

An Efficient Mechanism for Data Rate Adaptation in Wireless LAN's

Mohsen Kohan¹, Hassan Khotanlou² and Mohammad Nassiri³

¹ Department of Computer, Hamedan Branch, Islamic Azad University, Science And Research Campus, Hamedan, Iran
Kohan_eng@yahoo.com

² Computer Engineering Department, Bu-Ali Sina University, Hamedan, Iran
Khotanlou@basu.ac.ir

³ Computer Engineering Department, Bu-Ali Sina University, Hamedan, Iran
M.nassiri@basu.ac.ir

Abstract

Today, wireless LANs are growing dramatically. Most of these networks are designed and run based on the IEEE 802.11 standard. Current specification of 802.11 provides multiple transmission rates in physical layer, which are used in various modulations. In order to apply multiple rates capability, the sender must select best transmission rates dynamically at different times according to channel quality and without feedback; choose explicit information from the receiver. Such practice is called the rate adaptation. This paper uses the statistical criteria such as the number of successful/unsuccessful packets and RTS/CTS control frames (Request to Send/Clear to Send) to infer the channel conditions. The proposed method we will be less stringent in increasing the transmission rate from the current rate to the higher one. However, in order to reduce transmission rate from the current rate to the higher one we are very careful. NS2 simulation results confirm the effectiveness of the proposed method.

Keywords: LANs, rate adaptation, IEEE 802.11 standard, simulation.

1. Introduction

Today wireless LANs are growing dramatically. Most of these networks are designed and run based on the IEEE 802.11 standard. This standard includes details of the physical layer and access to common medium sub-layer.

Transmission rate adaptation is considered as a challenge to the performance of wireless networks in the IEEE 802.11 [1], current specifications of 802.11 provide physical multiple transmission rates in physical layer, For example, 802.11b [2] supports four rate encoding (1, 2, 5.5, 11 Mbps), 802.11a [3] provides eight rates (6, 9, 12, 18, 24, 36, 48, 54 Mbps) and 802.11g [4] supports all twelve encoding rate. Rate adaptation algorithms should select the best transmission rate dynamically at different

times according to channel quality and without feedback. Note that the best rate is not necessarily the higher rate; in some cases, going to the higher transmission rate reduces channel efficiency. The higher the rate is proportional to that; the probability of error in the channel becomes greater. Also if the lower rate is used to send, more time is spent to send packets, but the risk of errors in the channel is reduced; so choosing the best rate available can have a significant impact on the channel performance.

Most of the existing algorithms for transmission rate adaptation have used the idea that have considered a threshold to increase or decrease the transmission rate, which varies according to the adopted policy, for example, in some algorithms, statistical criteria such as the number of successful/unsuccessful packets, are used to determine the rate. So that in order to increase the rate, when a certain number of consecutive data packet reach the receiver, the rate is increased, and if it does not reach the receiver, rate is reduces such as ARF [5] and AARF [6] and SampleRate [7] and CRAA [8]. Several algorithms have also examined the channels through RTS/CTS control packets find channel quality and set the transmission rate, with respect to the fact that RTS/CTS (Request To Send / Clear To Send) Explore frames are very compact and has a low cost of sending and receiving, they are an option to check the quality of the channel, like CARA [9] and OAR [10] and RBAR [11]. The physical criteria such as SNR (Signal to Noise Rate) are also used to infer the quality of the channel. So that, based on the received signal strength can find channels quality such as RBAR and OAR. In some algorithms a combination of the above criteria are used, such as HRC [12]. Some algorithms use short-term and long-term data and a History, to infer the quality of channel such as Sample Rate and RRAA [13]. In recent years Algorithms have been proposed, that use of new criteria for giving feedback

to the sender using ACK [14], or the Markov chain and history of sending to infer the channel conditions [15]. In this article, statistical measure of the number of successful/unsuccessful packets and RTS/CTS control packets are used to infer the quality of channels in order to determine the transmission rate, full details are described in the relevant sections.

In this article, first we introduce the algorithms related to data transmission rate adaptation, and then discuss our proposed method. Finally, we compare the results with other algorithms.

2. Related Works

When hidden stations exist within the network, the network performance will be reduced. To resolve this problem, the exchange of two small control frames called RTS and CTS are used. Exchange of RTS/CTS is shown in Figure 1, so that wireless channel is reserved for transmitter station, after a successful exchange of RTS/CTS. According to the 802.11 standard, the decision making about sending RTS frame is on the sender. RTS frame is used when the frame size is greater than or equal to the RTS threshold. In most 802.11 products, RTS threshold is set to a large value, i.e. 2347, this means that the exchange of RTS/CTS is disabled.

It is clear that the exchange of RTS/CTS in the highly competitive wireless environment is particularly useful, since most of the unsuccessful sending are because of collision. Exchange of RTS/CTS is used for large data frames, because the exchange of RTS/CTS overhead is together with overload [16]. The author [16] showed that the value of much wasted time due to the collision for RTS/CTS mechanism is inconsiderable, regardless of the number of competing stations and collision chances.

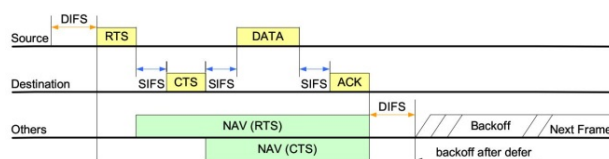


Figure 1 exchange mechanism for RTS / CTS in IEEE 802.11 standard

Exchange mechanism for RTS/CTS is shown in Figure 1, Source is indicative of data frame station, Destination represents recipient data frames and Other shows the other stations that are in range of the sender and the recipient. In this mechanism, the sending station sees the empty channel before sending data frames, a short

duration of the DIFS (Distributed Inter-Frame Space) is known to wait, and then sends RTS frames (RTS frames contain control information and time frames are sent), Stations that are located within the sender range, receives the RTS frame and NAV Vector (Network Allocation Vector) set itself to the end of the related sending data frames (this is also a corresponding acknowledgment message), and by the end of this time, no data is sent. in response to RTS Frame, the receiver after a short period known as the SIFS (Short Inter-Frame Space) (SIFS duration is less than DIFS) sends a CTS frame, stations that are in the range of recipients receive CTS frame, stations that receive the CTS frame set their NAV to end sender to send the frame, and upon receiving the CTS frame, the sender waits for the short time (SIFS) and then send its data frame. Thus, after a successful exchange of RTS/CTS it is ensured that the sender can send data safely, because the channel is reserved for it.

In recent years, several algorithms have been proposed for adaptive transmission rate and some of them have been used in actual products. The basic idea is to estimate the channel quality, and according to these estimations, the transmission rate can be adapted. This estimation is collected based on some measures of quality. Widely used criteria include, check the envelopes, the successful/failed sequence, physical criteria such as SNR and long-term statistics. The main idea of this paper is derived from the successful/failed sequence and Envelopes review. This section explains the algorithms that use these criteria.

First rate adaptation algorithm in wireless networks, namely ARF was introduced in 1997. This algorithm uses statistical idea of envelopes sending to estimate the channel quality. If two consecutive acknowledge message is not received correctly, the sender reduces its transmission rate to the one lower rate, and the timer is restarted. But to increase the transmission rate, it should receive more than ten consecutive authentication message, or timer comes to the end, then a packet to identify the quality of the channel is transmitted at higher rates, if the authenticated packet message was received correctly, the following transmissions will be sent at the rates the parcel is posted (i.e. higher rates), but if the authentication message is not received correctly, the rate will remain the same and the timer is restarted. ARF algorithm is easy to implement with no change in the 802.11 standard, and it is easy to manage and correct the sender, so it is actually beneficial. But when the covert channel exists in station and there are random encounters, it is weak. Another disadvantage of this algorithm is that it should be implemented step by step, it means that cannot be

switched from the current rate to either two higher or lower rate.

ARF algorithm on the basis of research in adapting the transmission rate in wireless networks, the heart of this algorithm in a way that many algorithms were developed, including AARF, Sample Rate and CARA, we briefly explain each. AARF algorithm with a slight modification in the algorithm can increase the efficiency of the ARF. The only way to increase the rate of ten consecutive sending, but due to the dynamic conditions of a different number of channels to be elected, as follows:

$$ARF \begin{cases} \theta_u^{ARF} = 10 \\ \theta_d^{ARF} = 2 \end{cases} \quad (1)$$

$$AARF \begin{cases} \theta_u^{AARF} = \begin{cases} \min(2 \cdot \theta_u^{AARF}, 50) \\ 10 (= \theta_u^{ARF}) \end{cases} \\ \theta_d^{AARF} = 2 \end{cases} \quad (2)$$

Sample Rate algorithm works as well as the ARF algorithm, with a little and creative difference, so that it selects transmission rate according to part of transmission time, thus:

- It holds Expected time for any rate and is updated after each post. (wnd = 1s-10s)
- A minimum time frame rate is expected to be sent.
- After ten frames, one packet at another randomly rate is selected and sent.
- It will switch to a lower rate after four successive failed transmissions.

CARA is also modeled by the ARF algorithm, but with an innovative idea could enhance the efficiency greatly. The main idea of CARA is based on the fact that the frame must separate frame encounters and missed posts due to channel error occurred and distinguish the difference between the two. Because the frame is not an issue in terms of separate channel and rate adaptation, quite error rate adaptation depends on the channel, and the rate adaptation algorithms based on dynamic channel conditions. CARA has put its main focus on reducing the rate and for increase rate serves as the ARF, it means after receiving the confirmation message sequence ten switches the transmission rate to a higher rate, and to reduce the rate uses two methods.

- ❖ Evaluation of Adaptive RTS
- ❖ collision detection through detection of CCA

In study of the adaptive RTS it is assumed that all RTS failures are due to collisions, and unsuccessful send after exchanging RTS/CTS must be due to channel errors. Also to reduce the overhead of the RTS, exchange RTS/CTS frames sent only when the data failed to be activated.

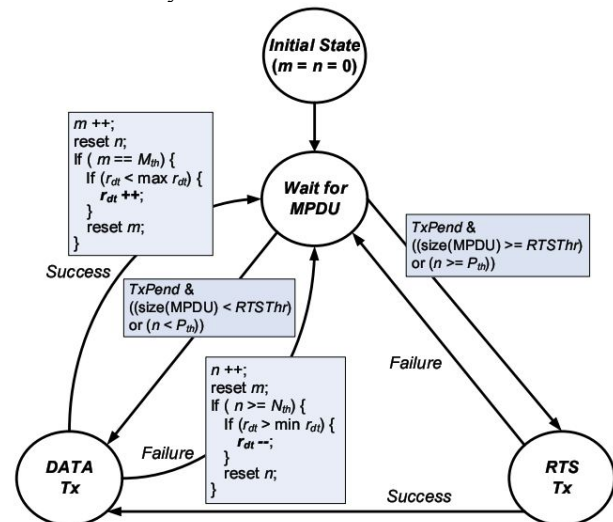


Figure 2 State transition diagram of RTS Probing in CARA

CARA in default sends data frame without exchanging RTS/CTS. However, if a failure occurs, the exchange of RTS/CTS is active for the next post, if this message is again unsuccessful; the rate switches to a lower rate. But if the post was successful, the next outgoing frame rate remains without exchanging RTS/CTS.

Table1 LIST OF NOTATIONS USED IN THE RTS PROBING PROCEDURE (CARA)

parameters	Discussion
M	Number of consecutive successful
N	Number of consecutive failed
Mth	Threshold of consecutive successful
Nth	Threshold of consecutive failed
TxPend	There are data to be sent
Rdt	An array of transmission rate 802.11a={6, 9, 12, 18, 24, 36, 48, 54 Mbps} 802.11b={1, 2, 5.5, 11 Mbps}
rdt	Transmission rate for an element Rdt
++	Transmission rate increase to one higher
--	Transmission rate decrease to one lower
Pth	RTS threshold to activate survey
RTSThr	activation threshold of RTS in standard

Details of the RTS by diagram of state transition of transmitter station are explained in Figure 2. So that the parameters in the figure are listed in Table 1. There are four modes in Figure.

- **The initial state:** is the starting point.
- **Wait for the data:** station is placed in this situation that have a new frame to send or the current frame transmission is encountered an error and needs to be re-sent.
- **Data transmission:** station is in this mode when the data is sent and station waits for a confirmation message.
- **RTS transmission:** station is in this mode when RTS frame is sent and it is waiting for CTS corresponding frames.

As shown in Figure 2 and Table 1, the number of consecutive failed is specified by n , that is compares with two different thresholds, the threshold of activation (P_{th}) and consecutive unsuccessful threshold (N_{th}). When n was equal to P_{th} , frame exchange RTS/CTS before sending the next frame is done, and, when the n reaches to the N_{th} , following data transmission can be done at a lower rate. Default values for P_{th} and N_{th} are 1 and 2 respectively.

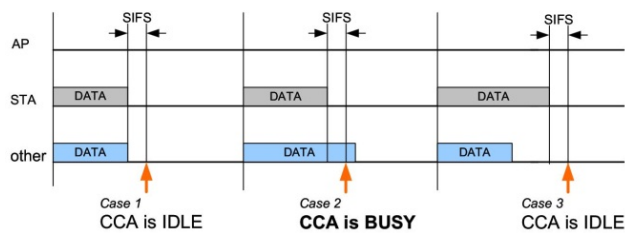


Figure 3 Three possible cases of collision. In the second case, the collision can be detected via CCA detection.

But in detection method through CCA discovery (Clear Channel Assessment), according to (3), collisions can be divided into three categories. In the first case, the data frames which are responsible for collision are at the same size, by the second and third frames are different. In the figure, STA is transmitter stations and other is other senders within channels. Detection method through CCA discovery works as follows.

In SIFS time after getting all the data, just when it is waiting to receive the acknowledgment message of transmitted frames it addresses the channel. If the channel is busy, collision is certainly occurred. If the channel is not busy, it cannot exactly say that this post has been unsuccessful due to collision or channel error. According to (3) figure only by two cases collision through the detection CCA can be discovered, and in other two cases, discovering CCA cannot help us.

3. The proposed method

Our proposed method is derived from CARA, CARA has put its main focus on reducing the rate, and with innovative ideas could greatly enhance the performance, and CARA idea in reducing the rate is so that it distinguishes between failed sending for collision and failed sending for channel error. Since for the collision the rate should not be reduced and collision by any rate cannot be prevented. CARA uses two methods for the detection of RTS and a diagnosis of CCA, both of which are described in the related sections. CARA acts the same as ARF to increase the rate; it means the rate may increase after ten consecutive posts. In this paper, our main focus is on reducing the rate, as opposite to CARA, we will focus on increasing rate, and in reducing rate we do just like CARA.

Table2 LIST OF NOTATIONS USED IN THE RTS PROBING PROCEDURE (Proposed Algorithm)

parameter	descriptions
M	Number of consecutive successful
N	Number of consecutive failed
M_{th}	Threshold of consecutive successful (set in 10)
N_{th}	Threshold of consecutive failed (set in 2)
$TxPend$	There are data to be sent
Rdt	An array of transmission rate 802.11a={6, 9, 12, 18, 24, 36, 48, 54 Mbps} 802.11b={1, 2, 5.5, 11 Mbps}
rdt	Transmission rate for an element Rdt
++	Transmission rate increase to one higher
--	Transmission rate decrease to one lower
P_{th}	RTS threshold to activate survey (set in 1)
$RTSThr$	activation threshold of RTS in standard
$M1_{th}$	Consecutive successful threshold, first level (set in 8)
$M2_{th}$	Consecutive successful threshold, second level (set in 14)
$M3_{th}$	Consecutive successful threshold, third level (set in 18)

Our proposed method uses three levels to increase the rate. This means we have to increase the rate in three thresholds, the first threshold that we show by $M1_{th}$ and its amount is equal to 8, following 8 consecutive successful sending goes to a higher transmission rate, this value is set to 10 in CARA, CARA After 10 consecutive successful post goes to a higher rate, and restart counter of successful consecutive sending (m) (make it zero). In our proposed method, after the rates go up, the counter successful consecutive sending (m) is not restarted. Second threshold we show in $M2_{th}$ is equal to 14, following 14 consecutive successful sending goes to two



higher transmission rate step by step (i.e. after 8 consecutive successful message sending to a higher rate and after next 6 consecutive success sending goes up to two higher rates). And the third threshold showed by M3th is equal to 18, following 18 consecutive successful sending goes to two higher transmission rate step by step. The code emulator is used to describe the problem better. Code snippet (1).

```

1  m++;
2  reset n;
3  if( m==M1th){
4      if( rdt<max rdt){
5          rdt++;}
6  else if( m==M2th){
7      if( rdt<max rdt){
8          rdt++;}
9  else if( m==M1th){
10     if( rdt<max rdt){
11         rdt++;
12         m=15;}}
```

code 1 The successful sending in proposed method

List of parameters used in the code (1) is given in Table 2. When we have a successful sending, Code (1) is run, in the first line, one is added to the number of consecutive successful sending, and then in line 2, the number of consecutive failed sending is restart (set to zero), and then the third line checks whether the threshold level is equal to the number of consecutive successful or not, If there is an equal or higher rate switches to a higher rate, and if not equal, it is compared with a second threshold (line 6), If the number of consecutive successful sending is equal to the second threshold, i.e. 14 consecutive successful have been submitted, and during the 14 consecutive successful sending, transmission rate have increased twice (once in the eighth and another in the Fourteenth times) and if is not equal to the second threshold, the third threshold is checked (line 9), If the number of consecutive successful third threshold is the 18th consecutive successful have been submitted. And during the 18 consecutive successful sending, the transmission rate has increased three times (ranked eighth and fourteenth and eighteenth). Another thing that remains is that when we got to the third level, the number of consecutive successful we set to 15 (line 12). This is because if we had more than 18 sending in a row the third level remain, and then after a successful three consecutive, rate goes up. It should be noted that after a send fails, the number of consecutive successful sending is restarted (i.e. set to zero).

```

1  m++;
2  reset n;
3  if( m==Mth){
4      if( rdt<max rdt){
5          rdt++;
6      }
7      reset m;
8  }
```

code 2 The successful sending in the CARA

Code (2) in the CARA, is run after successful sending; in this piece of code, after adding one to the number of consecutive unsuccessful sending, and restarting consecutive failed sending, we check the number of consecutive unsuccessful sending and consecutive unsuccessful sending threshold (the threshold is set to 10 consecutive successful in CARA). If it is equal, and the higher transmission rate exists, it switches to a higher rate, and then restarts the number of successful sending (line 7). This is not the same with our proposed code (code 1), it means after going to the higher rate, the number of consecutive successful sending is not restarted. However, this (i.e. restart the number after consecutive successful sending) is done when facing failed sending.

```

1  n++;
2  reset m;
3  if( n==Nth){
4      if( rdt>min rdt){
5          rdt--;
6      }
7      reset n;
8  }
```

code 3 The failed sending in the CARA and proposed method

Code (3) is common in CARA and our proposed method, and is run if we have failed sending, so that initially, it adds one to the consecutive failed sending, and then restart consecutive successful sending (set to zero). And then examine the number of failed consecutive sending or failed consecutive sending threshold (the threshold of consecutive unsuccessful in CARA and our proposed method was set to 2), if it is equal and the lower the rate exists, it switches to the lower rate, and then restarts the number of consecutive failed sending.

To better understand and describe the problem, we see the state transition diagram of the proposed method in Figure

(4). This graph is like the graph depicted in CARA, with the difference in the successful frame sending. Its parameters are listed in Table. In this diagram, there are four states which are described in related work. So we have just mentioned them in the graph.

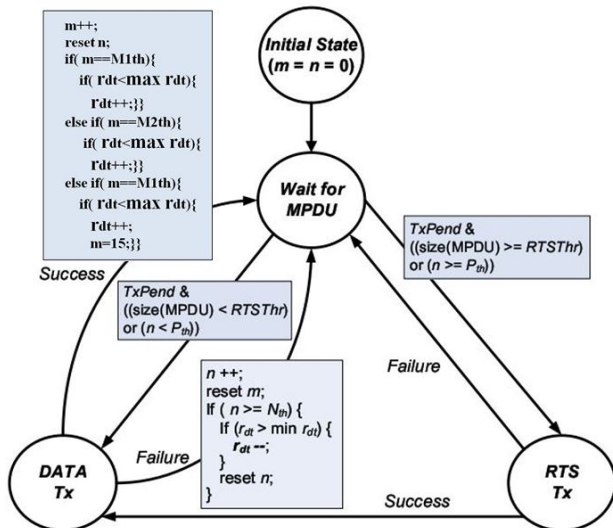


Figure 4 State transition diagram of RTS Probing in proposed algorithm

Initially, the station will want to begin work is on the initial state, in which the number of consecutive successful sending (m) and the number of consecutive failed sending (n) is zero, and it goes to waiting state. Then, if there is data to send, we have two modes, one going to RTS transmission mode, and the other, going to data transmission mode. It goes to RTS mode when size of sent data (size (MPDU)) exceeds RTS threshold (RTSThr), or the number of consecutive failed sending is Equal or greater than the threshold number of survey of RTS (Pth). In this mode, before sending the data, RTS/CTS exchange is used. But if the dialed size is smaller than the threshold of RTS and RTS threshold is lower than the number of consecutive failed sending, data transmission mode is used. In data transmission, data transmission is successful or failed, in each case, the related code runs and goes to standby mode.

4. Evaluation and comparison

We performed our proposed algorithm with ARF and CARA, to measure and compare the throughput (throughput) in NS2 [17]. Ad-hoc wireless network is an unstructured and decentralized network. Geographic area or range of ad-hoc network is all the points where the nodes are expanded. In ad-hoc networks, each node is equipped with a transceiver that makes communication

with other nodes in the network possible. In this type of network, flows are multichip. In multichip flows transceiver stations are located in a square area of 700m * 700m, Transfer threshold (RXThreshold) of network is 1000 meters, in the radio waves model, Two Ray Grand is used. Time of each simulation run is 50 seconds and the results presented are obtained from averaged over 10 runs. Each traffic type is FTP, which means that all the time there is traffic on the transmission queue. Each link's transmission queue holds up to 50 packets, and each packet size is 1000 bytes.

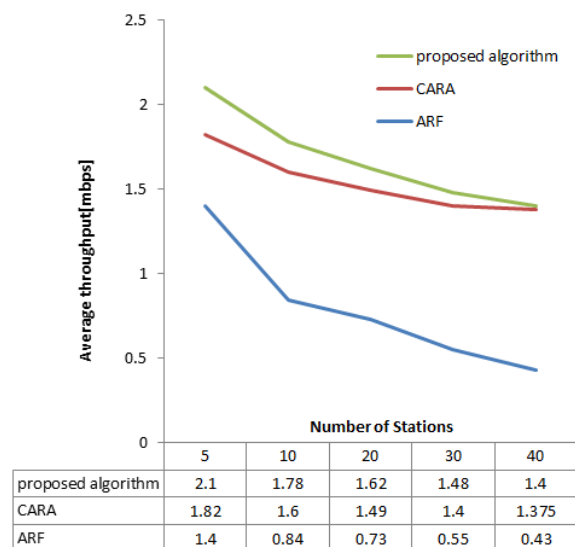


Figure 5 Throughput

Figure (5) evaluates throughput parameter with different number of nodes in a multihop network. Total throughput of ARF and CARA and proposed model is presented in the figure (5). The way calculate the throughput of the network depends on how efficiently a packet uses the channel when it is sending, as a result, to calculate it, the length packet should be divided into total channel time in order to achieve efficiency of channel for each node.

Increased throughput achieved by the proposed algorithm, compared with ARF is in average about 54% and compared with CARA is about 7%.

References

- [1] IEEE 802.11, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, IEEE Std. 802.11-1999, Aug.1999.
- [2] IEEE 802.11b, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications:



- Higher-speed Physical Layer Extension in the 2.4 GHz Band, Supplement to IEEE 802.11 Standard, Sep. 1999.
- [3] IEEE 802.11a, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: High-speed Physical Layer in the 5 GHz Band, Supplement to IEEE 802.11 Standard, Sep. 1999.
 - [4] IEEE 802.11g, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Further Higher Data Rate Extension in the 2.4 GHz Band, Supplement to IEEE 802.11 Standard, Jun. 2003.
 - [5] A. Kamerman and L. Monteban, "WaveLAN-II: a high-performance Wireless LAN for the Unlicensed Band," Bell Labs Technical Journal, vol.2, no.3, pp.118-133, Aug. 1997.
 - [6] M. Lacage; M. H. Manshaei, T. Turletti, "IEEE 802.11 Rate Adaptation: A Practical Approach," ACM MSWiM, 2004.
 - [7] M. Lacage, M.H. Manshaei, and T. Turletti, "IEEE 802.11 Rate Adaptation: A Practical Approach," Proc. ACM Int'l Symp. Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM '04), 2004.
 - [8] C. Yiu; S. Singh, "A Model for Comparing Rate Adaptation Algorithms," in Proceedings of the Fourth ACM Workshop on Wireless Network Testbeds, Experimental Evaluation and Characterization, WINTECH, Beijing, China, 2004.
 - [9] J. Kim, S. Kim, S. Choi, and D. Qiao, "CARA: Collision-Aware Rate Adaptation for IEEE 802.11 WLANs," Proc. IEEE INFOCOM, 2006.
 - [10] B. Sadeghi, V. Kanodia, A. Sabharwal and E. Knightly, "Opportunistic Media Access for Multirate Ad Hoc Networks," in Proc. ACM MobiCom'02, Atlanta, Georgia, Jul. 2002.
 - [11] G.Holland, N.Vaidya, and P.Bahl, "A Rate-Adaptive MAC Protocol for Multi-Hop Wireless Networks," in Proc. ACM MobiCom'01, Rome, Italy, Jul. 2001.
 - [12] I. Haratcherev; K. Langendoen, R. Lagendijk, H. Sips, "Hybrid Rate Control for IEEE 802.11," ACM MobiWac. 2004.
 - [13] S.H.Y. Wong, H. Yang, S. Lu, and V. Bharghavan, "Robust Rate Adaptation for 802.11 Wireless Networks," Proc. ACM MobiCom, 2006.
 - [14] I.Kim; Y.T.Kim, "Realistic Modeling of IEEE 802.11 WLAN Considering Rate Adaptation and Multi-Rate Retry," IEEE Transactions on Consumer Electronics. 2011.
 - [15] X. Chen; P. Gangwal; D. Qiao, "RAM: Rate Adaptation in Mobile Environments," IEEE Transactions on Mobile Computing, vol. 11, no. 3, March. 2012.
 - [16] G. Bianchi, "Performance Analysis of the IEEE 802.11 Distributed Coordination Function," IEEE Journal of Selected Area in Communications, vol. 18, no. 3, March. 2000.
 - [17] Network Simulator - NS2 , <http://www.isi.edu/nsnam/ns/>, online link.