

# A HIGH TCP PERFORMANCE RATE ADAPTATION ALGORITHM FOR IEEE 802.11 NETWORKS

Kehao Zhang<sup>1</sup>, Alvin Lim<sup>1</sup>, Shaoen Wu<sup>2</sup> and Qing Yang<sup>1</sup>

<sup>1</sup>Department of Computer Science and Software Engineering,  
Auburn University, Auburn, AL, USA

[zhangke@auburn.edu](mailto:zhangke@auburn.edu), [lim@auburn.edu](mailto:lim@auburn.edu) and [yangqin@auburn.edu](mailto:yangqin@auburn.edu)

<sup>2</sup>School of Computing, The University of Southern Mississippi,  
Hattiesburg, MS, USA

[shaoen.wu@usm.edu](mailto:shaoen.wu@usm.edu)

## ABSTRACT

*Rate adaptation is a link layer mechanism critical to the system performance by exploiting the multiple transmission rates provided by current IEEE 802.11 WLANs. The key challenge for designing such an algorithm is how to select the most appropriate transmission rate under different environments. The first generation rate adaptation schemes perform poorly in a collision dominant environment because they do not differentiate frame losses caused by collision from channel degradation. The second generation schemes use RTS/CTS control frames to differentiate frame losses. However, introducing the overhead may lower network performance especially when the data frame size is small. This paper gives several guidelines on how to design an efficient rate adaptation scheme and proposes an algorithm called Advanced Rate Adaptation Algorithm (ARA). ARA is implemented along with four other representative rate adaptation schemes on a Linux-based testbed. Experiment results show that ARA outperforms other rate adaptation schemes in most scenarios.*

## KEYWORDS

IEEE 802.11, Rate Adaptation, ARA Algorithm

## 1. INTRODUCTION

During the last decade, IEEE 802.11 [1] Wireless LAN (WLAN) has gained wide popularity for broadband wireless networking. IEEE 802.11 standard has been evolving from the earlier basic transmission rate to today's multiple transmission rates. The performance of IEEE 802.11 devices has been improved by exploiting multiple transmission rates. However, to take advantage of such evolution, a mechanism is required to choose the most appropriate rate under different circumstances, namely *rate adaptation*.

Rate adaptation is a link-layer mechanism critical to the system performance in IEEE 802.11 WLANs (Wireless LANs) and has been studied extensively in recent years. The key idea of this mechanism is to select the most appropriate transmission rate under different circumstances. When the channel condition degrades, the link may not be able to support current rate, therefore, frame losses occur and a lower data rate may be more desirable. However, when a frame is lost due to collision instead of channel degradation, the data rate should not be decreased for the following two reasons. First, a lower rate may exacerbate medium congestion because of longer frame transmission time and wider transmission range (more interference). Second, A lower rate is wasteful of bandwidth and unnecessary as channel conditions may well support a higher rate.

Many rate adaptation schemes have been proposed. These schemes can be classified into two generations: without loss differentiation and with loss differentiation. For the schemes without

loss differentiation, they do not differentiate the cause of frame losses and reduce the data rate when the transmission failure count or failure ratio reaches certain threshold. By doing this, these schemes assume that all frame losses are caused by channel degradation with limited congestion losses [2-4]. This restriction limits the use of these schemes. Some of the second generation rate adaptation schemes [5] use RTS/CTS control frames to differentiate frame losses caused by channel degradation and collision. However, there exist several problems. The first problem is the introduction of overhead. This problem is obvious since for each transmitted data frame, RTS/CTS exchange occurs before the data frame. Second, in the IEEE 802.11 standard, RTS is always transmitted at the lowest rate and this is not desirable in certain circumstances.

A proper rate adaptation scheme should be able to differentiate the cause of frame losses. It should also be able to adapt to the channel variations. Since the wireless signal is not stable due to its broadcast nature, an efficient rate adaptation scheme should take the opportunity to increase the data rate when the channel strength is strong and decrease the data rate when the channel strength becomes weak.

Another important factor in designing a good rate adaptation scheme is the rate adjustment metric. The metric can be classified into two categories, either using a threshold or using statistics from the past. Each method has its advantages and disadvantages. Using a threshold in the metric is simple and easy to implement but may be not accurate enough whereas using statistics from the past is more complex but may not be able to adjust the rate quickly. More details about the rate adaptation metric will be discussed in Section 3.1.

The last important part in designing a rate adaptation scheme is their compatibility. Many existing rate adaptation schemes [6-8] require the modification of IEEE 802.11 standard and therefore hard to implement and not easy to collect experimental results. These schemes normally only have theoretical meanings.

This paper investigates these problems and proposes several guidelines corresponding to the above problems. It is organized as follows: In Section 2, the relevant literature work is reviewed. In Section 3, several guidelines for designing an efficient rate adaptation scheme for IEEE 802.11 networks are given. This section also analyzes the problems in current rate adaptation schemes and explains the proposed algorithm in detail. Section 4 explains how ARA is implemented. In Section 5, several existing rate adaptation schemes are selected and implemented to compare with the proposed algorithm. In Section 6, the conclusion of this paper is given. In Section 7, the future work that will be carried out is laid out.

## **2. RELATED WORK**

This section explains the most typical and latest relevant rate adaptation schemes. It is divided into two subsections, namely First Generation and Second Generation.

### **2.1. First Generation: Rate Adaptation without Loss Differentiation**

A frame loss in IEEE 802.11 networks is generally from channel/signal fading or collision. First generation rate adaptation schemes do not differentiate the cause of these losses. In IEEE 802.11 networks, the use of RTS/CTS control frames can minimize the collisions. Since the first generation rate adaptation algorithms intended for networks using RTS/CTS, their lack of loss differentiation is reasonable because most of the data frame losses would come from channel fading.

*Auto Rate Fallback (ARF)* by Kamerman and Monteban [2] is the first rate adaptation algorithm proposed for IEEE 802.11 based wireless networks. It starts the transmission at the highest rate. When the ACK is missed following a successful transmitted package, the first retry

is transmitted at the same rate. If the ACK is missed again, the transmission rate is decreased and a timer is started. When either the timer expires or the successively received ACKs reaches a threshold  $N$  ( $N = 10$ ), the transmitting data rate will increase. A probe packet will be sent at the new rate, if the transmission fails, the system will immediately fall back to the original rate. ARF suffers from two problems. First, it cannot perform efficiently in an environment where the signal strength changes quickly. It either waits for the timer to expire or the success transmission reaches the threshold before the rate can be adjusted. Second, if the channel condition does not change at all, ARF still keeps upgrading the rate when the success transmission count reaches threshold, which will lead to a periodical failures and rate oscillations.

*Receiver Based AutoRate (RBAR)* [6] by Holland, Vaidya, and Bahl is the first rate adaptation that takes advantage of the control frames RTS/CTS transmitted at the basic rate. However, this algorithm suffers from several major problems: 1) The modification of the RTS/CTS control frame is not compatible with current IEEE 802.11 standard and therefore cannot be deployed in current IEEE 802.11 networks; 2) The RTS/CTS frame exchange needs to appear before the transmission of each data frame even if there are no hidden terminals exist and this is a great waste of the bandwidth; 3) It assumes that the SNR of a transmitted packet is available at the receiver, which is not always true.

Pavon and Choi proposed the *Link Adaptation Strategy* for IEEE 802.11 WLAN via Received Signal Strength Measurement (**LA-RSS**) [3]. This rate adaptation scheme is based upon two facts: 1) The frame error probability depends on the received frame length and its transmission rate; 2) The transmitting station can estimate channel behavior by keeping track of the Received Signal Strength (RSS) measured from the frames sent by the Access Point (AP). Besides the problem it does not differentiate the cause of frame losses, this scheme is designed for IEEE 802.11b [10]. For IEEE 802.11g [11], there are 12 different transmission rates. This greatly increases the algorithm complexity and inaccuracy due to the rate adjustment metric it is using.

*Adaptive Auto Rate Fallback(AARF)* by Lacage, Manshaei, and Turletti [4] is a rate adaptation scheme based upon ARF. It reduces the ARF oscillation problem by using a dynamic threshold  $N$  for the successful transmissions required to increase the data rate. If the first transmission fails after the rate is just increased, AARF falls back to the lower rate and doubles the threshold to  $2N$  (with a maximum of 50).

Rate adaptation scheme **SampleRate** [9] by Bicket is based on transmission statistics over cycles. Every tenth data packet, it picks a random rate that may do better than the current one to send the data packet. If the selected rate provides smaller transmission time, it will switch to this rate. SampleRate reduces the number of rates it must sample by eliminating those rates that cannot provide better performance than the current one. The problem with SampleRate is that it may pick a random rate which yields even worse throughput compared to the current transmission rate.

## 2.2. Second Generation: Rate Adaptation with Loss Differentiation

Recent rate adaptation schemes use several ways to differentiate the cause of frame losses and thus respond accordingly to these losses.

Pang, Leung and Liew proposed a rate adaptation scheme with loss differentiation ability called Loss Differentiating-ARF (**LD-ARF**) [8] for IEEE 802.11 WLANs by combining ARF with a loss-differentiating MAC [12] they developed. The loss differentiation is performed at the receiver side in LD-ARF. The authors argue that the frame header in IEEE 802.11 is small and may not be corrupted by channel fading. If the frame header is corrupted, then it must be caused by collision because if two stations transmit in the same time slot, the whole frame is corrupted including the frame header. Therefore, if a frame header can be decoded by the receiver while

the payload cannot, then the cause of the frame loss is due to channel fading. Otherwise, the frame loss is due to collision. If a frame loss is diagnosed as due to channel degradation, a negative ACK (*NACK*) is sent back to the sender to reduce its rate with the assumption that the source address is still available in the decoded frame header. Besides the loss differentiation, all other operations are the same as in ARF. This scheme suffers from several problems. First, the author assumes that the frame header in IEEE 802.11 can only be corrupted by collision, which is not always true. Second, it is possible that the source address is corrupted, which will cause the *NACK* unable to send back to the source. Thus, the scheme will not work in this situation. Third, modification of the IEEE 802.11 standard makes the scheme impractical.

Scheme *Collision-Aware Rate Adaptation*(**CARA**) [5] by Kim *et al.* believes that the effectiveness of a rate adaptation algorithm depends on how fast it can response to the change of channel conditions and also depends on how the collision can be detected and processed. CARA assumes that the transmission error probability of a RTS control frame is negligible because of its small size and robust transmission rate. Therefore, all the failure transmission of RTS control frames is due to collision. In this algorithm, it mandates the use of RTS/CTS control frames in case of a frame loss. Other than turning on RTS/CTS control frames, CARA adjusts the data rate in the same way as ARF.

The *Robust Rate Adaptation Algorithm* scheme (**RRAA**) [13] by Wong *et al.* is a statistical rate adaptation scheme with loss differentiation. It consists of three modules: Loss Estimation, Rate Change and Adaptive RTS Filter. The Loss Estimation module is used to assess the channel condition by using a time window (5 - 40 frames) to keep track of the frame loss ratio. The Rate Change module decides whether to change or keep current rate based on the estimated loss ratio. The Adaptive RTS Filter module is used to selectively turn on or turn off RTS/CTS exchange to reduce the collision losses. However, RRAA is not efficient in some situations. It does not change data rate even if the frame loss is detected due to channel degradation because it does not adjust its rate until the end of each estimated window.

### 2.3. Summary of Rate Adaptation Algorithms

Above rate adaptation algorithms can be grouped under different criteria: whether it differentiate the frame losses, where the data rate is adjusted, and what is the indicator of channel conditions. Table 1 summarizes these criteria.

Table 1. Summary of Rate Adaptation Schemes

Schemes	Loss Differentiation	Based Location	Condition Indicator
ARF	No	Sender Based	Loss ratio
RBAR	No	Receiver Based	SNR
LA-RSS	No	Sender Based	RSS
AARF	No	Sender Based	Loss ratio
SampleRate	No	Sender Based	Loss ratio
LD-ARF	Yes	Receiver Based	Loss ratio
CARA	Yes	Sender Based	Loss ratio
RRAA	Yes	Sender Based	Loss ratio

## 3. DESIGN

This section first gives several guidelines in designing an efficient rate adaptation scheme for IEEE 802.11 networks, then analyzes the problems in current rate adaptation schemes. And finally, the proposed algorithm is explained in detail.

### 3.1. Design Guidelines

#### 3.1.1. Ability to Differentiate Frame Losses

In IEEE 802.11 standard, RTS/CTS control frames exchange is disabled by default to minimize overhead. First generation rate adaptation schemes treat all the frame losses as from channel fading and decrease the data rate whenever the frame failure ratio reaches certain threshold. If a frame loss is caused by collision, decreasing the data rate will not help solve the problem but make it worse. Lower transmission rate means longer transmission time and wider broadcast range, which will lead to more collisions and therefore make the situation worse. An efficient rate adaptation scheme should be able to differentiate the frame losses and respond accordingly to these different causes.

#### 3.1.2. Able to Response to the Variation of Channel Fast

An efficient rate adaptation algorithm should be able to adapt the environment changes fast, otherwise, the algorithm may lose the opportunity to send the data frame at a higher rate or keep sending the data frame at a high rate where successful delivery is not possible.

#### 3.1.3. A Good Metric to Adjust Data Rate

Some of the current rate adaptation schemes adjust the data rate by monitoring the consecutive success and failure count. While this method is simple, it is not accurate. According to RRAA [13], the probability to successfully transmit a data packet following ten consecutive successes is only 28.5%. And the probability of a failure in a data transmission after two consecutive failures is only 36.8%. These statistics show that the consecutive success or consecutive failure count should not be used as the metric to adjust transmission rate. Other schemes use the signal to noise ratio (SNR) as the indicator to adjust the rate. However, according to SampleRate [9] and RRAA [13], SNR is *NOT* a good indicator of the channel condition and therefore should not be used in the metric.

#### 3.1.4. Compatibility with Current Commercial Product

A practical rate adaptation scheme should be compatible with current commercial product, which means we cannot modify the IEEE 802.11 standard. Several existing rate adaptation schemes [6-8] require the modification of IEEE 802.11 standard and therefore are not compatible with current commercial products.

The above four guidelines give us some basic ideas on how to design an efficient and practical rate adaptation scheme. The next section will discuss the proposed algorithm in detail.

### 3.2 Algorithm

In this section, an efficient and practical rate adaptation algorithm called Advanced Rate Adaptation Algorithm (**ARA**) is proposed and explained in detail.

The key idea of ARA is to make use of the RTS control frame as the probe packet. RTS control frame has several characteristics that make it suitable for such use. First, it is very small in size. According to the IEEE 802.11 standard, it is 20 bytes in length. Compared to a normal data packet, which is normally larger than 1000 bytes, a small packet has a smaller probability of colliding with other packets because of its shorter transmission time. Second, RTS/CTS control frames has the ability to reserve the bandwidth. Therefore, a failed packet transmission after a successful RTS/CTS exchange must be caused by channel degradation.

ARA differentiates the frame losses through the use of RTS control frame. When a data frame transmission fails, a RTS control frame will be sent at the *same rate* as the failed frame. If the

CTS control frame can be received, then current channel condition has a high probability of supporting the current transmission rate. Therefore, the loss of the previous data frame is caused by collision. However, if the RTS/CTS exchange is successful but the following data frame fails, then the frame loss must be caused by channel fading. Through this method, ARA can precisely differentiate the frame losses caused by both channel fading and collision.

For the second guideline, ARA uses a fast but precise rate adjustment mechanism. As explained above, if the RTS/CTS exchange is successful but the next data frame still fails, the failure is caused by channel fading. Therefore, the data rate will be decreased in response to such environment changes. If the frame loss is caused by collision, the transmission rate will remain unchanged. The rate adjustment in ARA is fast but accurate given the condition that ARA can precisely determine the cause of frame losses.

Most of the current rate adaptation algorithms use consecutive success count or consecutive failure count as their metric to adjust the data rate. However, this method is not only inaccurate, but also loses the opportunity to grab the short gain period of strong signal. ARA uses RTS and the following data frame to identify channel fading and therefore be able to decrease the transmission rate quickly and precisely. ARA also uses the success count to increase the data rate but it does not necessarily need to be consecutive. As stated previously, the probability of a successful transmission after ten consecutive success transmission is only 28.5%.

By satisfying all of the guidelines displayed in the above section, an efficient and practical rate adaptation algorithm is proposed. Figure 1 shows the state transition diagram for ARA.

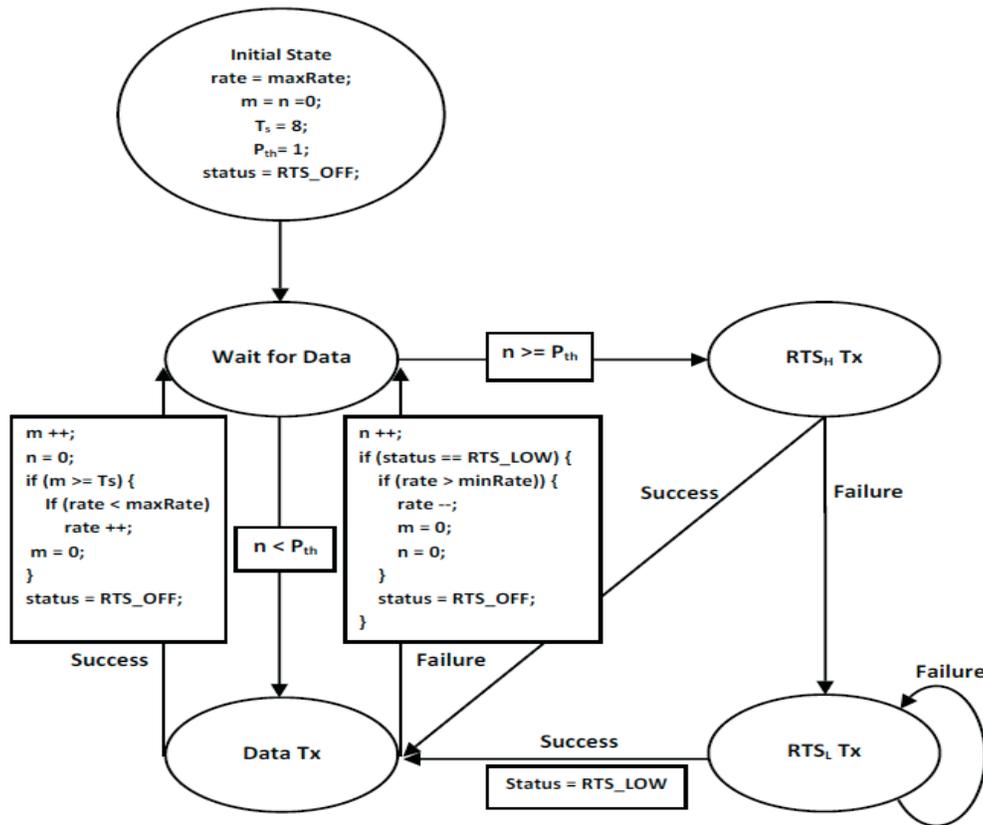


Figure 1. ARA Algorithm

#### 4. IMPLEMENTATION

ARA is implemented on Madwifi [14], which is an open source IEEE 802.11 device driver for Atheros cards in Linux and FreeBSD.

In this version, ARA defines two threshold parameters named  $T_s$  ( $T_s = 8$ ) and  $P_{th}$  ( $P_{th} = 1$ ),  $T_s$  is the threshold used to increase the transmission rate.  $P_{th}$  is the threshold used to initiate the RTS/CTS exchange. If a data transmission fails, ARA will enter a state where RTS/CTS exchange occurs to help differentiate the cause of this frame loss. It also holds two rate index sets, namely  $rix$  and  $cix$ .  $rix$  is used to set the current data transmission rate index and  $cix$  is used to set the current RTS transmission rate index. A rate index corresponds to the real transmission rate, for example, in IEEE 802.11g, there are in total 12 different transmission rates ranging from 1 Mbps to 54 Mbps and the rate index ranges from 0 to 11, respectively.

At the beginning of the state transition, ARA sets the initial rate to be the maximum rate. It also initializes variables  $m$ ,  $n$ , and  $status$ .  $m$  is used to hold the number of frames being successfully delivered at current transmission rate and  $n$  is the failure count. It has to be pointed out that even though  $n$  is the failure count, it is *NOT* used in the rate adjustment metric to decrease the data rate. It is only used as a condition test to decide whether ARA needs to initiate RTS/CTS exchange.  $status$  holds the current status: RTS\_OFF and RTS\_LOW.

After the variables have been initialized, ARA will wait for the coming data frames. Whenever a data frame is found in the transmission queue, ARA will try to send the data without the help of RTS. If the data is delivered successfully, ARA will increase  $m$  by 1 and reset  $n$  to 0. It will also reset  $status$  to RTS\_OFF because next frame will be transmitted without the help of RTS/CTS. When  $m$  reaches the threshold  $T_s$  and current transmission rate is not maximum rate, the rate will be increased. However, if a transmission fails, ARA will increase  $n$  by 1. Note that at this point, we do not reset  $m$  because we do not know whether the frame failure is indeed caused by channel degradation.

When  $n$  is equals to 1, it means the previous data transmission has failed. ARA will try to differentiate the cause of this failure. It will first send a RTS at the same rate as the failed data frame ( $cix = rix$ ), denoted as RTS<sub>H</sub>Tx. If the corresponding CTS can be received, several things can be confirmed. First, the channel condition may support the current transmission rate since RTS is transmitted at the same rate as the failed frame. Second, there is a high probability that the previous frame loss is caused by collision. Third, the bandwidth has been reserved because of the successful exchange of RTS/CTS. Since the cause of the previous data frame loss is collision, we do not need to change the transmission rate index  $rix$ . Success count  $m$  will be resumed because the channel condition has not change. It has to be pointed out that as long as the channel condition does not change,  $m$  should not be changed. This is quite different from other rate adaptation schemes and is one of the reasons why ARA is more efficient.

However, it is still possible that the RTS<sub>H</sub>Tx failed, this happens when the channel condition degrades. Since RTS has a very small size and has a low probability to collide, there is a high probability that the failure is caused by channel fading. In order to confirm this, ARA will send the RTS at the lowest speed ( $cix = 0$ ), denoted as RTS<sub>L</sub>Tx, and also  $status$  will be set as RTS\_LOW. At this point, it has a high probability that RTS<sub>L</sub>Tx will be transmitted successfully because of the small size and robust transmission rate. After the corresponding CTS has been received, the bandwidth has been reserved for the next data packet. Recall that the data rate index  $rix$  is still not changed in the above steps. Therefore the data will still be transmitted using the original rate. It is likely that this frame will not be delivered successfully. Since the cause of this failure is clearly due to channel fading, the transmission rate will be decreased and we reset every variable. ARA falls back to initial state and the whole process repeats.

It is very interesting that in some rare situations, the signal strength is so weak that no frame can be delivered successfully. ARA will enter a state that it will keep sending RTS at the lowest rate. In this case, the RTS is serving as a probe packet to monitor the channel conditions. ARA will not deliver any data frame until at least one successful exchange of RTS/CTS has occurred. At this point, ARA will try to resume the transmission.

## 5. EVALUATION

In order to evaluate the proposed algorithm, several existing rate adaptation schemes are selected. For those schemes whose rate adjustment metric is based upon RSSI or SNR, they are not chosen for the following two reasons. First, RSSI or SNR is not a good indicator of channel conditions as explained in [9, 15]. Second, current WLAN adaptors do not provide RSSI or SNR specification mapping to data rates, which makes the implementation quite difficult. Table 2 summarizes the characteristics of the selected representative rate adaptation schemes.

Table 2. Representative Rate Adaptation Schemes

Loss Differentiation	Consecutive success or failure Count	Loss Statistics
No	AARF	SampleRate
Yes	CARA	RRAA

### 5.1. Experiment Setup

The experiments are conducted on an indoor Linux based testbed, the proposed algorithm is implemented based upon Madwifi Project [14]. Different types of environments are designed to evaluate these rate adaptation schemes. In these experiments, we use Iperf [16] to generate TCP traffic. We have selected three different locations, where Location 1 has the strongest signal strength and Location 3 has the weakest signal strength.

#### 5.1.1. Static Station in a Collision Free Network

This is a typical home network with only one fixed server and one static wireless station. The purpose of this experiment is to evaluate how different rate adaptation algorithms perform in a simple environment where the frame loss is only caused by channel fading. The network configuration is shown in Figure 2.

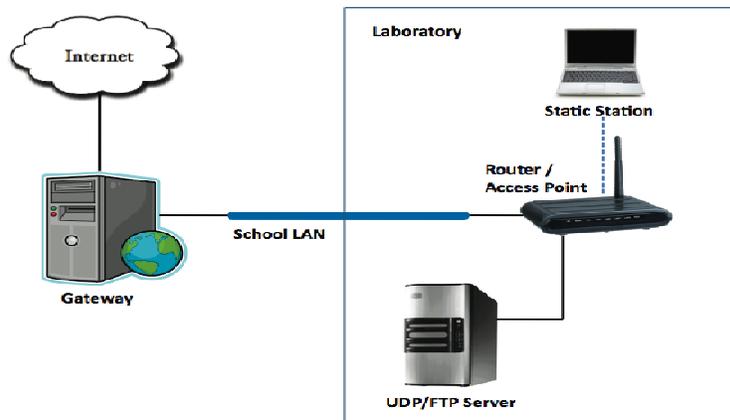


Figure 2. Static Station in a Collision Free Network

### 5.1.2. Static Station in a Collision Dominant Network

Such network topology is often seen in an office environment where many users access WLAN network at the same time. In such an environment, many users may access the AP simultaneously, which may lead to a lot of collision. This is especially true when there is a hidden terminal. To simulate this environment, we put one collision station within the AP's range and may send traffic to the AP within the experiment time. The collision station is loaded with SampleRate, which is the default rate adaptation module in Madwifi. Figure 3 shows the network configuration for this scenario.

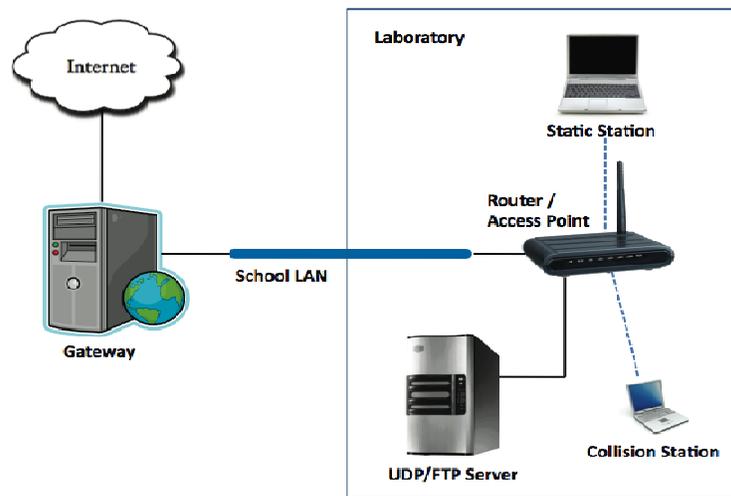


Figure 3. Static Station in a Collision Dominant Network

### 5.1.3. Mobile Station in a Collision Free Network

This experiment emulates a person using his PDA phone to access the WLAN network at home to make a VoIP call or synchronize business emails while moving around in his house. The purpose of this experiment is to measure the impact of mobility on the performance of these algorithms. The network topology is the same as Figure 2, except that we are carrying one laptop and moving it around the AP at a constant speed.

### 5.1.4. Static Station in Campus network

This is the field test for a static station connecting to the school Access Point and keeps sending data to a server connected to the school LAN. The static station is put in the student center where many students are using the same AP. The server is configured in the Shelby Center, which is the main engineering building in our school. The network topology for this field test is shown in Figure 4.

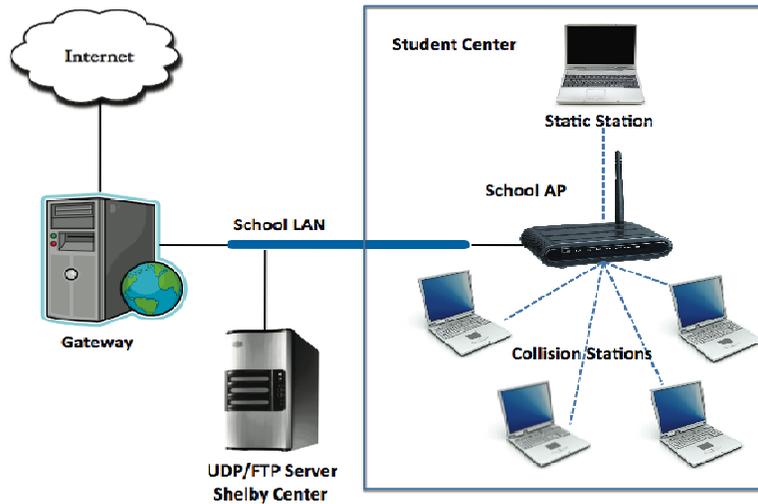


Figure 4. Static Station in Campus Network

## 5.2. Experiment Results

### 5.2.1. Static Station in a Collision Free Network

The experimental results for this scenario are shown in Figure 5. Note that AMRR [4] is the implemented version of AARF [4] in Madwifi [14]. As shown in this figure, ARA performs the best in all these three locations, especially in Location 1 where the signal strength is the strongest. It has to be pointed out that even though ARA performs the best, the throughput of all these schemes are close to each other. The reason is because in this scenario, there is no collision. Therefore, the second generation rate adaption schemes do not have an advantage over the first generation schemes.

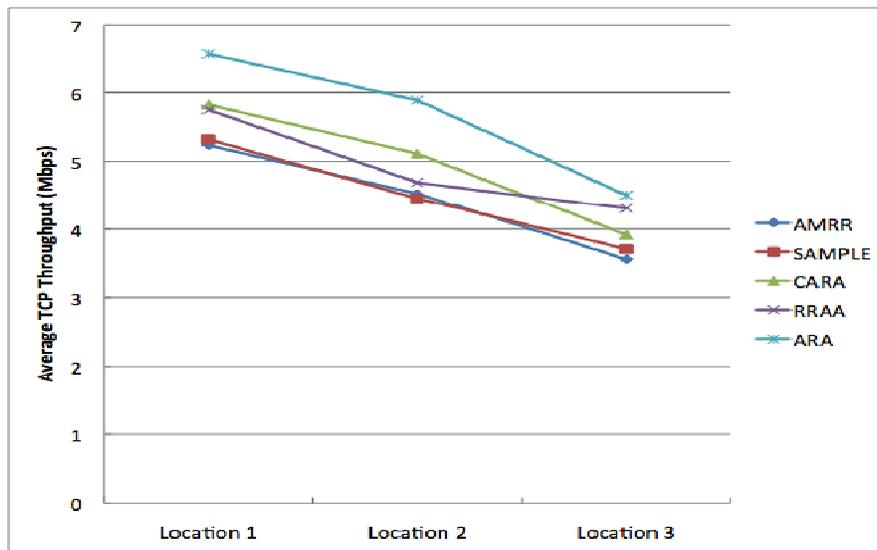


Figure 5. TCP Throughput for a Static Station in a Collision Free Network

### 5.2.2. Static Station in a Collision Dominated Network

This is a situation where both channel fading and collision exists. Figure 6 shows the TCP throughput in this scenario. ARA performs very well when the signal strength is strong, when the channel quality decreases, both channel fading and collision may cause the frame to fail. For the first generation schemes, they do not differentiate the cause of frame losses and therefore perform poorly in Location 3.

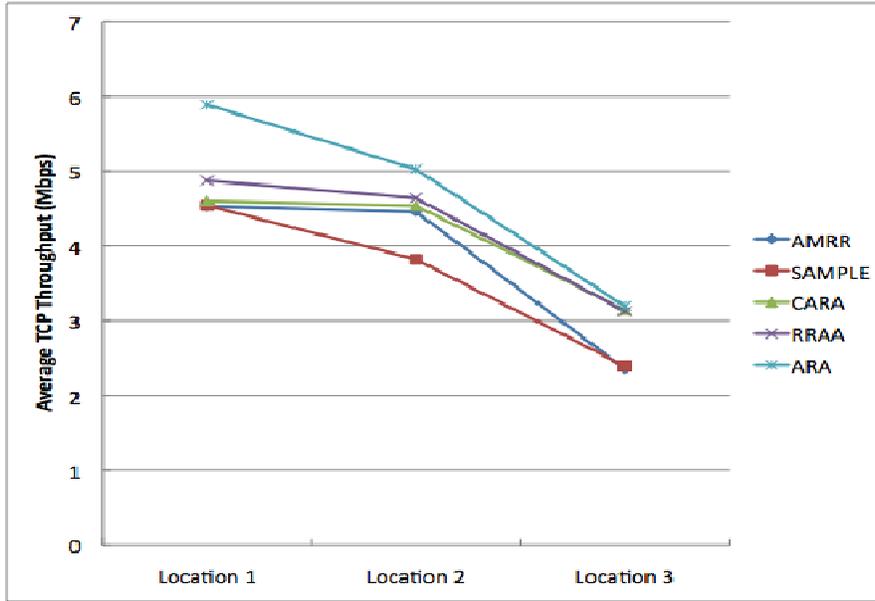


Figure 6. TCP Throughput for a Static Station in a Collision Dominated Network

### 5.2.3. Mobile Station in a Collision Free Network

The experiment results are shown in Figure 7. From this figure, we can see that ARA performs the best among all the schemes. However, the throughput is still close because there is no collision.

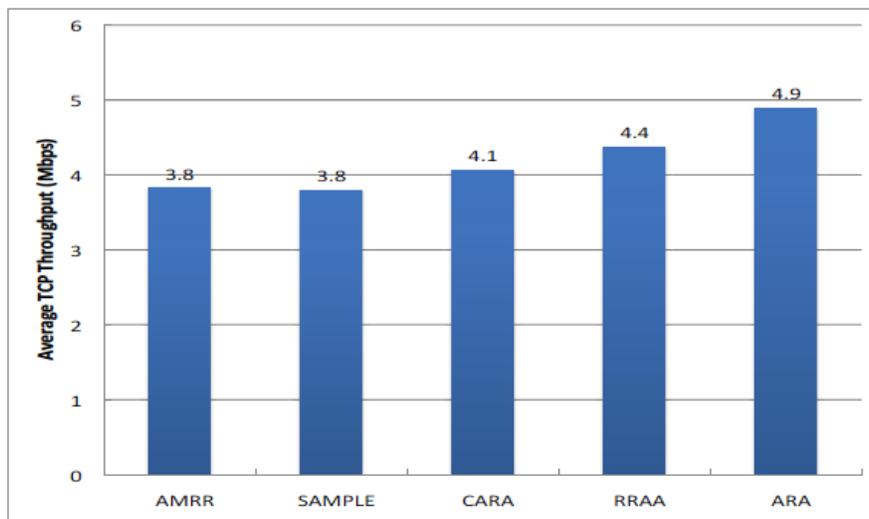


Figure 7. TCP Throughput for a Mobile Station in a Collision Free Network

#### 5.2.4. Static Station in Campus network

This is a realistic scenario. There are a lot of collisions expected in this scenario. The results for this scenario are shown in Figure 8. In this figure, we can see that ARA outperforms all other schemes and the throughput is quite different among these schemes. The reason for this difference is because: 1) In this scenario, there are a lot of collisions. Therefore, most frames losses are caused by collision instead of channel fading. AMRR mistakenly treats all the frame losses as being due channel fading and therefore keep decreasing the transmission rate. Which greatly decrease the throughput; 2) Even though SampleRate [9] does not differentiate the cause of frame losses, it transmits the frames in cycles. At each cycle, it uses a transmission rate that may theoretically produce a better throughput than the previous one. This prevents SampleRate from decreasing the transmission rate as AMRR; 3) CARA [5] and RRAA [13] somehow differentiate frame losses and therefore produce better results compared to AMRR; 4) ARA precisely differentiates the frame losses and provides the best throughput.

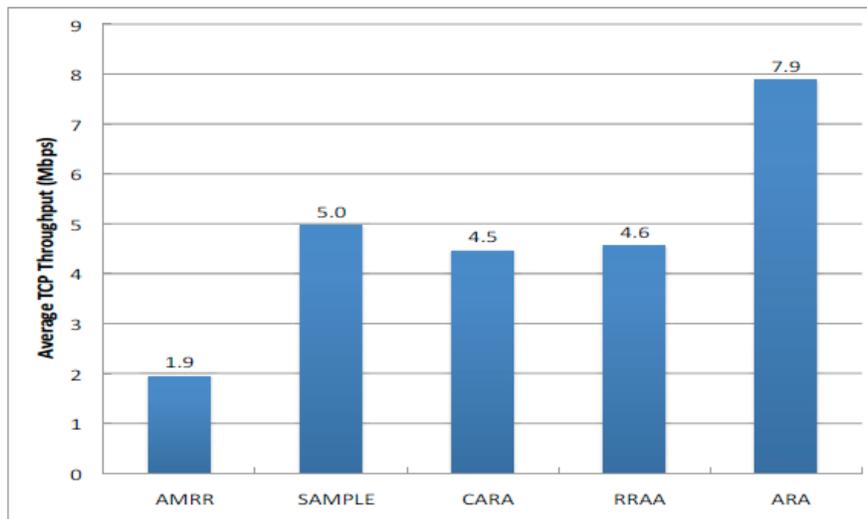


Figure 8. TCP Throughput for a Static Station in Campus Network

## 6. Conclusion

This paper analyzes the existing representative rate adaptation schemes and divides them into two generations. The first generation schemes do not differentiate the frame losses and treat all the losses as being caused by channel degradation and therefore perform poorly in an environment where there are a lot of collisions. The second generation schemes differentiate the frame losses and perform much better than the first generation schemes in the congestion dominant network. However, while most second generation rate adaptation schemes may differentiate frame losses caused by channel degradation, they cannot precisely differentiate the frame losses caused by collision. This paper proposed a rate adaptation algorithm called Advanced Rate Adaptation (ARA) that can precisely differentiate frame losses caused by both channel degradation and collision. In the experiment, ARA is compared to other representative rate adaptation schemes in both controlled experiments and field test. ARA outperforms other rate adaptation schemes in most scenarios. In the field test, through a campus wireless network, ARA provides 300% throughput improvement over some of the first generation rate adaptation schemes and 78% throughput improvement over second generation rate adaptation schemes.

## REFERENCES

- [1] IEEE 802.11 (1999) "Wireless LAN medium access control (MAC) and physical layer (PHY) specifications".
- [2] Ad Kamerman and Leo Monteban, (1997) "WaveLAN-II: a high-performance Wireless LAN for the Unlicensed Band", Bell Labs Technical Journal, vol.2, no.3, pages 118-133.
- [3] Javier del Prado Pavon and Sunghyun Choi. (2003) "Link Adaptation Strategy for IEEE 802.11 WLAN via Received Signal Strength Measurement", In ICC, pages 1108-1123.
- [4] Mathieu Lacage, Mohammad HosseinManshaei, and Thierry Turletti, (2004) "IEEE 802.11 Rate adaptation: A Practical Approach", MSWiM'04, pages 126-134.
- [5] Jongseok Kim, Seongkwan Kim, Sunghyun Choi, and DajiQiao, (2006) "CARA: Collision-Aware Rate Adaptation for IEEE 802.11 WLANs", IEEE INFOCOM'06.
- [6] Gavin Holland, NitinVaidya, and ParamvirBahl, (2001) "A Rate-Adaptive MAC Protocol for Multi-Hop Wireless Networks", ACM SIGMOBILE.
- [7] B. Sadeghi, V. Kanodia, A. Sabharwal, and E. Knightly, (2002) "Opportunistic Media Access for Multirate Ad Hoc Networks," MOBICOM02.
- [8] Qixiang Pang, Victor C.M. Leung, and Soung C. Liew, (2005) "A Rate Adaptation Algorithm for IEEE 802.11 WLANs Based on MAC-Layer Loss Differentiation", In IEEE BROADNETS 2005 - Broadband Wireless Networking Symposium, pages 709-717.
- [9] John C. Bicket. (2005) "Bit-rate Selection in Wireless Networks", Masters thesis, Massachusetts Institute of Technology.
- [10] IEEE 802.11b (1999) "Wireless LAN medium access control (MAC) and physical layer (PHY) specifications: high-speed physical layer (PHY) Extension in the 2.4GHz Band".
- [11] IEEE 802.11g (2003) "Wireless LAN medium access control (MAC) and physical layer (PHY) specifications: further higher data rate extension in the 2.4GHz Band".
- [12] Qixiang Pang, Soung C. Liew, and Victor C. M. Leung, (2005) "Design of an Effective Loss-Distinguishable MAC Protocol for 802.11 WLAN", IEEE Communications Letters, 9:781-783.
- [13] Starsky H.Y. Wong, Hao Yang, Songwu Lu and VaduvurBharghavan, (2006) "Robust Rate Adaptation for 802.11 Wireless Networks", MobiCom06.
- [14] Madwifi. <http://madwifi-project.org/>.
- [15] Daniel Aguayo, John Bicket, SanjitBiswas, Glenn Judd, and Robert Morris, (2004) "Link-level Measurements from an IEEE 802.11b Mesh Network", SIGCOMM04.
- [16] Iperf. <http://iperf.sourceforge.net/>.

## Authors

Kehao Zhang received his B.E. degree in computer science and technology department from Beijing University of Aeronautics and Astronautics (China) in 2001. After graduated, he worked as a lecturer at Taiyuan University of Science and Technology for two years. He received his Ph.D. degree in the department of computer science and software engineering at Auburn University in 2010. His research interests include rate adaptation algorithm in IEEE 802.11 networks, network security and computer graphics.



Alvin Lim is currently an associate professor of computer science and software engineering at Auburn University. He received his Ph.D. degree in computer science from University of Wisconsin at Madison in 1993. His research interests include self-organizing sensor networks, mobile and pervasive computing, network security, wireless networks, reliable and dynamically reconfigurable distributed systems, complex distributed systems, mobile and distributed databases, distributed operating systems, and performance measurement and analysis. He has published widely in journals and conferences in these networking and distributed systems areas. He is a subject area editor of the International Journal of Distributed Sensor Networks. His work had been supported by the National Science Foundation, the DARPA SensIT program, U.S. Air Force Research Lab and the U.S. Army.



Shaoen Wu received the Ph.D. from the Auburn University. He is presently an Assistant Professor of the School of Computing at the University of Southern Mississippi. He has worked at Bell Laboratories (China) and ADTRAN Inc. His current research is in the areas of wireless networking and mobile computing. His research has been funded by various agencies, including the National Science Foundation. Shaoen Wu has served on the committees of several conferences, including as program chair for the ANSS 2011. For more information, please visit <http://www.cs.usm.edu/~swu>.



Qing Yang received his B.E. and M.E. degree in computer science and technology from Nankai University (China) and Harbin Institute of Technology (China) in 2003 and 2005, respectively. He is currently a Ph.D candidate in the department of computer science and software engineering, Auburn University. He is the student member of IEEE and ACM, and has been the Vodafone Fellow from 2005 to 2008. His research interests include target tracking in distributed sensor networks, routing in vehicular networks (VANETs), location security and privacy in VANETs and performance measurement and analysis.

