Design and Experimentation of Rate Adaptation for IEEE 802.11n WLANs

Arafet Ben Makhlouf and Mounir Hamdi
Department of Computer Science and Engineering
The Hong Kong University of Science and Technology
Clear Water Bay, Kowloon, Hong Kong
Email: {ninoara, hamdi}@cse.ust.hk

Abstract— **Wireless Local Area Networks (WLANs) have become increasingly popular due to the recent availability of affordable devices providing multiple and high rate capabilities. Optimizing the performance of WLANs for emerging new internet applications that demand High Throughput (HT) is an important and a highly challenging issue. In this paper, we present a novel practical Rate Adaptation (RA) algorithm, termed L3S, which extends legacy schemes with new MIMO features suitable for the forthcoming 802.11n high-speed MIMO-based WLAN products. We implemented our rate adaptation algorithm in real hardware devices and evaluated its performance and compared it to the existing rate control mechanisms. Our experiments will demonstrate that our scheme allows the current 802.11n devices to have a greater adaptability to a variety of wireless channel conditions, and performs better than state-of-the-art rate adaptation algorithms.**

Keywords— **Rate Adaptation, IEEE 802.11n, MIMO, Ath9k, Implementation, Experimentation.**

I. INTRODUCTION

In the next generation WLAN standard, termed IEEE 802.11n [1], new Physical (PHY) and Medium Access Control (MAC) layer enhancements have been introduced. These improvements have given birth to high data rates to keep up with current and upcoming Internet applications. The fundamental problem is that 802.11 wireless channel suffers from time-varying losses, due to mobility, interference and contention from hidden stations, leading to poor and inconsistent throughput performance. In particular, rate adaptation (RA) is a fundamental resource management issue for 802.11 devices; its goal is to optimize the wireless link throughput in various environments. In addition, all the IEEE 802.11 standards do not specify any algorithm for automatic rate control.[1]

The wide variety of data rates provided in 802.11n high speed WLANs demands an efficient control scheme for optimal rate selection and also fast rate adaptation based on time-varying channel conditions. In the literature, there are some RA algorithms designed for this recent standard. But, to the best of our knowledge, many of them are implemented and tested only by using the network simulator NS-2. Moreover, these proposed RA algorithms have not been implemented in practice using existing device drivers so far because they require modifications to the IEEE 802.11 standard.

For WLANs, the existing RA algorithms are simple and have intuitive rules to find the best rate, but they are very slow and theirs performances degrade in highly variable channels. Then, we need a compatible 802.11 standard and a robust algorithm against various dynamics that solves the stability and the responsiveness simultaneously.

In this paper, we design a practical rate control algorithm for 802.11n WLANs, based on a probing system that guarantees that it is has **L**ong-Term **S**tability and **S**hort-term **S**tability (L3S). We then implement it in commercial devices using the actual Ath9k driver without modifications to the existing standard. The new rate adaptation classifies transient and sustained changes in the link conditions. Then, it controls both short-term and long-term channel quality variations respectively by continuously monitoring transmission history and intelligently probing at new data rates that may outperform the current rate. Our proposed rate control algorithm adapts rapidly to these changes by adjusting the efficient transmission rate. Thus, it optimizes the throughput (or delay) performance on a wireless link.

The rest of the paper is organized as follows. In Section II, we briefly introduce the IEEE 802.11n standard and related work. The basic ideas and the practical implementation of the proposed algorithm, L3S, using the actual Ath9k driver are presented in the section III in detail. Section IV gives an extensive

[1] This research is sponsored by HKUST Grant RPC10EG21.

experiment evaluation of L3S compared to the current state-of-the-art existing MIMO rate control scheme used in 802.11n Atheros chipsets. Finally, the Section V concludes the paper.

II. BACKGROUND AND RELATED WORK

A. *IEEE 802.11n Standard*

The IEEE 802.11n [1] incorporates new features to boost performance of WLANs. It makes a number of changes to the PHY layer to increase the effective throughput.

Multiple-Input Multiple-Output (MIMO): Based on OFDM technology [2], the 802.11n devices are able in simultaneously transmitting and/or receiving data through multiple antennas. Data signals are split into multiple parts. Each independent signal, called a spatial stream, is sent from a transmit antenna.

Guard Interval (GI): It is a period of time added between adjacent OFDM symbols to minimize Inter Symbol Interference (ISI) caused by the effects of multipath. The legacy 802.11a/b/g standards use only a GI of 800ns, but 802.11n can operate with 800ns and 400ns.

Channel Bandwidth: Legacy 802.11 systems use a radio channel spacing that is approximately 20MHz wide, while 802.11n can operate in the 20MHz and 40MHz modes. Over two adjacent channels bonded together, the 40MHz operation doubles the data rate.

The 802.11n provides in total 78 Modulation Coding Schemes (MCSs) or transmission rates that vary from 6.5Mbps to 600Mbps at the maximum. Using these high transmission rates lead to unwanted overhead caused by inter-frame gaps, header and acknowledgement (ACK) of each transmitted frame. For this reason, 802.11n standard addresses these issues by making also changes in the MAC layer to increase the amount of transmission time. The IEEE 802.11n MAC enhancements are principally based on the reduction of overhead by including a Reduced Inter-Frame Space and aggregation mechanisms. The main idea of aggregation is to bundle multiple frames or ACKs together for a single transmission.

B. *Rate Adaptation Algorithms*

This section gives a brief description of the various RA algorithms that have been proposed in the recent literature. Based on the Channel State Information (CSI) used for channel quality estimation, the RA algorithms can be grouped in two classes as follows. The first class contains statistic-based algorithms (e.g., ARF [3], AARF [4], SampleRate [5], and ONOE [6]) and the second class comprises signal-measurement based algorithms (e.g., CHARM [7], RBAR [8], OAR [9], and Goodput Analysis [10]). Table I shows a classification of some related work. In [11] a hybrid approach was proposed, which uses the collected transmission statistics and the measured Received Signal Strength Information (RSSI) to improve the performance of network multimedia applications. Performance evaluations of some RA algorithms are available in [12, 13, and 14].

The legacy 802.11 standards support only open-loop RA, while the 802.11n standard supports both open-loop and closed-loop rate adaptation schemes to optimize the performance on a point to point link. The open-loop schemes are based on implicit feed-backs by monitoring the ACK packets to estimate the link quality between the sender and a specific receiver. On the other hand, the closed-loop schemes are based on explicit feedbacks that are carried in control packets. In that case, the receiver provides instantaneously its estimated CSI or suitable MCS to help the sender to choose the optimal MCS that satisfies both sides.

Table I

Statistic-based Vs. Signal-measurement-based Approaches

Statistic-based Approaches	**Signal-measurement-based Approaches**
ARF, AARF, ONOE, SampleRate	CHARM, RBAR, OAR, Goodput Analysis
collect transmission statistics	measure the signal strength
not require RTS/CTS	may require RTS/CTS
no changes to standard	need changes to the standard
performance degradation in many cases	good performance (if neglect the overhead)

C. *MIMO Atheros Chipsets and Ath9k Driver*

Ath9k [15] is a completely free and open-source Linux device driver for Atheros chipsets. It is designed to operate with 802.11n WLAN cards. The current Ath9k implements most, but not all functionalities of the 802.11n standard. It can operate on both 2.4GHz and 5GHz bands with both 20MHz and 40MHz modes. Also, it supports up to two spatial streams. Then, the data rates go up to 135Mbps with 20MHz and they go to 270Mbps with 40MHz. The IEEE 802.11n standard introduces MCS feedback mechanism, but the current driver does not implement this feature.

Ath9k driver maintains a number of FIFO (First-In First-Out) queues of transmission descriptors with many detailed control information. Whenever a data is received from the network layer, the driver adds the appropriate IEEE 802.11 MAC header with the necessary information and inserts the packet into one of the FIFO queues. According to the transmission descriptor parameters, the PHY layer is responsible for sending the packet over the wireless medium.

In Ath9k, Atheros Communications specifies four retry series (r_0/c_0, r_1/c_1, r_2/c_2, r_3/c_3) for every frame ready to be sent, each pair with a specific rate r_i and its maximum transmission attempts c_i. Therefore, the lost packet is transmitted at four different decreased rates until a success or exceeding the retry limit. Every frame is discarded after $c_0 + c_1 + c_2 + c_3$ unsuccessful transmission attempts. In addition, the Ath9k driver uses ONOE rate adaptation algorithm, as mentioned in [18], which sends a constant small fraction (10%) of the data packets at the adjacent rates of the current rate. The throughput calculation is based on the measured PER that is periodically updated. Finally, based on the status field, the descriptor is updated after the frame has been transmitted successfully or discarded.

III. OUR ALGORITHM DESIGN

In this paper, we focus on designing a practical transmitter-based algorithm without link knowledge at the sender, which is self-tuning and fast responsive over the time-varying of wireless channel conditions. In particular, our mechanism is an open-loop RA algorithm for MIMO WLANs that does not require any explicit feedback from the receiver; instead it monitors only the binary ACK (implicit feedback). Our algorithm, termed L3S, is based on the following observations:

- The wireless interfaces provide the signal strength measurements to the upper layer. In practice, they are un-calibrated and only for very limited area. Moreover, there is no explicit method to exchange the RSSI between the stations. For these reasons, the RSSI measurements cannot be helpful for stability and responsiveness of the algorithm.
- The static nature of some previously proposed RA algorithms with static thresholds and timers (e.g., ARF, ONOE, and etc) are application dependent, and generally fixed at the design time. For this reason, they are less versatile to very dynamic link conditions.
- In practice, it is pretty difficult to accurately differentiate between collision and wireless loss. The current MIMO RA algorithm, embedded in Ath9k driver for 802.11n products, probes with only one attempt that makes it sensitive to the single failure of a probe packet. Hence, the RA algorithms must be robust against collision losses.
- Also, the current drivers for WLAN devices are pretty slow in adapting to the time-varying channel conditions. The RA algorithm should be quickly adaptive and accurate in rate adjustment to maximize the channel efficiently, which means it is able to immediately jump to the rate that may yield the best performance.

In 802.11n WLANs, the choice of the MCS has a direct impact on the throughput performance. A static transmission (i.e., with a fixed number of antennas) cannot maximize the link's capacity due to the rapid changes in the wireless channel state. It has been shown in [19] that a dynamic transmission strategy is needed to improve the network's performance by taking advantage of spatial diversity. Then, the goal of link adaptation is to choose the suitable number of spatial streams that maximize the throughput or minimize the Packet Error Rate (PER), which are directly related to the SNR in the channel.

The L3S RA algorithm maintains present channel transmission statistics (acknowledged packets, PER, and achieved throughput) for each used transmission rate. These statistics provide an elegant way to obtain the necessary CSI feedback for link state estimation. In our proposed L3S RA algorithm, the collected statistics at the sender are divided into two categories for a best adaptation to the time-varying channel characteristics. The Long-term and the Short-term Statistics are continuously updated by monitoring the transmission results. They are reset whenever the principal rate is changed. The scheme also decides how long the sender should stay at the current transmission rate, when and how to probe at candidate rates, and whether or not to switch to a new rate.

A. *Long-term Statistics for System Stability*

The *Long-term Statistics* are maintained to adapt the transmission rate, which provides the best throughput, against sustained changes in the link. The potential best rate is determined by estimating the observed throughput performance whether it has been used in the past transmissions or by the judgment that enhances the system stability. *Long-term Statistics* (achieved throughput, packet error rate) within a window of certain number of packets can decide whether a probing rate can be selected. These statistics are related to the effective throughput, and then they guarantee that this throughput is maximized on the long-term. Thus, our proposed algorithm probes intelligently by sending data packets at the candidate rates that may improve the performance. Based on PER, the expected throughput is calculated to gauge the performance on the currently used rate. In order to have an advanced and efficient rate selection that instantaneously estimates the channel quality, the long-term transmission rate is not changed during the probing and its throughput performance is evaluated with that of probing rates. As long as the sender observes that the selected rate does not perform as well as expected, it falls back immediately to the pervious MCS. Specifically, our scheme continually examines the long-term data rate during the transmission periods and can tune to the channel variations as soon as possible.

In the (Single-Input Single-output) SISO case, the decision is simply to up or down the scaling by one, which may lead to a premature rate switch decisions. However, in the MIMO technique case, the additional transmitter antenna(s) can be used for diversity gain. The *probe( )* function is periodically invoked, which intelligently probes at the possible probing directions. The probing rates are adjusted according to the multi-rate retry mechanism. Figure 1 shows the six proposed probing directions, which correspond to the following functions: *Up_Probe( )*, *Down_Probe( )*, *Right_Probe( )*, *Left_Probe( )*, *Left_Up_Probe( )*, and *Right_Down_Probe( )*.

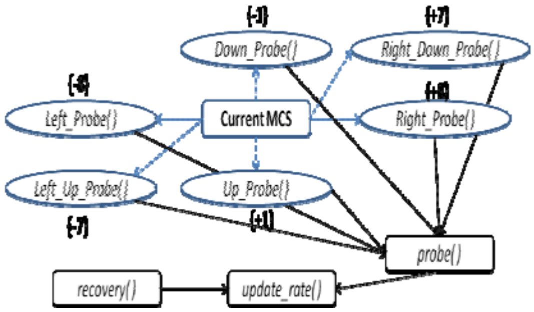


Figure 1: Different Probing Directions

B. *Short-term Statistics for System Responsiveness*

The Short-term Statistics are used to control the transmission rate against transient variations that improve the system responsiveness. A wireless link usually behaves diversely and does not follow a fixed success or failure pattern. In that case, consecutively received and lost ACK frames may respectively reflect a fast improving and deteriorating channel conditions in a short interval. However, the sender should, before a probe, justify whether there is short-term or long-term variations in the channel quality. Thus, the short-term changes are handled by a multi-rate retry mechanism. Incorporating these series, we use the counter *consecutive_retries* for each retry rate initialized in the transmission descriptor, which denotes the number of consecutive packets that fail at the attempts. Also, the ACK packets provide binary information to the sender whether its choice of the MCS was supported by the channel. Similar to the ARF-like algorithms in the SISO case, our sender maintains two other counters of *consecutive_successes* and *consecutive_failures*, but used to update dynamically the beginning of the next probing period. They mean the total number of consecutive packets that succeed and fail at all of the attempts. These two counters tackle the long-term variations by calling the probe function to change the long-term rate immediately or after a period of time.

C. *Implementation of Algorithm in Ath9k*

In our algorithm, the short term changes in the link quality are controlled by multi-rate retry series, while the long term variations are manipulated by changing the values of r_i. However, it keeps the mechanism of multi-rate retry with four retry series, but modifies the values of c_i's. Thus, this mechanism can be used to solve transient channel variations by retransmitting the lost packet at different data rates, which are always in decreasing order, until a success or exceeding the retry limit. The successive retry values of the four multi-rate retry series are important to the responsiveness performance. A high value may cause additional retransmissions where the channel condition is really degrading. On the other hand, a small value may cause a transmission at a reduced rate fairly quickly. The multi-rate retry series mechanism is used during the transmitting (*Tx* state) and the probing (*Probe* state) periods. For both states and with more than one attempt, we do not rush to a reduced rate once the first attempt fails, because it is very probably that such a failure is caused by a collision. Therefore, we set c_1 to two, which means that the first and second attempts use the long-term-rate in the case of transmission period or the probe-rate in the gauging period. Similarly, c_2 and c_3 retry limit are set to two to ensure fast adaptation to short-term channel variations. Whereas, the first rate series r_0 with four as successive retry limit, is always used to send the control packets at the lower MCS with a single antenna before data transmissions.

Table II
MCS Set Comparison of Existing and Proposed Algorithms

Modulation Type	Existing RA Algo.		Proposed RA Algo.	
	1 stream	2 streams	1 stream	2 streams
BPSK	13.50	-	13.50	27.00
QPSK	27.00	-	27.00	54.00
QPSK	40.50	-	40.50	81.00
16-QAM	54.00	108.00	54.00	108.00
16-QAM	81.00	162.00	81.00	162.00
64-QAM	-	216.00	108.00	216.00
64-QAM	-	243.00	121.50	243.00
64-QAM	-	270.00	135.00	270.00

The driver Ath9k uses a basic MIMO mode where the same modulation and coding schemes are employed for all spatial streams of an MCS. Table II shows the basic MCS set, with a maximum of two spatial streams for 40 MHz and GI of 800 ns, which can be supported by the current driver. For this existing MIMO rate control, the used transmission rates are sorted, which is similar to the rate control algorithms in SISO case. This strategy makes the adaptation very slow in the presence of random wireless errors. Eventually there are too many packet losses before it finds a proper rate for the reason that it can only step down to the next lower rate for each interval. Whereas, our L3S algorithm takes advantage of all the available MCSs in the IEEE 802.11n standard.

In our scheme, the long-term transmission rate *tx_rate* for each station is adapted according to the sender's state. The state transitions and the used functions are illustrated in Figure 2. Specifically, a sender has two states, the *Tx* state and the *Probe* state. In each round, the transmitter moves periodically between these two states and updates continuously the associated statistics. When the sender adjusts the long-term MCS, the statistics of both periods are reset which means the commencement of a new round.

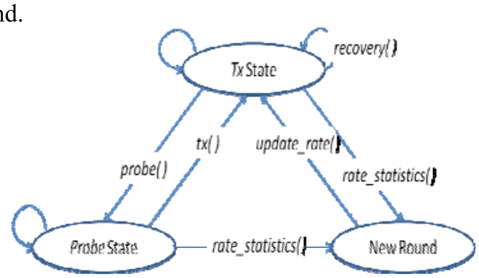


Figure 2: State Transitions at the sender

The rate selection is controlled by the function *get_rate( )*, which decides the rates r_i and their corresponding retry numbers c_i for a new outgoing packet. At the beginning of each round, the sender is in the *Tx* state and transmits data at $r_1 = tx\_rate$. In the same way as in ONOE, the remaining rates r_2 and r_3 of multi-rate retry mechanism are filled with the next successive lower rates. Meanwhile, the sender observes the transmission results of the current rate. Specifically, it calculates the throughput by monitoring the PER. In practice, the average collision rate for TCP is around 11%, so the PER values less than this threshold are ignored. After a period of time, the sender sets up two successive probe series and enters into the Probe state, but *tx_rate* is not altered. Similarly, it keeps on observing the probing results at the candidate rate. The first series is employed to probe in the current used spatial stream(s), where a constant fraction (10%) of the data is sent at the two adjacent rates of the long-term rate. Therefore with multi-rate retry series, $r_1 = Up\_Probe(\,)$, $r_2 = tx\_rate$ and $r_3 = Down\_Probe(\,)$. On the other hand, the probing in the second series is based on the number of streams currently used. Using the same method, another window of ten percent of the packets is sent at other candidate rates that are sorted as always in decreasing order. Thus, if the sender transmits with only one spatial stream, we call the functions *Right_Probe( )*, *Right_Down_Probe( )* that means an additional transmit antenna is used for the following queued packets. Otherwise, the called functions are *Left_Probe( )* and *Left_Up_Probe( )* for the cases, the long-term transmission rate *tx_rate* is kept in the multi-rate retry series during the probing periods. After these two probe series, the sender goes back to *Tx* state. In the *update_rate( )* procedure, the performances of all five rates are evaluated and the best rate is selected as a new long-term MCS, which is critical to the achieved throughput. Whereas, the *recovery( )* function describes when the sender is needed to fall back immediately to the previous long-term MCS. After each long-term transmission rate adjusting, the next probing time *probe_interval* is set to the default period of 60ms.

The trigger of the next probing period is dynamically changed depending on the present channel characteristics. The sender collects transmission statistics in the function *rate_statistics( )*. After a frame transmission, the returned hardware status in the descriptor is either *successes* or *retries*, which is used to decide when the next *probe* state will be triggered. Note that the transmission *successes* status does not necessarily mean that only the first attempt is successful, but that the frame may be successfully transmitted after several retries. The *retries* status means that the attempt is failed.

IV. Performance Evaluation

A. *Experiment Settings*

We implemented our rate selection mechanism using Ath9k [15] driver for IEEE 802.11n devices with Atheros chipsets. We considered the network topology shown in Figure 3. The different hardware and software configuration parameters used in our experiments are listed in Table III. The Access Point (N_0) and the PC (N_1) wireless card adopt "2x2" MIMO technology for data rates of up to 300 Mbps. In order to make the scenario more realistic, we used a tunable waveform generator that allows injecting Adaptive White Gaussian Noise (AWGN) at different power levels and a desired frequency band. Since, TCP (Transmission Control Protocol) use processes to check that the packets are correctly sent to the receiver, we used Netperf [17] to generate continuous saturated TCP packets and report the achieved throughput for each network link between the client (N_1) and the server (N_2).

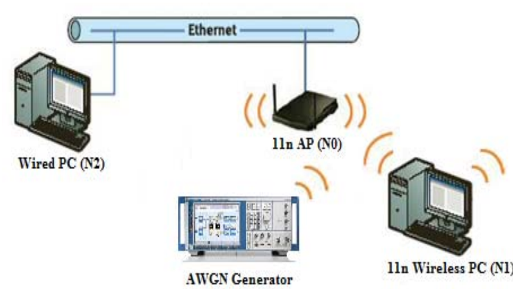


Figure 3: Network Topology in the Experiments

Table III: Configuration Parameters

Parameters	Values
Signal generator	Rohde&Schwarz SMU200A
WLAN card	TL-WN851N (2Tx/2Rx)
WLAN router	TL-WN841N (2Tx/2Rx)
Operating System	Linux kernel 2.6.32.12
Device driver	Ath9k
Wireless PHY	802.11n
Channel Bandwidth	40 MHz
Transmit power	20 dBm
Frequency band	2.427 GHz

All equipments are placed in a typical office environment with several concrete obstacles, and then the paths between the AP and the client PC may not be line-of-sight. We run experiments with a static client but in different channel conditions by changing the SNR. To get RSSI variations such as in mobility cases, we modify the noise power level that is injected by the AWGN generator. Every experiment is run for about 20 minutes. The throughput results for each RSSI value are grouped and a statistic average measure is reported over three experimental runs. For performance evaluation, we designed three different experimental scenarios. In the scenario S_1 and S_3, the injected noise is respectively varied slowly in decreasing order and rapidly in increasing and decreasing order, at the sender area. For the scenario S_2, we decreased slowly the SNR at both the client and the AP sides.

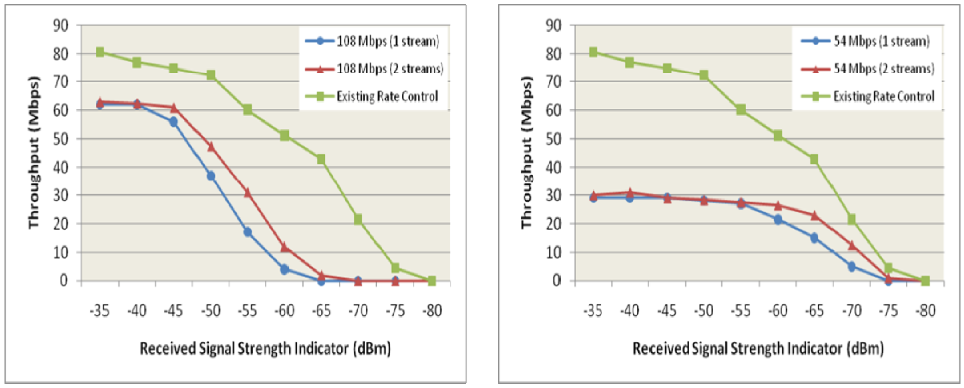
B. *Experiment Results*

We have implemented our RA algorithm on a real 802.11n wireless card based on Atheros AR9220 chipset [16]. To analyze its performance compared to the scheme embedded in Ath9k driver, we carried out extensive experiments under various link conditions. So, we measure the stability (by average throughput) and the responsiveness (by response time). Stability means the ability to maintain optimum rate. Thus, the RA algorithm can achieve the desired performance reference and it is robust to disturbance. On the other hand, responsiveness measures how fast RA takes for the algorithm output to achieve the new desired rate, in reaction to channel changes. It reflects the ability to adapt to quick channel variations in loss rate.

At first and in order to check the effects of the improvement reached on a MIMO WLAN system by the additional transmitter antennas, we study the TCP throughput performance of existing Atheros MIMO RA with static rates via different number of antennas. Figure 4(a) shows the throughput achieved by the transmission rate 108 Mbps across one and two antenna(s) with different RSSI values. So as to verify again the effects of an extra antenna, we repeat the same experiment but with the transmission rate of 54 Mbps, the throughput achieved is shown in Figure 4(b). The throughputs achieved by these two fixed rates are included as performance benchmarks. For both figures, initially, when the channel condition is good, the transmissions with a single stream and the transmissions with double streams can achieve similar optimal throughput. When the channel deteriorates, the throughput achieved by using a single antenna reduces faster than the throughput achieved by simultaneous transmissions. The reason is attributed to the signal reflection and multipath that improve the received signal strength. All these points out the fact that transmitting a single spatial stream across two antennas at the same time can greatly improve the reliability. Also, these two first figures 4(a)-(b) show that the gain in throughput achieved by the existing RA is extremely high compared to that achieved by a fixed transmission data rate. This proves that the automatic rate selection capability in IEEE 802.11 devices is very important to deal with channel conditions.

Now, we study the throughput performance of two algorithms under different scenarios. In all scenarios, our new RA algorithm achieves the highest throughput by managing to send more packets at higher rates, mainly due to the intelligent and dynamic probing. The comparative throughput with varying RSSI is shown in Figure 5. In Figure 5(a), where the link between the client and the AP is stable and with slow RSSI variations, both algorithms can initially achieve almost similar high throughput when the channel condition is good (RSSI > -45). But, when the SNR increases, the throughput achieved by the existing scheme reduces faster than that of our algorithm. The reason is attributed to the several failed transmissions and the longer time spent to reach the best data rate. The gain in throughput achieved is almost ~ 20%. This is also true in Figure 5(b), where the channel condition worsens, but with 15% of gain. This figure demonstrates that the TCP throughput achieved by our proposed RA algorithm significantly outperforms that achieved by the existing Atheros MIMO RA scheme, which can hardly attain the threshold to scale up by one. However, our rate control can respond to the variations rapidly enough by probing intelligently at the candidate rates and jumping immediately to the new long-term data rate. Also, the feature of dynamic probing time can improve more the TCP throughput performance by decreasing the waste of time, as shown in Figure 5(a). This is due to the fact that our new scheme probes quickly when the channel quality changes by using the short-term statistics. On other hand, the existing RA spends much more time before reaching the best MCS. It waits almost 10 seconds at each MCS before scaling to the next upper rate.

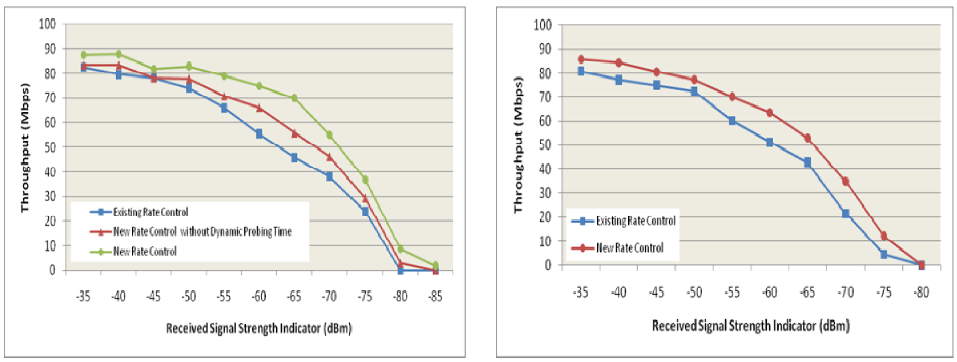
Another feature of our proposed scheme is its rate stability by using long-term statistics. The dominant factor to decide a rate change is the expected PER or equivalently the expected throughput. In the probing periods, the long-term rate is not changed, since it may be suboptimal under some transient channel conditions. Also, in our scheme, transient variations are resolved by using the multi-rate retry mechanism. Indeed, if it selects a wrong data rate to probe, it can perceive this error after two consecutive failed attempts and continue to transmit and evaluate at the same time the next lower MCS.



(a) Throughput Achieved by 108 Mbps with Different Number of Spatial Stream

(b) Throughput Achieved by 54 Mbps with Different Number of Spatial Stream

Figure 4: Throughput Comparison of Fixed Rates for TCP Traffic under Scenarios S_1



(a) Comparative Throughput under Scenario S_1

(b) Comparative Throughput under Scenario S_2

Figure 5: Throughput Comparison for TCP Traffic on Two Different Scenarios

Under favorable link conditions, the transmit queue is almost empty resulting in less packet delay. However, with link quality degrading, the number of packets in the queue grows which is caused by the retransmissions. The goal of the rate adaptation algorithm is to select instantaneously the best MCS and transmit the packets back to back until the queue becomes empty again.

V. Conclusions

IEEE 802.11 is the most popular standard used for WLANs today. IEEE 802.11n is a recent amendment for supporting high transmission rates, by introducing enhancements, over the legacy a/b/g standards. Many factors make the wireless channel very dynamic. Rate adaptation is extremely important to the throughput performance of WLANs with multi-rate capabilities.

In this paper, we have proposed a new sender-base RA algorithm, which controls both short-term and long-term channel quality variations. The novelty of our scheme lies in its great adaptability to a variety of link conditions. The experiments on typical 802.11n networks show that it is able of achieving noticeable improvements in throughput compared to the present MIMO RA implemented in Ath9k driver. The reason is attributed to our proposed strategy of quick rate adaptation and mostly the intelligent and efficient rate selection method.

References

[1] IEEE Draft Std P802.11n/D3.01, "Draft Standard for Information Technology Telecommunications and Information Exchange between Systems-Local and Metropolitan Area Networks-Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Amendment4: Enhancements for Higher Throughput," Nov. 2007.

[2] B. Bjerke, J. Ketchum, R. Walton, S. Nanda, I. Medvedev, M. Wallace, and S. Howard, "Packet Error Probability Prediction for System Level Simulations of MIMO-OFDM based 802.11n WLANs," in *Proc. IEEE ICC*, (Seoul, Korea), May 2005, pp. 2538-2542.

[3] A. Kamerman and L. Monteban, "WaveLAN-II: "A High-Performance Wireless LAN for the Unlicensed Band," *Bell Labs Technical Journal*, pp. 118-133, Summer 1997.

[4] M. Lacage, M. Manshaei, and T. Turletti, "IEEE 802.11 Rate Adaptation: A Practical Approach," in

Proc. ACM MSWiM'04, (Venice, Italy), pp. 126-134, Oct. 2004.

[5] J. C. Bicket, "Bit-rate Selection in Wireless Networks," Master's thesis, Feb. 2005.

[6] "ONOE Rate Control." [Online], Available: http:// www.madwifi.org/browser/madwifi/trunk/ath_rate/onoe

[7] G. Judd, X. Wang, and P. Steenkiste, "Low-Overhead Channel-Aware Rate Adaptation," in *Proc. ACM MobiCom'07*, (Montreal, Canada), pp. 354-357, Sept. 2007.

[8] G. Holland, N. Vaidya, and P. Bahl, "A Rate-Adaptive MAC Protocol for Multi-Hop Wireless Networks," in *Proc. ACM MOBICOM'01*, (Rome, Italy), pp. 236-251, July 2001.B.

[9] Sadeghi, V. Kanodia, A. Sabharwal, and E. Knightly, "OAR: An Opportunistic Auto-Rate Media Access Protocol for Ad-Hoc Networks," *Wireless Networks*, vol. 11, no. 1-2, pp. 39-53, Jan. 2005.

[10] D. Qiao, S. Choi, and K. Shin, "Goodput Analysis and Link Adaptation for IEEE 802.11a Wireless LANs," *IEEE Trans. on Mobile Computing*, vol. 1, no. 4, pp. 278-292, Oct.-Dec. 2002.

[11] I. Haratcherev, K. Langendoen, R. Lagendijk, and H. Sips, "Hybrid Rate Control for IEEE 802.11," in *Proc. ACM MobiWac'04*, (PA, USA), pp. 10-18, Oct. 2004.

[12] 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.

[13] Y. Yang, M. Marina, and R. Bagrodia, "Experimental Evaluation of Application Performance with 802.11 PHY Rate Adaptation Mechanisms in Diverse Environments," in *Proc. IEEE WCNC'06*, (Las Vegas, USA), pp. 2273-2278, Apr. 2006.

[14] K. Ramachandran, H. Kremo, M. Gruteser, I. Seskar, and P. Spasojevic, "Experimental Scalability Analysis of Rate Adaptation Techniques in Congested IEEE 802.11 Networks," in *Proc. IEEE WoWMoM'07*, (Helsinki, Finland), pp. 1-12, June 2007.

[15] "Ath9k 802.11n Drivers." [Online], Available: http:// www. wireless.kernel.org/en/users/Drivers/ath9k/.

[16] "Atheros Communications," [Online], Available: http:// www. atheros.com/news/AR9220.html/.

[17] Netperf, "The Networking Benchmark." [Online], Available: http://www.netperf.org/netperf/.

[18] A. Roy, N. Saxena, "A New Channel Aware Rate Adaptation in High Speed WLANs," *IEICE Trans. on Communication Letters*, Vol. E92.B, No. 6, pp.2345-2348, Jun.2009.

[19] "HT PHY Specification, Enhanced Wireless Consortium v1.27," [Online], Available: http:// www.hydrus.et.put. poznan.pl/~rkotrys/802_11/EWC_PHY_ spec_V127.pdf.