A Performance Evaluation for Rate Adaptation Algorithms in IEEE 802.11 Wireless Networks

M.A. Mohamed
Electronics and Communications Engineering Department
Faculty of Engineering – Mansoura University
Mansoura – Egypt

W.M. Bahget
Information Technology Department
Faculty of Computer and Information Sciences – Mansoura University
Mansoura - Egypt

S.S Mohamed
Electronics and Communications Engineering Department
Faculty of Engineering – Mansoura University
Mansoura - Egypt

ABSTRACT
The wireless LAN standards IEEE 802.11 (a/b/g) provides multi-rates capabilities. To achieve high network performance, devices need to adapt their transmission rate dynamically under varying conditions. Rate adaption algorithm is a critical component of their performance, only very a few algorithms have been published and the differences between their mechanisms have been publicly discussed. In this paper rate adaption algorithms categorized according to the metric that sender and/or receiver used to calculate the best bit rate uses to transmission date. In addition, a comparative study conducted in IEEE 802.11-based ad-hoc networks of most existing rate adaptation algorithms.

General Terms
Performance, Algorithms, Simulation, Wireless Network

Keywords
IEEE 802.11, Rate Adaption, ns3.

1. INTRODUCTION
Wireless channels are extremely variable and can be affected by a number of different factors, such as interference from other wireless devices, multi-path fading and signal attenuation [1].

The current IEEE 802.11 specifications mandate multiple transmission rates at the physical layer (PHY) that use different modulation and coding schemes. For example, the IEEE 802.11b PHY supports four transmission rates (1~11 Mbps), the IEEE 802.11a PHY offers eight rates (6~54Mbps), and the IEEE 802.11g PHY supports twelve rates (1~54Mbps).

As such, one of the key components of an IEEE 802.11 system is the rate adaptation mechanism, which adapts the data rate used by a wireless sender to the wireless channel conditions. If sender use a rate that is too high, many of the packets will be dropped due to bit errors, however if we use a rate that is too low, the wireless channel is not fully utilized.

The effectiveness of a rate adaptation scheme depends on how fast it can respond to the variation of wireless channel. In addition, in a multi-user environment where frame collisions are inevitable due to the contention nature of the 802.11 DCF, the effectiveness of a rate adaptation scheme also depends greatly on how the collisions may be detected and handled properly.

The proposed rate adaptation schemes can be roughly grouped into two categories [15]. First, open-loop rate adaptation algorithms do not consider the collision effect, and hence, may malfunction severely when many transmission failures are due to collisions. Second, closed-loop rate adaptation algorithms which diagnose the cause of a loss and appropriately adjust the data rate. Most of them are designed to work in realistic scenarios without RTS/CTS.

In recent years, a number of algorithms for rate adaptation [2-11] have been proposed in the literature and some [2-6] have been used in real products. Their basic idea is to estimate the channel quality and adjust the transmission rate accordingly. This is typically achieved by using a few metrics collected at the sender and the associated design rules, consecutive successes/losses [2-7] and PHY metrics such as SNR [9-11].

Rate adaptation schemes proposed for IEEE 802.11 networks, Characteristics and common features are summarized and evaluate through the ns3 simulation [23]. The rest of the paper is organized as follows. Section 2: gives an overview of related work on rate adaptation; Section 3: Performance evaluation of five rate algorithms based on ns3 simulation tool. Section 4: concludes the paper.

2. RATE SELECTION ALGORITHMS IN IEEE 802.11 NETWORKS

2.1 Open-loop Based Approaches
Use either consecutive packet transmission successes or failures, or packet delivery ratios in a time window, to sequentially increase or decrease bit rates. Frame-based are in nature not responsive to variations of link status, because they require multiple frame transmissions to converge to a meaningful estimated value of link status. In addition, they usually adopt the fixed threshold mechanism (e.g., 10 consecutive successes or 1 failure) to adjust the bit rate, which cannot work well in different environments. ARF, AARF, AMRR, SampleRate, Minstrel and Onoe are collision-ignored and reduce bit rates once packet loss occurs [16][17].

2.1.1 Automatic Rate Fall back (ARF)

The idea behind ARF is that each sender tries to use a higher transmission rate after a set number of successful transmissions at the current rate. If it experiences one or two consecutive losses, it falls back to the next lower rate. When ARF increases the sending rate, the subsequent transmission decides if ARF will continue to use the higher rate or fall back to the previous lower one. The first packet is often referred to as a probing packet [2].

$$\theta_U^{ARF} = 10 \qquad \theta_d^{ARF} = 2 \tag{1}$$

First problem in AARF, it does not perform well if the channel conditions change very quickly. Second, even though the channel condition does not change at all or change very slowly, it change it to the next higher bit-rate every 10 successful transmissions. As a result, it may increase error rate, and reduce the throughput.

2.2.2 Adaptive ARF (AARF)

AARF tries to optimize the ARF problem by using history of the channel, and increase the number of consecutive successful transmissions if the channel is stable and works best with a fixed transmissions rate.

AARF behaves more or less like ARF, but unlike ARF it increases the number of consecutive successful transmissions it needs before it tries to send a transmission at a higher rate than the current one. It does this by remembering the number of failed attempts to probe the channel at a higher rate, and each time the probe transmission fails, the algorithm multiplies the number of consecutive successful transmissions by two, up to a maximum of 50. If a packet fails twice while in the current transmission, it lowers the transmission rate one step and resets the consecutive successful transmission counter to 10 just like ARF [3].

$$\theta_U^{AARF} = \begin{cases} \min(2 * \theta_U^{AARF}, 50) & \text{if Up rate probing fails,} \\ 10 & \text{if rate downshift occures} \end{cases}$$

$$\theta_d^{AARF} = 2 \tag{2}$$

2.1.3 Adaptive Multi-Rate Retry (AMRR)

AMRR algorithm employs the multiple rate retry capabilities of the MadWifi driver based on ARF with an additional Binary Exponential Backoff (BEB) much like AARF. AMRR set c0 = c1 = c2 = c3 = 1, namely, each rate is tried just once. Then, r3 is always set as the lowest bit rate (i.e., 1 Mbps in 802.11b/g, and 6 Mbps in 802.11a), while r1 is the rate immediately lower than r0 , and r2 is the rate immediately lower than r1 .

To select r0, the AMRR algorithm employs the following simple heuristic: if less than 10% of the packet transmissions failed during the last observation period (and total frame transmissions are at least 10), then increase the transmission rate; otherwise, if more than 33% of the packet transmissions failed during the last period (and total frame transmissions are at least 10), then decrease the transmission rate. By default, an observation period is one second in AMRR [4].

```
R= [ r0 , r1 , r2 , r3 ]

Send MPDU at R=r0
check packet loss

if ( Packet_loss <= 10% ) then
    R--
if ( Packet_loss >= 30% ) then
    R++
```

Figure 1: AMRR algorithm operation

2.1.4 Atsushi Onoe Algorithm

This algorithm is well known because it has been used as the default rate control algorithm for the madwifi driver. I am not aware of any publication or reference about this algorithm beyond the madwifi source code [5].

ONOE algorithm is a variant of the AMRR scheme. Specifically, ONOE uses larger retransmission counts than AMRR (i.e., c0 = 4 and c1 = c2 = c3 = 2), while it sets r1, r2, r3 rates as AMRR. The major difference between the two schemes is that the ONOE algorithm associates a number of credits to the current rate r0. More precisely, the credit increase when less than 10% of packets require retransmission at a particular rate. It increased until it reaches a value of 10, then the algorithm switch to the next available higher rate, and the credit is reset to 0. if the retransmissions occurred for more than 10% of the packets sent in the last period, the rate will be switch the next lower rate and the process is restarted with a credit value of 0. Credit is determined using a relative long period of time because that it is less sensitive to individual packet loss than ARF and AMRR [5].

```
R= [ r0 , r1 , r2 , r3 ]
Send MPDU at R=r0
check packet loss

if ( Packet_loss >= 10% ) then
    R--
else if ( 10% of packet or more need to retry ) then
    credit --
else
    credit ++
end if

if ( credit >= 10 ) then
   R++
else if ( credit < 10 ) then
   R ( not change )
end if
```

Figure 2 : Onoe algorithm operation

2.1.5 SampleRate

SampleRate sends data at the highest possible bit-rate and stops using a bit-rate if it experiences four successive failures, so when first sending packets over a link SampleRate will decrease the bit rate until it finds a bit-rate that is capable of sending packets. The algorithm keeps track of previously transmitted frames in a table for each rate for each station. Each rate in the table contains the number of attempted transmissions, the number of successive failed transmissions, the number of ACK'ed frames, the total transmission time, average transmission time and lossless transmission time. This helps the algorithm select the rate which gives the maximum throughput even if this is not the highest rate available.

To calculate each bit-rate's average transmission time, SampleRate uses feedback from the wireless card to calculate how much time each packet transmission required. For each transmitted packet, 802.11 wireless cards indicate whether the packet was successfully acknowledged and how many times the card sent the packet. To avoid using stale information, SampleRate only calculates the average transmission time over packets that were sent within the last averaging window. This averaging window is set to ten seconds that the best bit-rate changes little over ten second periods.

$$TX - Time = backof\ f(r+1) + (r+1)(\Delta + L * \frac{8}{b}) \quad (3)$$

Where, r is number of retransmissions needed to successfully transmit a frame of size L (in bytes) with bit rate b. backof f (r+1) expresses the average back off delay introduced after r retries, and Δ accounts for the fixed MAC overheads. The MadWifi implementation of SampleRate adopts a multiple rate retry strategy similar to ONOE [6].

Table 1: Multi Rate Parameter [18]

RAA	Tx Rate	count
AMRR	r0	1
	r1 = r0 - 1	1
	r2 = r0 - 2	1
	r3 = lowest rate	1
ONOE	r0	4
	r1 = r0 - 1	2
	r2 = r0 - 2	2
	r3 = lowest rate	2
SampleRate	r0	2
	r1 = r0 a	3
	r2 = lowest rate	3
	r3 = 0	0

2.1.6 Minstrel

It has many similarities to SampleRate, but has two different primary metrics. While SampleRate selects its best rate based on the transmission time of each frame, Minstrel selects its bit-rate based on which rate can reach maximum throughput taking into account the expected number of retransmissions, based on statistical history of the wireless channel.

Minstrel, like SampleRate, probes the wireless channel in order to get an overview of its condition. Minstrel defines these probing packets as Look a round packets and spends a set amount of time probing bit rates other than the current best. The bit rates selected for the probing frames are selected more intelligently than in SampleRate. Minstrel, for example, does not sample rates that cannot possibly provide a better throughput than the current best [7].

2.2 Closed-loop Based Approaches

Use the timely collected SNR value to select an appropriate bit rate through looking up a predefined SNR-rate table. Compared with frame- based ones, the SNR-based ones react more quickly to the changes of link status. However, it is difficult to obtain the accurate SNR values, and capture the exact SNR-BER relationship in different propagation environments, especially in mobile environments. RBAR, CARA and RRAA use adaptive RTS/CTS exchanges belong to this category [16][17].

2.2.1 Collision-Aware Rate Adaptation (CARA)

Without using RTS/CTS frames under good conditions CARA designed to handle collisions. Except for the use of RTS/CTS, CARA adjusts the data rate similarly to ARF. Use RTS/CTS to reduce collisions from hidden terminals. To minimize the overhead from the use of RTS/CTS, CARA suggests that a transmitting station switches its adapter to sense the channel immediately after a transmission is over.

If the channel is sensed busy and the transmission gets lost, this loss is obviously inferred from collision without the need of an RTS. It should be noted that the busy channel sensed at the source station does not necessarily result in collision at the destination station [9]. Figure 3 the algorithm of CARA where, m: success count, n: failure count, RTS_{th}: RTS Threshold, M_{th}: success threshold, N_{th}: failure threshold and p_{th}: probe activation threshold.

```
m=n=o
R= arry of avalibal rate in standard
if ( MPDU ready to send )  then
    if ( SIZE(MPDU) >= RTSth ) || n > pth then
        Send RTS probe
    else if
            Send MPDU at R=r
end if
if ( MPDU sent successfully ) then
    m ++
    n = 0
    if ( m == Mth ) then
       if ( r < R_Max ) {
           r ++
       }
      m = 0
else if ( MPDU sent Fail ) then
    n ++
    m = 0
    if ( n >= Nth ) then
       if ( r > R_min ) {
           r--
       }
      n = 0
end if
```

Figure 3 : CARA algorithm operation

2.2.2 Robust Rate Adaptation Algorithm (RRAA)

The algorithm utilizes short-term loss ratio to assess the channel and adapt the transmission bit-rate to the dynamic change of the radio-channel condition. It also leverages an (adaptive RTS) filter to prevent collision losses with small overhead caused by decreasing rate.

It quickly responds to the significant and obvious change of the channel condition and also responds properly in the hidden terminals. The algorithm work in three phases keeping track of the frame loss ratio within a short time window, rate Change to decide whether to change the rate based on the estimated loss ratio or not and adaptive RTS Filter to selectively turn on RTS/CTS exchange in order to suppress collision losses [10].

```
r =  ( highest rate at standard )
count=estimation window (r)
loop
   send frame at r
   update loss ratio at r
   count --
   if ( count == 0 ) then
       if ( loss ratio > Max Tolerable loss thr.) then
       r = ( next lower rate )
       else if ( loss ratio < Opportunistic Rate Increase thr.) then
       r = ( next higher rate )
       else
       r = r ( Don't change it )
       end if
       count = estimation window (r)
   end if
end loop
```

Figure 4: RRAA algorithm operation

2.2.3 Receiver Based Auto-Rate (RBAR)

It requires incompatible changes to the 802.11 MAC and PHY protocol. The interpretation of some MAC control frames is changed and each data frame must include a new header field. In RBAR the rate adaptation mechanism is in the receiver instead of in the sender, by using RTS/CTR. The sender and the receiver exchange a pair of RTS/CTS before transmitting data. The receiver of the RTS frame calculates the highest bit-rate that would achieve less Bit Error Rate (BER). Then the receiver piggybacks the calculated bit-rate on the CTS frame. Then, the sender uses the bit-rate informed by the receiver [11]. Finally, table 2 summarize the rate adaption algorithms based on the above detailed description of them [1]-[11]

Table 2: Summary of Rate Adaption Algorithm Characteristics

ALGORIT-HM	RTS /CTS	LAY-ER	METRIC S	RATE ADJUST
ARF	NO	MAC	Probe Pkt	UP/DOWN
AARF	NO	MAC	Probe Pkt	UP/DOWN
AMRR	NO	MAC	Probe Pkt	UP/DOWN
Onoe	NO	MAC	Probe Pkt	UP/DOWN
SampleRate	NO	MAC	Probe Pkt	BEST RATE
Minstrel	NO	MAC	Probe Pkt	UP/DOWN
CARA	YES	MAC	SNR	UP/DOWN
RRAA	YES	MAC	SNR	UP/DOWN
RBAR	YES	PHY	SNR	BEST RATE

3. PERFORMANCE EVALUATION

3.1 Experiment Setting

Experiments were carried out with IEEE 802.11a MAC on simulator ns-3 [23] to evaluate the performance of six rate adaption algorithms. Three schemes were chosen form Open-Loop based Approaches: (i) AARF; (ii) Minstrel, and (iii) Onoe, and from Closed-Loop (iv) Cara, (v) RRAA and (vi) Ideal (it is Closed-Loop approach and similar to RBAR). Algorithms chosen based in compatibility with IEEE 802.11 protocol and implemented in wireless drive. In all experiments the simulation runs for 200 second.

Network topology considers 16 clients Ad-Hoc network configuration as shown in the figure 1. Ad hoc networks are dynamic and distributed entities with no centralized controller where nodes need to adapt their transmission parameter depending on the channel status and network dynamics e.g. variation in node density, traffic, and mobility. Ad-Hoc network are chosen to measure the performance of algorithms under uncontrolled and variation channel conditions.

The throughput of each algorithm is measured in two cases: a static network and mobile network environments and report as average results. Average Throughput = (Total Packet Transmitted / Total Simulation Time) Mbps (4)

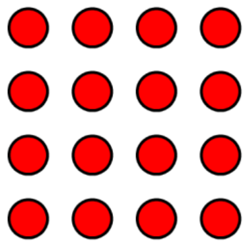

Figure 5: Network topology

3.2 Experiment Simulation

3.2.1 Static Network Performance

In this experiment, the nodes send a constant UDP traffic to its neighbor for ranges of transmissions distances: 30, 60, 90 and 120. Data was generated by a 10Mbps rate.

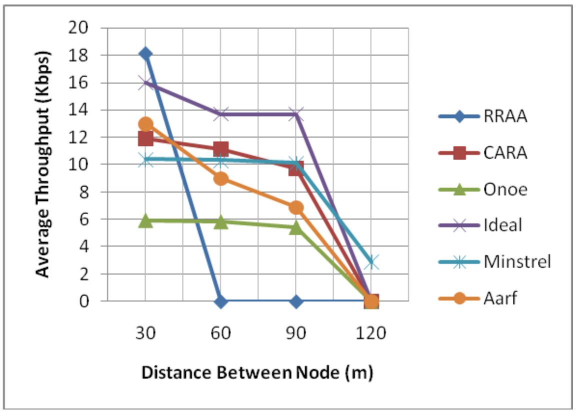

Figure 6: Throughput comparison under static environment

In figure 2 as expected with increase in transmissions distance average throughput decrease generally. All algorithms nearly have steady throughput at 60 m transmission distance, but it descends heavily upon arrival at the point 90 m as maximum allowable transmission distance. RRAA throughput descends heavily upon arrival at the point 40 m. IEEE 802.11 b only support range from (35-120) m as transmission distance [1], so all algorithms give zero throughput at the end of range expect Minstrel give a few Kilometers throughput.

3.2.2 Mobile Network Performance

The impact of number of host in the network measured for each algorithm as shown in the figure 3.

As the number of node increase RRAA get highest throughput and remain constant after 25 clients attach the network. The experiment can also measure the effect of collision in algorithms performance.

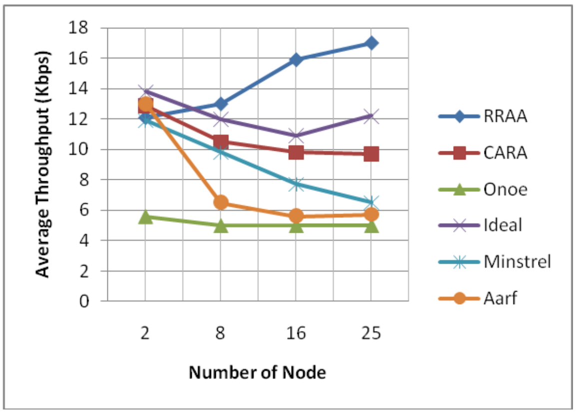

Figure 7: Throughput comparison of rate adaption algorithms with various number of contending node.

The performance of the algorithms with different nodes speed: 2, 4, 6, 8 and 10 m/s with no pause of node movement to get fast channel changing condition measured. In figure 4, all algorithms nearly are a constant with increasing of the nodes speed. RRAA still give the highest throughput due to its Adaptive-RTS filter.

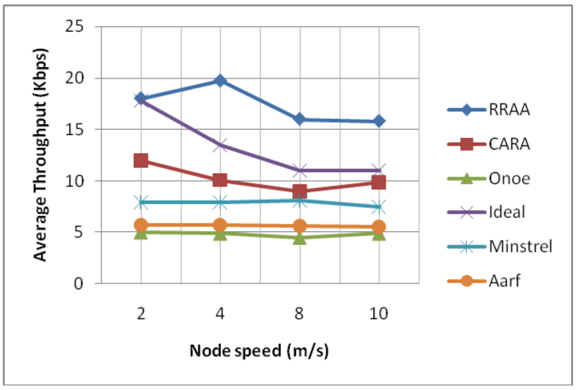

Figure 8 : Performance comparison for single CBR connection in multi-point network

Performance of algorithms measured at different type of traffic and packet size. The experiment runs at different ranges of packet size: 500, 1000, 1500, 2000 and 2500. The result showed that as packet size increase the average throughput increase also. RTS/CTS used at packet size equal 2500 bytes and at all experiment RTS/CTS leave as it defines in algorithm.

The throughput decrease at 2500 bytes with all algorithms due to uses RTS/CTS at each time of sending data. As result shows in figure 5 RRAA gained highest throughput due to its Adaptive-RTS filter.

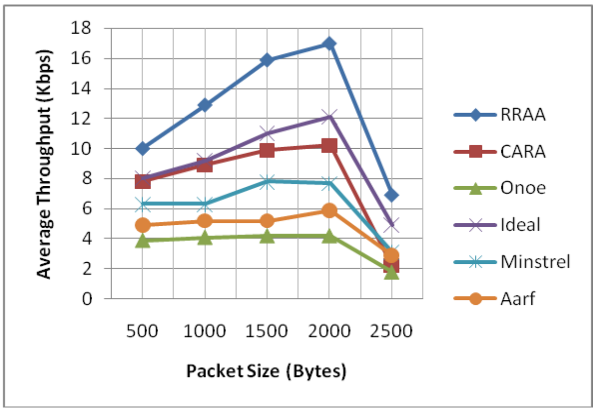

Figure 9 : Impact of the packet size on performance

4. CONCLUSIONS

In this paper, we have conducted an extensive comparative simulation study of several well-known rate adaptation algorithms using ns-3 simulator. Two categorize of the rate adaption algorithms, based on the collision consideration, are presented in this paper. The performance evaluation study, conducted in IEEE 802.11-based ad-hoc networks of most existing rate adaptation algorithms.

In our simulations, we did not consider the impact of traffic pattern on the performance of rate adaptation algorithms, which leaves as our future work. The results presented in the thesis demonstrate that research on rate adaptation is challenging and far from being completed, especially in recent extremely complicated communication environments. For example, more and more mobile equipments need to access the networks, which will lead to more congested wireless bands and more collisions.

RRAA outperforms four representative schemes AARF, Ideal, Onoe and Cara. RRAA still needs more development to make it effective under most network and traffic conditions. If RRAA developed to support RTS/CTS protocol under certain condition not depend only on number of successful or failure transmission, but also on power so we can improve the performance of it in large transmission area . And try to make packet size as factor on adjust rate.

5. REFERENCES

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

[2] Kamerman, Ad, and Leo Monteban. "WaveLAN®-II: a high-performance wireless LAN for the unlicensed band." Bell Labs technical journal 2, no. 3 (1997): 118-133.

[3] Lacage, Mathieu, Mohammad Hossein Manshaei, and Thierry Turletti. "IEEE 802.11 rate adaptation: a practical approach." In Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems, pp. 126-134. ACM, 2004.

[4] Xia, Qiuyan, Mounir Hamdi, and Tsz Ho Chan. "Practical Rate Adaptation for IEEE 802.11 WLANs." In GLOBECOM. 2006.

[5] https://madwifiproject.org/wiki/UserDocs/RateControl [Last cited on 2013 December 25].

[6] Bicket, John Charles. "Bit-rate selection in wireless networks." PhD diss., Massachusetts Institute of Technology, 2005.

[7] Description of the Minstrel algorithm. http://madwifi project.org/browser/madwifi/trunk/ath_rate/minstrel/ minstrel.txt [Last cited on 2014 March 9].

[8] Xia, Dong, Jonathan Hart, and Qiang Fu. "Evaluation of the Minstrel rate adaptation algorithm in IEEE 802.11 g WLANs." In Communications (ICC), 2013 IEEE International Conference on, pp. 2223-2228. IEEE, 2013.

[9] Kim, Jongseok, Seongkwan Kim, Sunghyun Choi, and Daji Qiao. "CARA: Collision-Aware Rate Adaptation for IEEE 802.11 WLANs." In INFOCOM, vol. 6, pp. 1-11. 2006.

[10] Wong, Starsky HY, Hao Yang, Songwu Lu, and Vaduvur Bharghavan. "Robust rate adaptation for 802.11 wireless networks." In Proceedings of the 12th annual international conference on Mobile computing and networking, pp. 146-157. ACM, 2006.

[11] Holland, Gavin, Nitin Vaidya, and Paramvir Bahl. "A rate-adaptive MAC protocol for multi-hop wireless networks." In Proceedings of the 7th annual international conference on Mobile computing and networking, pp. 236-251. ACM, 2001.

[12] Son, Giyeong. "Experimental Performance Evaluation of Bit-Rate Selection Algorithms in Multi-Vehicular Networks." (2011)..

[13] Vutukuru, Mythili, Hari Balakrishnan, and Kyle Jamieson. "Cross-layer wireless bit rate adaptation." ACM SIGCOMM Computer Communication Review 39, no. 4 (2009): 3-14.

[14] Gangwal, Prateek. "Implementation and experimental study of rate adaptation algorithms in IEEE 802.11 wireless networks." (2009).

[15]Biaz, Saad, and Shaoen Wu. "Rate adaptation algorithms for IEEE 802.11 networks: A survey and comparison." In Computers and Communications, 2008. ISCC 2008. IEEE Symposium on, pp. 130-136. IEEE, 2008.

[16] del Prado Pavon, Javier, and Sunghyun Choi. "Link adaptation strategy for IEEE 802.11 WLAN via received signal strength measurement." In Communications, 2003. ICC'03. IEEE International Conference on, vol. 2, pp. 1108-1113. IEEE, 2003.

[17] Huang, T., Chen, H., Zhang, Z., & Cui, L. "EasiRA: A hybrid rate adaptation scheme for 802.11 mobile wireless access networks," In Wireless Communications and Networking Conference (WCNC), 2012, pp. 1520-1525.

[18] Koçi, N., & Marina, M. K, "Understanding the role of multi-rate retry mechanism for effective rate control in 802.11 wireless LANs", In IEEE local computer networks, 2009, pp. 305–308.

[19] S. Pal, S. R. Kundu, K. Basu, and S. K. Das, "IEEE 802.11 Rate Control Algorithms: Experimentation and Performance Evaluation in Infrastructure Mode," in Proc. PAM'06, Adelaide, Australia, Mar. 2006.

[20] Vitturi, Stefano, Lucia Seno, Federico Tramarin, and Matteo Bertocco. "On the rate adaptation techniques of IEEE 802.11 networks for industrial applications." Industrial Informatics, IEEE Transactions on 9, no. 1 (2013): 198-208.

[21] Tang, Suhua, Hiroyuki Yomo, Akio Hasegawa, Tatsuo Shibata, and Masayoshi Ohashi. "Joint Transmit Power Control and Rate Adaptation for Wireless LANs." Wireless Personal Communications 74, no. 2 (2014): 469-486.

[22] E. Ancillotti, R. Bruno, and M. Conti, Experimentation and performance evaluation of rate adaptation lgorithms in wireless mesh networks," in PE-WASUN 08: Proceedings of the 5th ACM symposium on erformance evaluation of wireless ad hoc, sensor, and biquitous networks. Vancouver, BC, Canada: ACM, 2008, pp. 7-14.

[23] Network Simulator - ns-3, available from: http://www.nsnam.org/ns-3-19/ [Last cited on 2013 December 25].