



He, J., Kaleshi, D., Munro, ATD., & McGeehan, JP. (2006). Modeling link adaptation algorithm for IEEE 802.11 wireless LAN networks. In *3rd International Symposium on Wireless Communication Systems, 2006 (ISWCS '06), Valencia* (pp. 500 - 504). Institute of Electrical and Electronics Engineers (IEEE).  
<https://doi.org/10.1109/ISWCS.2006.4362348>

Peer reviewed version

Link to published version (if available):  
[10.1109/ISWCS.2006.4362348](https://doi.org/10.1109/ISWCS.2006.4362348)

[Link to publication record in Explore Bristol Research](#)  
PDF-document

## University of Bristol - Explore Bristol Research

### General rights

This document is made available in accordance with publisher policies. Please cite only the published version using the reference above. Full terms of use are available:  
<http://www.bristol.ac.uk/red/research-policy/pure/user-guides/ebr-terms/>

# Modeling Link Adaptation Algorithm for IEEE 802.11 Wireless LAN Networks

Jianhua He, Dritan Kaleshi, Alistair Munro, Joe McGeehan

Department of Electrical & Electronic Engineering,  
University of Bristol, Bristol, UK, BS8 1UB, United Kingdom.  
Email: {Jianhua.He, Dritan.Kaleshi, Alistair.Munro, Joe.McGeehan}@bristol.ac.uk

**Abstract**—Link adaptation is a critical component of IEEE 802.11 systems. In this paper, we analytically model a retransmission based Auto Rate Fallback (ARF) link adaptation algorithm. Both packet collisions and packet corruptions are modeled with the algorithm. The models can provide insights into the dynamics of the link adaptation algorithms and configuration of algorithms parameters. It is also observed that when the competing number of stations is high, packet collisions can largely affect the performance of ARF and make ARF operate with the lowest data rate, even when no packet corruption occurs. This is in contrast to the existing assumption that packet collision will not affect the correct operation of ARF and can be ignored in the evaluation of ARF. The work presented in this paper can provide guidelines on configuring the link adaptation algorithms and designing new link adaptation algorithms for future high speed 802.11 systems.

## I. BACKGROUND

IEEE 802.11 wireless local area network (WLAN) physical layers (PHYs) support multiple transmission rates [1]. Today, three different physical (PHY) layers for the IEEE 802.11 WLAN are available (802.11a/b/g); they all provide multi-rate capabilities. 802.11n is under standardization process. It will provide higher data rates and more data rate options. In order to achieve the highest possible throughput, the transmission rate should be chosen in an adaptive manner since the wireless channel condition varies over time due to such factors as station mobility, time-varying interference, and location-dependent errors [7]. The PHY rate to be used for a particular frame transmission is solely determined by the transmitting station, either with or without assistance from the receiving station.

Due to the commercial reasons, only very few algorithms such as Auto Rate Fallback (ARF) or Receiver Based Auto Rate (RBAR) have been published, and the implementation challenges associated with these mechanisms have never been publicly discussed [7] [6]. The link adaptation algorithms can be classified into two categories, SNR based or packet retransmission (loss) based [2] [3] [4] [6] [7] [8] [9]. We will give a simple overview of the existing link adaptation algorithms in Section II. While rate adaptation algorithm is a critical component of 802.11 WLAN systems, to the best of our knowledge, the performance of the link adaptation algorithms have not been analyzed for multiple user 802.11 network scenarios.

In this paper, we analytically model a retransmissions based Auto Rate Fallback (ARF) link adaptation algorithm. Both packet collisions and packet corruptions are modeled with the algorithm. The models can provide insights into the dynamics of the link adaptation algorithms and configuration of algorithms parameters. In the left of the paper, we will introduce the existing link adaptation algorithms in Section II. Modeling of the ARF algorithm will be presented in Section III. Analytical results for ARF algorithm will be presented and discussed in Section IV. Section VII concludes the paper.

## II. LITERATURE REVIEW

As we stated previously, link adaptation algorithms can be classified into two categories, SNR based or packet retransmission based. In the SNR based link adaptation algorithms, received signal strength (RSS) is used as the indication of link quality. Then transmission rate is selected based on the average or instantaneous RSS information from a predetermined SNR-rate table. Normally, the RSS information is collected by the transmitting station from previous frames received from the receiving station [2] [3] [4]. Receiver based Rate Fallback (RBRF) is a typical example of such algorithms. But RBRF requires incompatible modification to the standard and RTS/CTS access mechanism must be enabled to measure the link quality. SNR based link adaptation algorithms can achieve good performance when the link quality does not change too frequently, the estimation of the SNR and the mapping of SNR to transmission rate are correct. However, it is argued that the link quality is hard to predict. And it is also difficult to design proper and adaptive SNR-rate table.

In the packet retransmission based link adaptation algorithm, the transmitting station counts the outcome (either successful or failed) of each transmission attempt. Based on the packet transmissions (losses) history, the transmitting rate can be adaptively raised by a level or fallback or be kept. Auto Rate Fallback (ARF) is the first documented bit-rate selection algorithm [7]. The ARF algorithm also performs well in situations where link conditions change on the order of tens of packets. It reacts particularly well to link degradation; within a few packets it can step down to the lowest bit-rate. Additionally, there are several other publicly available retransmission based link adaptation algorithms, including the Onoe algorithm, SampleRate [9] and AARF which are all

implemented in device driver for Atheros cards in Linux [11]. In this paper, we will focus on modeling the ARF as an example of retransmissions based algorithms. Modeling other link adaptation algorithms will be our future work.

#### A. Auto Rate Fallback (ARF)

As ARF is one of the two link adaptation algorithms we will model in this paper, we will give more introduction to this algorithm. ARF was developed for WaveLAN-II 802.11 cards, one of the earliest multi-rate 802.11 cards and could send at 1 and 2 megabits. ARF aims to adapt to changing conditions and take advantage of higher bit-rates when opportunities appear. ARF was also designed to work on future WaveLAN cards with more than 2 bit-rates. For a particular link, ARF keeps track of the current bit-rate as well as the number of successive transmissions without any retransmissions. Most 802.11 wireless cards offer feedback about packet transmission after the transmission has either been acknowledged or exceeded the number of retries without an acknowledgment. When the ARF algorithm starts for a new destination, it selects the initial bit-rate to be the highest possible bit-rate. Given the number of retries that a transmission used and whether or not it was successfully acknowledged, ARF adjusts the bit-rate for the destination based on the following criteria:

- 1) Set a timer.
- 2) If the packet was never acknowledged, move to the next lowest bit-rate and return to Step 1); else, continue.
- 3) If  $N_s$  (default value of  $N_s$  is 10) successive transmissions have occurred without any retransmissions, move to the next highest bit-rate and return to Step 1); else, continue.
- 4) If the timer expires, move to the next highest bit-rate and return to Step 1). Otherwise, continue at the current bit-rate and return to Sept 2).

#### B. IEEE 802.11

802.11 uses carrier sense multiple access with collision avoidance (CSMA/CA) to access channel [1]. Each node uses a back-off window to track how long it should defer sending packets after the medium has become idle. When a node wants to transmit a packet, it waits until the medium is idle for at least a Distributed Inter-Frame Gap (DIFS) and then picks a random time within its back-off window and waits until this time expires. If the medium has been idle during the entire back-off period, it sends the packet and resets the back-off window to the minimum value. Otherwise, it doubles the back-off window, waits until the medium is idle for at least a Distributed Inter-Frame Gap (DIFS) period of time, and begins the back-off period again.

### III. ANALYTICAL MODEL

In this section, we will present the analytical model for 802.11 distributed coordination function (DCF). The analytical model is extended from the general model proposed in [12]. Using this analytical approach, the impacts of rate adaptation and packet collision/corruption can also be taken into

account. Following the idea used in [14], we can also study the performance of 802.11 backoff parameter based service differentiation.

#### A. Model Assumption

We assume a single hop wireless LAN where  $N$  stations are identical to each other. Each station has saturated traffic to transmit to one of its neighbors.

To facilitate the modeling work, we introduce a notion of virtual time slot. The concept of virtual time slot has been used in [14]. We assume that system time is slotted with each time slot of  $\delta$  second. There are two general independent events for the backoff process of a node. The first event is called transmission event, which starts from the time of transmissions from a tagged node or nodes within its carrier sense area and ends after the channel is continually sensed idle for DIFS period. In this event, the transmissions may be asynchronous and may fail independently. The inactive nodes freeze their backoff counters. The second event is called active backoff event, during which the backoff counter is active and will decrement one in the corresponding system time slot. The occurrence of these two events depends on the transmission activities of the tagged nodes and nodes within its carrier sense area. To simplify analysis, the two events are generalized to a virtual backoff event. The period of the generalized event is called virtual time slot. Denoted  $T_{avg}$  as the average duration of a virtual time slot. In the remaining of the paper, time slot will be refer to virtual time slot if not explicitly pointed out.

We assume that each node will transmit with probability of  $\tau$  independently to its neighbors at each virtual time slot, and  $\tau_r$  with transmission rate of  $R_r$  bps,  $r \in [1, N_t]$ . It is clear that  $\tau = \sum_{r=1}^{N_t} N_t \tau_r$ . Assume that data packet and ACK packet have fixed length of  $L_{dt}$  and  $L_{ack}$  bits. Transmissions of a data packet and an ACK packet at transmission rate of  $R_r$  will last for  $T_{dt}(r)$  and  $T_{ack}(r)$  time slots:  $T_{dt}(r) = L_{dt}/(R_r T_{avg})$  and  $T_{ack}(r) = L_{ack}/(R_r T_{avg})$ .

#### B. Model Analysis

Let  $p_{er}$  denote the average probability that an acknowledgement (ACK) packet can not be correctly received by a transmitting station from the receiving station for a data packet transmit attempt, and  $p$  denote the probability that a time slot is idle. Then from the algorithm of ARF and DCF, we can produce a Markov chain (shown in Figure 1) to model the dynamic backoff process of a general station transmitting to a fixed neighbor [12] [14].

In the Markov chain, states  $(i)$  are virtual Markov states, introduced to facilitate expression,  $i \in [1, N_t]$ . The states  $(i, j, k)$  are the real states. In a state  $(r, i, j)$ ,  $r$  represents the order of the transmission rate  $R_r$ ;  $i$  represents the backoff state, meaning that the backoff process is in either the first transmission ( $(i = 0)$ ) or retransmission attempt ( $i \in [1, m - 1]$  for the  $i$ th retransmission and  $m$  for the  $[m, \infty)$  retransmission);  $j$  represents the value of the backoff counter,  $j \in [0, W_i - 1]$ , where  $W_i$  is the contention window,  $W_i = 2^i W_0$ ,  $i \in [0, m]$ ,  $W_m$  is the maximum contention

window, and  $m$  is the maximum retransmission stage. After  $m$  retransmissions of a data packet, if the packet can not be successfully acknowledged, then it will be simply discarded.

With the above assumptions and definitions, we can calculate the transmission probability  $\tau$ . Denote  $P(b|a)$  the transition probability of a Markov state  $a$  to state  $b$ . Let  $P(r_1, i_1, j_1|r_0, i_0, j_0)$  denote the transition probability of a Markov state  $(r_0, i_0, j_0)$  to  $(r_1, i_1, j_1)$ ,  $P(j_1|j_0)$  denote the probability of Markov state  $(j_0)$  to  $(j_1)$ ,  $P(r_1, i_1, j_1|j_0)$  denote the probability of Markov state  $(j_0)$  to  $(r_1, i_1, j_1)$ ,  $P(j_1|r_0, i_0, j_0)$  denote the probability of Markov state  $(r_0, i_0, j_0)$  to  $(j_1)$ . Then we have the following formula for the probabilities of state transition described in the Markov chain:

$$\left\{ \begin{array}{ll} P\{r|r, 0, 0\} = p_{cr}, & r \in [1, N_t], \\ P\{r+1|r, 0, 0\} = p_{r(r+1)}, & r \in [1, N_t - 1], \\ P\{r-1, 1, j|r, 0, 0\} = p_{r(r-1)}, & r \in [2, N_t], \\ P\{1, 1, j|1, 0, 0\} = p_{i0}, & j \in [0, W_1 - 1] \\ P\{1, i, j|1, i-1, 0\} = p_{c1}/W_0, & i \in [2, m-1], \\ & j \in [0, W_i - 1] \\ P\{1|1, i, 0\} = p_{c1}, & i \in [1, m-1], \\ P\{1|1, m, 0\} = 1, & \\ P\{r, 0, j|r\} = 1/W_0, & r \in [1, N_t], \\ P\{r, i, j|r, i, j+1\} = 1-p, & r \in [1, N_t], \\ & i \in [0, m], \\ & j \in [0, W_i - 1] \\ P\{r, i, j|r, i, j\} = p, & r \in [1, N_t], \\ & i \in [0, m], \\ & j \in [0, W_i - 1] \\ P\{r-1, i+1, j|r, i, 0\} = p_{cr}/W_i, & r \in [2, N_t], \\ & i \in [2, m-1], \\ & j \in [0, W_i - 1], \\ P\{r-1|r, m, 0\} = p_{cr}, & r \in [2, N_t], \\ P\{r|r, m, 0\} = 1-p_{cr}, & r \in [2, N_t]. \end{array} \right. \quad (1)$$

$$\left\{ \begin{array}{ll} p_{cr} = 1 - (1-p)(1-p_{er}), & r \in [1, N_t], \\ p_{fr} = (1-p_{cr})^{N_s}, & r \in [1, N_t - 1], \\ p_{sr} = \sum_{i=1}^{N_t-1} (1-p_{cr})^i, & r \in [1, N_t - 1], \\ p_{br} = 1 - p_{sr} - p_{fr}, & r \in [1, N_t - 1], \\ p_{sr} = 1 - p_{cr}, & r = N_t, \\ p_{br} = p_{cr}, & r = N_t \end{array} \right. \quad (2)$$

Denote  $b_{rij}$ , as the distributions of states  $(r, i, j)$  states,  $r \in [1, N_t]$ ,  $i \in [0, m]$ ,  $j \in [0, W_i - 1]$ . Then we can calculate  $b_{rij}$  using the state transition probabilities (1) and the following condition (3),

$$\sum_{r=1}^{N_t} \sum_{i=0}^m \sum_{j=0}^{W_i-1} b_{rij} = 1 \quad (3)$$

Transmission probability  $\tau$  and  $\tau_r$  can be approximated by  $b_{ri0}$ ,

$$\left\{ \begin{array}{l} \tau = \sum_{r=1}^{N_t} \sum_{i=0}^m b_{ri0}, \\ \tau_r = \sum_{i=0}^m b_{ri0} \end{array} \right. \quad (4)$$

Under the assumption that the channel states sensed by the neighbors of a node is the same as that sensed by the node, we can calculate  $p$  as,

$$p = 1 - (1 - \tau)^{N-1} \quad (5)$$

where  $N$  is the number of nodes in the network.

It is clear that  $p$  is a function of transmission probability  $\tau$ . Given  $p_{er}$ , for  $r \in [1, N_t]$ , with only two unknowns  $\tau$  and  $p$ , numerical methods can then be used to calculate them from (4) and (5).

### C. Throughput Calculation

Denote  $S$  the single node throughput, defined as number of data bits successfully transmitted by a node during a second. It can be calculated as the ratio of payload information successfully transmitted by a node in a time slot to the length of a time slot. Let  $p_{idle}$  be the probability that the channel is sensed idle by the tagged node in a time slot. Let  $p_{err}(r)$  denote the probability of an event (denoted by  $E_{err}(r)$ ) that an acknowledgement (ACK) packet for a data packet transmission is not correctly received in a time slot with the lowest data rate used for the packet(s) transmitted in the time slot being  $R_r$ . The reason An ACK packet is not received for a data packet maybe that collision happens or neither of data and ACK packets are not successfully decoded. Let  $p_{suc}(r)$  denote the probability of an event (denoted by  $E_{suc}(r)$ ) that an ACK packet for a packet transmission is successfully received in a time slot, with the lowest data rate used for the packet(s) transmitted in the time slot being  $R_r$ . Let  $T_{err}(r)$  ( $T_{suc}(r)$ ) denote the average duration of a time slot during which single event  $E_{err}(r)$  ( $E_{suc}(r)$ ) happens. Then  $T_{err}(r)$  and  $T_{suc}(r)$  can be expressed for the basic access scheme as below:

$$\left\{ \begin{array}{l} T_{suc}(r) = DIFS + SIFS + (L_d + L_a)/R_r \\ T_{err}(r) = DIFS + L_d/R_r + EIFS \end{array} \right. \quad (6)$$

For RTS/CTS access scheme,

$$\left\{ \begin{array}{l} T_{suc}(r) = DIFS + 3SIFS + (L_r + L_c + L_d + L_a)/R_r \\ T_{err}(r) = DIFS + L_r/R_r + EIFS \end{array} \right. \quad (7)$$

$SIFS$  and  $EIFS$  are shortest inter-frame space and extended inter-frame space, defined by IEEE 802.11 standard for different types of channel access.

$$\left\{ \begin{array}{l} p_{idle} = (1 - \tau)^{N-1}, \\ p_{suc}(r) = N\tau_r(1 - p_{er})(1 - \sum_{j=1}^r \tau_r)^{(N-1)}, \\ p_{err}(r) = \sum_{i=2}^N (1 - \sum_{j=1}^i \tau_r)^i + N\tau_r p_{er}(1 - \sum_{j=1}^r \tau_r)^{(N-1)} + \\ N\tau_r(1 - p_{er})[(1 - \sum_{j=1}^r \tau_r)^{(N-1)} - (1 - \tau_r)^{(N-1)}] \end{array} \right. \quad (8)$$

Then the average duration of a time slot  $T_{avg}$  and single node throughput  $S$  can be calculated by (9):

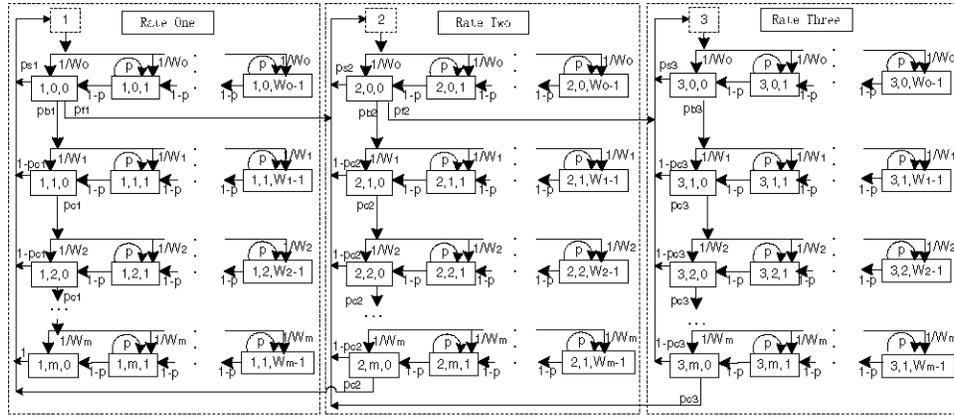


Fig. 1. Markov chain for the backoff procedure of a station operating with 3 transmission rates.

$$\begin{cases} T_{avg} = p_{idle}\delta + \sum_{r=1}^{N_t} [p_{err}(r)T_{err}(r) + p_{suc}(r)T_{suc}(r)] \\ S = \frac{L_d \sum_{r=1}^{N_t} \tau_r (1-p_{err}(r))}{T_{avg}} \end{cases} \quad (9)$$

We can use the same procedures to calculate the single node throughput for RTS/CTS access scheme and 802.11 based service differentiation schemes.

#### IV. ANALYTICAL RESULTS FOR MULTI-RATE AND SINGLE SNR NETWORK SCENARIOS

In this Section, we present some typical analytical throughput results of ARF, with different channel conditions (to be included soon). We consider the one-hop wireless LAN, which has 5, 15, 25, 35 wireless stations. All the wireless stations are identical, with saturated traffic and operate with ARF algorithm. Presented in Fig.2 and Fig.3 are the single node throughput for ARF with 3 and 8 optional rates respectively, with RTS/CTS enabled. In each figure, we present the results for 2, 5 and 10 successive transmission which is required to increase a rate to a higher level in the ARF algorithm. The 3 rates set is [6 12 18] Mbps, and the 8 rates set is [6 8 12 18 24 36 48 54] Mbps. In the current analytical results, the SNR is set high enough to achieve  $P_{er} = 0$ . Therefore, we can completely study the impact of packet collision and traffic load on the performance of ARF algorithm.

Initially, ARF is designed to adapt to the link quality. When the link quality is unchanged (here SNR is very high), ARF should be stabilized and operate with the highest possible transmission rate. From the analytical results, we observed that when there are low to medium number of competing stations, ARF can always adapt to achieve the highest possible throughput. In this case, packet collisions does not affect the performance of ARF. However, when the number of competing stations is increased to 35, the packet collision probability increase to about 0.4 and ARF is significantly affected when the number of required successive transmissions is large (5 and 10 in Fig.2 and Fig.3). The stabilized transmission rate is around the one with the lowest throughput (the aggregate

network throughput is 6 Mbps), instead of that with the highest throughput (the network throughput should be 10 and 35 Mbps respectively). The results are not expected and not desired. Therefore, ARF should be reconsidered to cope with high number of stations. In the next step, we will analyze the performance of ARF with the network scenarios of  $P_{er} > 0$  and consider the improvement on the ARF algorithm to deal with well both packet collision and corruption.

#### V. CONCLUSIONS

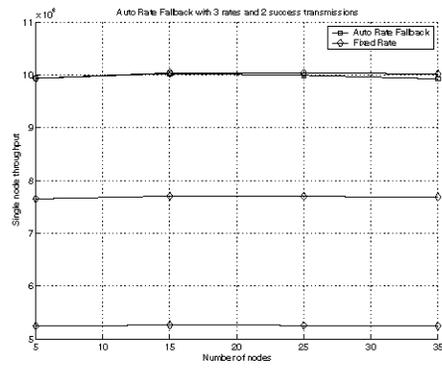
In this paper, we modeled the Auto Rate Fallback (ARF) rate adaptation algorithm. The model is efficient and scalable. It can be used to thoroughly investigate the behavior of ARF algorithm and adaptively configure the operation parameters of the algorithm under various network and communications environments.

#### ACKNOWLEDGMENT

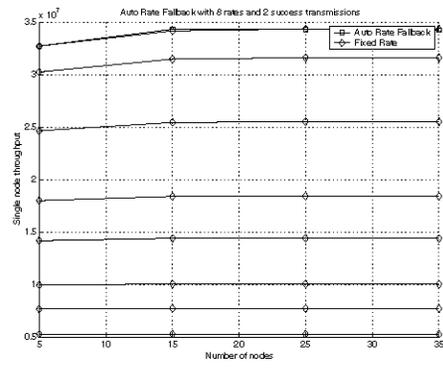
The work reported in this paper is part of the OSIRIS project within the 3C Reserach programme on convergent technology for digital media processing and communications, whose funding and support is gratefully acknowledged. For more information, please visit [www.3cresearch.co.uk](http://www.3cresearch.co.uk).

#### REFERENCES

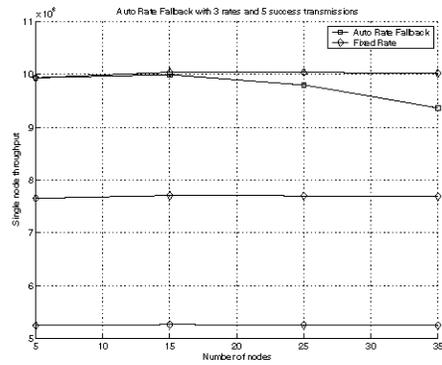
- [1] IEEE. IEEE standard for wireless lan medium access control (MAC) and physical layer (PHY) specifications. November 1997.
- [2] J. Pavon, S. Choi, Link Adaptation Strategy for IEEE 802.11 WLAN via Received Signal Strength Measurement, *Proc. of IEEE International Conference on Communications*, 2003.
- [3] Daji Qiao and Sunghyun Choi, Goodput Enhancement of IEEE 802.11a Wireless LAN via Link Adaptation, *in Proc. IEEE International conference on Communications*, pp. 1995-2000, June 2001.
- [4] Daji Qiao, Sunghyun Choi, and Kang G. Shin, Goodput Analysis and Link Adaptation for IEEE 802.11a Wireless LANs. *IEEE Trans. on Mobile Computing*, vol. 1, no. 4, pp. 278-292, October-December 2002.
- [5] J. Wu, Huihui-Huan Liu and Yi-Jen Lung, An Adaptive Multirate IEEE 802.11 Wireless LAN. *in Proc. 15th International Conference on Information Networking*, pp. 411-418, 2001
- [6] G. Holland, N. Vaidya, and P. Bahl. A rate-adaptive MAC protocol for multi-hop wireless networks. *In Proc. ACM/IEEE MobiCom*, July 2001
- [7] A. Kamerman and L. Monteban. WaveLAN-II: A high-performance wireless lan for the unlicensed band. *ATT Bell Laboratories Technical Journal*, pages 118-133, 1997.



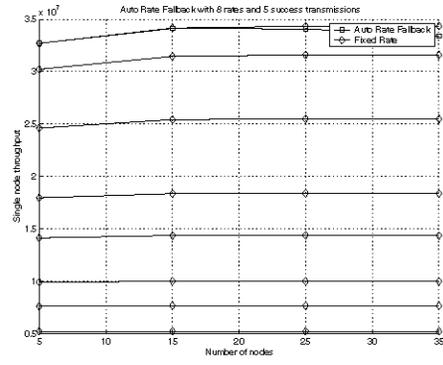
(a) 2 successive transmissions



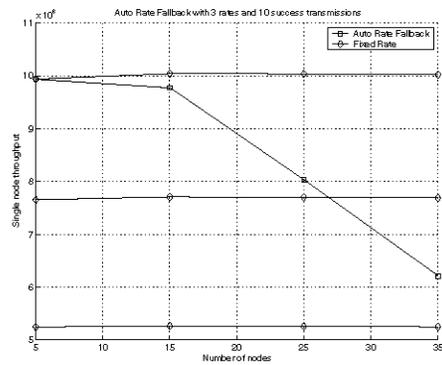
(a) 2 successive transmissions



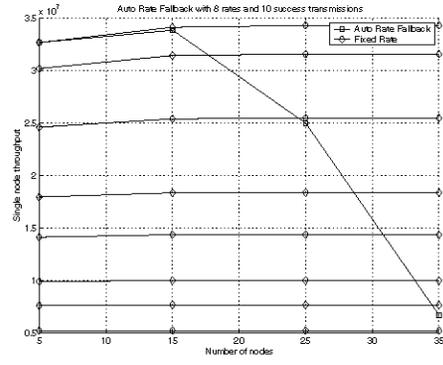
(b) 5 successive transmissions



(b) 5 successive transmissions



(c) 10 successive transmissions



(c) 10 successive transmissions

Fig. 2. Aggregate network throughput of auto rate fallback algorithm with 3 rates and 2, 5, 10 successive transmissions fallback.

Fig. 3. Aggregate network throughput of auto rate fallback algorithm with 8 rates and 2, 5, 10 successive transmissions fallback.

[8] Mathieu Lacage, Hossein Manshahi, and Thierry Turletti. IEEE 802.11 rate adaptation: A practical approach. *ACM conference MSWiM'04*, 2004.

[9] John C. Bicket, Bit-rate Selection in Wireless Networks. *Thesis of Master*, MIT, February 2005.

[10] Eddie Kohler, Robert Morris, Benjie Chen, John Jannotti, and M. Frans Kaashoek. The Click modular router. *ACM Transactions on Computer Systems*, 18(4), November 2000.

[11] Madwifi. <http://sourceforge.net/projects/madwifi>.

[12] G. Bianchi. Performance analysis of the IEEE 802.11 distributed coordination function. *IEEE JSAC*, 18:535–547, March 2000.

[13] J. He, L. Zheng, Z. Yang, and C. Chou. Performance analysis and service differentiation in IEEE 802.11e EDCA. In *Proc. of LCN*, pages 184–190, October 2003.

[14] J. He, L. Zheng, Z. Yang, and C. Chou. Investigation of service differentiation mechanisms for IEEE 802.11 wireless LAN. *IEICE Trans. on Commun.*, June 2004.