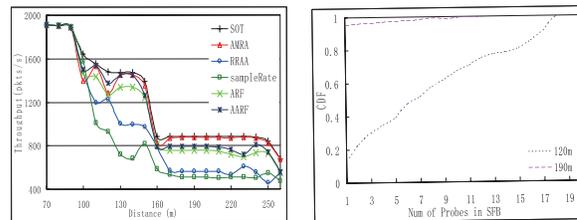


Figure 12: The SOT and the throughput of different RA solutions at varying distance **Figure 13: The cumulative throughput of different RA solutions at varying distance**

We have evaluated AMRA and four well-known RA solutions, ARF, AARF, RRAA and sampleRate. AMRA outperforms all the four solutions in most scenarios. These results validate the importance and effectiveness of both introducing rate testing as an independent phase and embracing the active measurement in rate testing.

In experiments, all hosts with an 802.11g card run Linux with the latest madwifi driver. The outdoor communication range is less than 280m. RTS/CTS is disabled by default. All experiments run ten times and the average results are reported. Each run lasts for about 100 seconds. The experiments are performed at mid-night to avoid the external interference.

The metric to evaluate performance is throughput, e.g., the number of packets per second. Without special statement, the packet size is 1500bytes. As for comparison, the single optimal throughput (SOT) of data flow is also provided in some scenarios: the experiments are repeated at different fixed transmission rate for data flow, and then the highest throughput is regarded as SOT. Note that SOT may be not the upper bound of the performance, but just the best that using single rate can reach.

6.1 Scenarios without interference

Two-node experiments is at first performed. Fig.12 shows the SOT and the throughput when RA is enabled. One can find that: AMRA outperforms the others significantly in most cases, especially when the rates higher than the optimal one all perform poorly. For example, the performance of AMRA is quite close to the SOT and much better than the others when the distance is in between 170m and 240m, where the optimal rate is 18Mbps and the throughput at the higher rates (36Mbps, 48Mbps and 54Mbps) are all very low. On the other hand, AMRA performs not as well as AARF when at least one high rate though not optimal has nontrivial success ratio. The example is when the distance is 120m, the optimal rate is 36Mbps but the transmissions at 48Mbps still have about 50% success probability.

Further, when AMRA is adopted, the most percentage of transmissions are performed at the optimal rate including at 120m. The reason for the poor performance of AMRA at 120m attributes to far less total number of transmission. But why? Fig.13 shows the cumulative distribution of the number of probes used in SFB. To reject 36Mbps at 190m, in about 95% cases, less than 5 frames are needed. The cost is very low. At 120m, however, one needs more than 10 transmissions to reject 48Mbps in the 40% cases. The overhead of SFB is so high that AMRA has less time in data transmission. At this time, SFB can still identify and reject the sub-optimal higher rate, but the cost is too high. The tradeoff between the accuracy and the overhead is still an interesting issue for future.

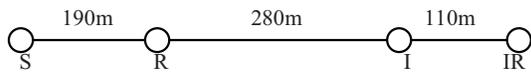


Figure 14: Linear Topology: four nodes

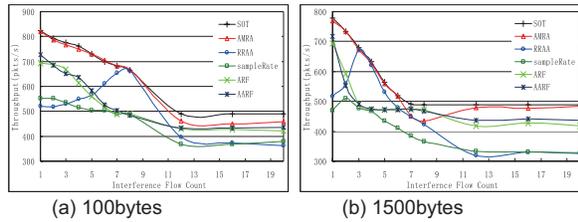


Figure 15: The SOT and the throughput of different RA solution, each interference flow is 100pps.

6.2 Fixed interference strength

These experiments are performed in the topology shown in Fig.14. The flow of I→IR is interference traffic with volume varies from 100pps to 2000pps. Fig.15 presents the saturated throughput of flow S→R, where (a) and (b) refers to the case that the size of packet sent by I is 100bytes and 1500bytes respectively.

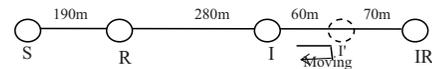
One can see that: first, the consequence of OCR interference is severe even when the interference load is light. In particular, at 190m, the SOT is about 880 without interference. But with interference, even only 100pps of 1500bytes packet, the SOT is down to 780, over 10% loss! Second, the throughput of AMRA is significantly higher than all other algorithms in most cases. In particular, in both Fig.15(a) and Fig.15(b), when AMRA is in use, the throughput is the closet to the SOT in most times. Moreover, when the interference is 1200pps of size 100 bytes, AMRA works at the optimal rate 9Mbps for much longer time than all the others, confirming that SFB can certainly capture the OCR interference and further identify the optimal rate. On the contrary, RRAA wastes too many chances in transmitting at the too higher rate, e.g., 18Mbps. The cyclic switch between 12Mbps and 18Mbps results in the poor performance of RRAA.

Both ARF and AARF take as probe the very next transmission after up-scaling and continue to use the new rate if the probe is successful. Such decision is problematic. For example, when the interference level is 1200pps of size 100bytes, the success ratio is about 32% for transmission at 18Mbps. The probability that ARF and AARF decide to use such sub-optimal rate after probe can therefore be still as large as one third! In experiments, AARF uses 18 Mbps in nearly 16% transmissions, which however contribute only about 7% successful transmissions in total. The results expose the shortcoming of using too few probes in passive fashion to deal with the OCR interference.

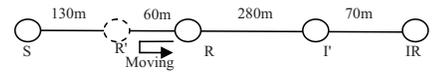
6.3 Fixed interference density

Now let's focus on the effect of varying interference strength by fixing the interference flow as 1200pps with size 100 bytes. Fig.16 describes the details of the three different cases and Fig.17 shows the throughput results for each case.

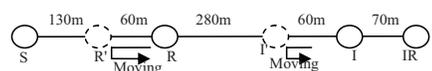
One can see that AMRA performs the best in almost all cases and the difference between the throughput of AMRA and the SOT is also very small. SampleRate takes use of the optimal rate the least times. The reason is that SampleRate must probe periodically at the rate that has lossless throughput higher than the current throughput.



(a) I moves, S and R hold

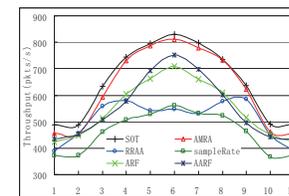


(b) R moves, S and I hold

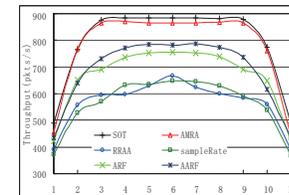


(c) I and R move, S holds

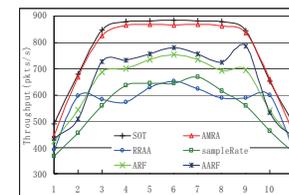
Figure 16: The topologies with node mobility



(a) Variable interference level



(b) Both levels are varied



(c) Variable data level

Figure 17: The system throughput under various interference levels

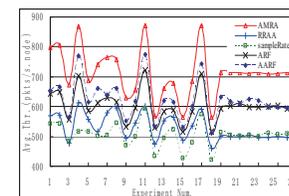


Figure 18: The throughput with multiple data flows

Too much time is spent in probing when the number of such rates is large. For example, when the interference level is 1200pps of size 100bytes, more than two third transmissions are waste in probing the suboptimal rates! Finally, in [15], SampleRate stops the probe for a rate when it experiences four consecutive losses. Such mechanism is however not functioning well since before four losses, the rate is usually switched out so that the protection mechanism is never triggered.

6.4 Complex Topology

We finally perform experiments with multiple data flows in a building. We place six laptops to conduct three data flows. By positioning every destination node in three different locations, we in total get 27 different combinations. The average overall throughput is shown in Fig.18. One can see that: first, consistent trend is observed as in single flow experiments. For example, since the optimal rate here is merely up to 18Mbps, sampleRate performs every poorly in most time. Second, AMRA always yields the best performance. More particularly, as compared to the best one among the four algorithms, the throughput gain of AMRA is 5%-30% and over 20% in most cases.

6.5 Summary

In summary, the above experiments consistently show that: First, the performance of the direct method, i.e. sampleRate, is generally quite poor since very large amount of chances are wasted in keeping track of the performances of the sub-optimal higher rates. Second, the intuitive threshold-based ways, i.e. RRAA, suffers from the infinite cyclic rate switch under stable channel. Third, just using one or few probes in passive fashion, such as in ARF or AARF, can not always understand the channel condition, thus frequently attempt the use of the high rate which is beyond the support of wireless link. Finally, SFB, despite very simple, can identify exactly the channel quality level though sometimes at the expense of a bit more overhead. Such effectiveness is further transformed into the desirable performance of AMRA.

7. CONCLUSIONS

We study the sender-side rate adaptation in 802.11 networks and show that efficient RA is not an easy task even in relatively stable wireless networks due to the insufficient understanding at the sender about the channel, which we identified as inference discrepancy. Furthermore, to mitigate the severe effects, we strive not to resist it directly, but to at first detect the appearance of such discrepancy as quickly and early as possible, and then cancel the corresponding rate decision. More precisely, we add a new phase called rate testing in the RA structure and design a powerful testing mechanism, SFB, to fulfill this task. The evaluations show that our scheme is highly effective. Our work, to the best of our knowledge, is the first to embrace the active measurement in the rate adaptation for 802.11 wireless networks. We believe that this study is also helpful to solving other similar problems.

8. ACKNOWLEDGMENTS

This work is in part supported by the Outstanding PhD student program of National University of Defense Technology (No.B070606) and the Outstanding PhD student scholarship of the Hunan province.

9. REFERENCES

[1] IEEE 802.11, Part II: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. IEEE Std. 1997 2005.

[2] C-C Chen, H. Luo, E. Seo, N. Vaidya and X. Wang. Rate-adaptive Framing for Interfered Wireless Networks. In *IEEE INFOCOM 2007*.

[3] D. Qiao and S. Choi. Fast-Responsive Link Adaptation for IEEE 802.11 WLANs. In Proc. *IEEE ICC 05*, Korea, May, 2005.

[4] J. P. Pavon and S Choi. Link Adaptation Strategy for IEEE 802.11 WLAN via Received Signal Strength Measurement. In Proc. *IEEE ICC 03*, May, 2003.

[5] J. Kim, S. Kim, S. Choi and D. Qiao. CARA: Collision-Aware Rate Adaptation for IEEE 802.11 WLANs. In Proc. *IEEE INFOCOM 2006*.

[6] S. Wong, H. Yang, S. Lu and V. Bharghavan. Robust Rate Adaptation for 802.11 Wireless Networks. In Proc. *ACM MOBICOM 2006*.

[7] D. Aguayo, J. Bicket, S. Biswas, G. Judd and R. Morris. Link-level Measurements from an 802.11b Mesh Network. In Proc. *ACM SIGCOMM 2004*, Sep. 2004.

[8] A. Miu, G. Tan, H. Balakrishnan and J. Apostolopoulos. Divert: Fine-grained Path Selection for Wireless LANs. In Proc. *ACM MOBISYS 2004*, Boston, Jun. 2004.

[9] M. Heusse, F. Rousseau, G. Berger-Sabbatel and A. Duda. Performance Anomaly of 802.11b. In Proc. *IEEE INFOCOM 2003*, Mar. 2003.

[10] A. Kamerman and L. Monteban. WaveLAN-II: A High-performance Wireless LAN for Unlicensed Band. Bell Labs Technical Journal, 1997.

[11] M. Lacage, H. Manshaei and T. Turetli. IEEE 802.11 Rate Adaptation: A Practice Approach. Technical Report, INRIA, France, 2004.

[12] G. Holland, N. Vaidya and P. Bahl. A Rate-Adaptive MAC protocol for Multi-hop Wireless Networks. In Proc. *ACM MOBICOM 2001*.

[13] B. Sadeghi, V. Kanodia, A. Sabharwal and E. Knightly. Opportunistic Media Access for Multirate Ad hoc Networks. In Proc. *ACM MOBICOM 2002*, Sep. 2002.

[14] Z. Ji, Y. Yang, J. Zhou, M. Takai and R. Bagrodia. Exploiting Medium Access Diversity in Rate Adaptive Wireless LANs. In Proc. *ACM MOBICOM 2004*.

[15] J. Bicket. Bit-rate Selection in Wireless Networks. MIT Master Thesis 2005.

[16] I. Haratcherev, K. Langendeen, R. Lagendijk and H. Sips. Hybrid Rate Control for IEEE 802.11. In proc. *ACM MobiWac 2005*.

[17] C-C. Chen, E. Seo, H. Kim, and H. Luo. Self-learning Collision Avoidance for Wireless Networks. In Proc. *IEEE INFOCOM 2006*.

[18] K. Xu and M. Gerla. Effectiveness of RTS/CTS Handshake in IEEE 802.11 based Ad Hoc Networks. In *Journal of Ad hoc Networks*, 2003.

[19] S. Rayanchu, A. Mishra, D. Agrawal, S. Saha and S. Banerjee. Diagnosing Wireless Packet Losses in 802.11: Separating Collision from Weak Signal. In proc. *IEEE INFOCOM 2008*.