Wireless Networking

Simone Orru'- Hiram Rayo Torres

April 2018

1 Introduction

This assignment consists in the implementation of a rate control algorithm to dynamically adjust the modulation coding scheme of an 802.11ac channel. The purpose is to maximize throughput while lowering bit error rate.
As baseline for this assignment, we use the "TransmitRateControl" Matlab example and develop our implementation on top of the existing simulation. In addition, the example is used as a comparison meter to evaluate the results obtained by different implementations and make performance assessments.

2 Algorithm Description

First, we decided to search through literature in order to find a suitable algorithm that we could implement in our testing scenario, and that promised good results. After considering our options, we decided to focus in reproducing the results of nDira[1], a two stage rate control algorithm based on channel efficiency tables and bit error rate exponential moving average.

The paper proposes a two stage approach: the first determines whether to change the current modulation scheme by comparing the estimated SNR per packet with a series of pre-computed threshold tables. If the estimated SNR is found to be within a certain range, the MCS is increased or decreased accordingly. In order to calculate these thresholds, an estimation of the link efficiency needs to be previously computed. The paper proposes to build large efficiency tables sweeping over all the different modulation and coding schemes in order to estimate those thresholds for each different setup. The second stage has the purpose of verifying the decision of the first stage by calculating the Exponential Moving Average (EMA) on the bit error rate over the previous 100 packets. If the BER value surpasses a certain threshold, the decision of the first stage is discarded and the MCS value is lowered instead to prevent the bit error rate from increasing. It is important to mention that the threshold needs to be fine tuned to avoid compromising the throughput in exchange of really low bit error rate.

3 Motivation

We opted to emulate this algorithm due to the interesting approach it presented and the possibility of reproducing this results on our setup. Even though the BER and SNR are strictly correlated we don't think the two stages conflict with each other. The first stage uses the estimated SNR of the last packet to check if the modulation scheme is the one with the highest efficiency, trying to maximize the throughput and basing the estimation only on the last packet. The second stage introduces the concept of memory in the system, in order to discard any decision of the first stage that goes against the general trend of the channel.

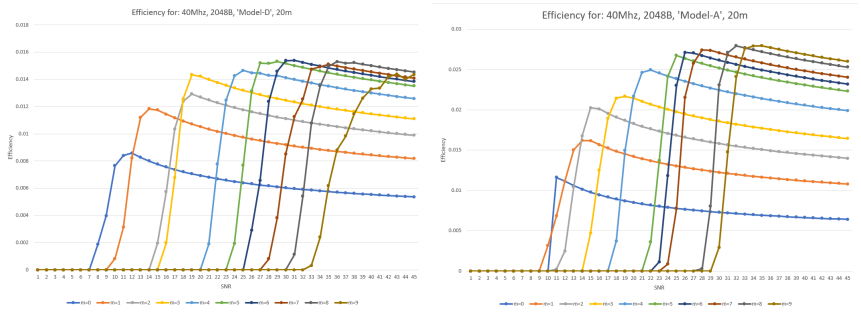
4 Algorithm Implementation

In order to obtain our simulation efficiency tables, we started by modifying the experimental setup to run simulations with stationary and user defined SNR.

To measure the channel efficiency we used the following formula:

$$\eta_L = \frac{1}{N_F} \sum_{i=1}^{N_F} \frac{PL_i}{t_{f,i} B \log_2 (1 + SNR)}$$

Where N_F is the total number of packets sent, PL_i is the payload of packet i in bits (only if received correctly, 0 otherwise), $t_{f,i}$ is the time of transmission of packet i in seconds, and B is the bandwidth of the channel in Hz.
Due to timing constraints and also for simplicity of the implementation, we focused only in building the efficiency tables of setups which were considered relevant. We considered bandwidth of 20-40-80-160MHz in order to cover the whole spectrum offered by the simulation environment. We then modified the payload, using firstly 1024Byte then 2048Byte and finally 4096Byte. In addition, we tried changing other parameters, like the distance and channel delay profile.
Finally, we swept the SNR from 1 to 45 for each one of these configurations to obtain the graphs from where the threshold tables would be extracted. It is important to mention that we limited the number of packets to 100 to prevent the simulations to become increasingly complex and time consuming.



In order to obtain the highest possible efficiency, which implies a higher throughput and low bit error rate we built our thresholds tables to make sure that we remain always on the highest curve for each SNR value. However, for simplicity of our implementation, we neglected the small variations due to distances and delay profiles and only implemented threshold tables based on payload and Bandwidths

Note: All the graphs from where the thresholds were extracted can be found in the Git Repository.

Once the thresholds have been defined, we proceeded with the exponential moving average (EMA) implementation. We calculate the EMA over bit error rate using the following equation:

$$EMA = \frac{x_{t-1} + \lambda x_{t-2} + \lambda^2 x_{t-3} + \lambda^3 x_{t-4} + ... + \lambda^{n-1} x_{t-n}}{1 + \lambda + \lambda^2 + ... + \lambda^{n-1}}$$

Where n is the window size, and λ is a weighting factor empirically determined. The value obtained from this equation will be compared to a fixed threshold ϵ that can lead to the confirmation of the MCS proposed at the first stage or to its replacement with a lower MCS in case of a threshold violation due to a high EMA.

The publication suggests that a window of 100 packets, a threshold value of 0.5 and a weighting factor(λ) of 0.75 gives good results. However, because of our limited number of packets, we chose to reduce the window size to 20 packets and to use an experimentally chosen threshold of 0.0138.

5 Experiment Results

In order to make an assessment of our implementation. We created a script to test and compare our implementation with the Matlab example in all possible scenarios by modifying delay profiles, distances, bandwidths, etc. We tested both implementations with a randomized SNR pattern in order to simulate real

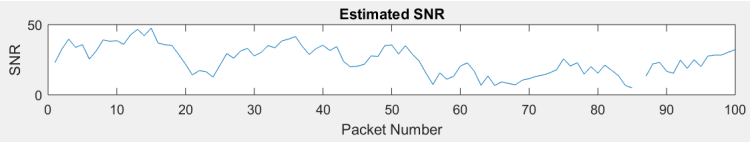


Figure 1: SNR used for testing

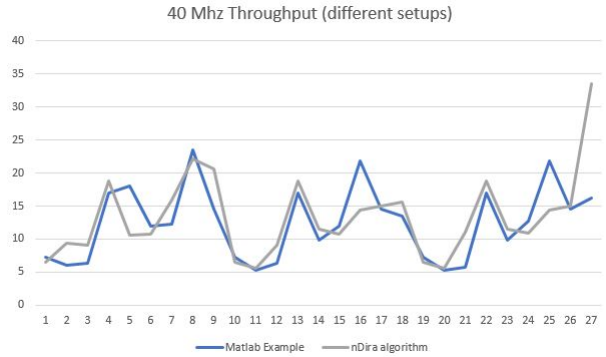


Figure 2: 100 packet Throughput using 40MHz different setups

life SNR behaviour to determine whether our implementation actually gives better results.

Note: Each number on the horizontal axis of the graphs represent a different setup using the same Bandwidth. To analyze the difference between setups, refer to the excel tables provided in the repository.

Looking at the graphs we can observe that the results obtained with our implementation only provide better results under specific circumstances. However, we do observe that the bit error rate of our implementation is overall better than the results given by Matlab, which implies that the threshold chosen for the EMA could be modified to obtain a better throughput at the cost of increasing the bit error rate. However, a much deep study could be made to choose a desired perfect balance between both.

To further test our implementation, we try to approximate the testing scenario even more to real life applications by sending more than a 100 packets. For this, we test with 200 packets to analyze the performance of both implementations. It is important to mention that since we defined our thresholds based on averaging the behaviour of a 100 packets, it is quite possible that the results are not as good as they could be if we defined the thresholds once again with an even higher number of packets. Also, the EMA window and threshold could be re-adjusted to perform better

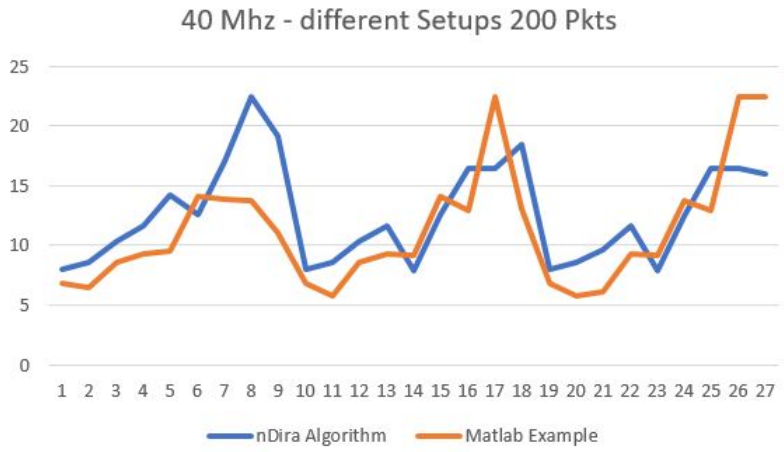


Figure 3: 200 packet Throughput using 40MHz different setups

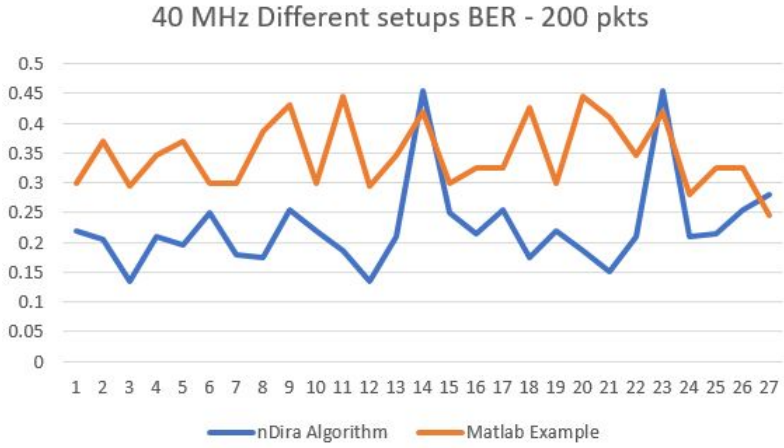


Figure 4: 200 BER - 40MHz different setups

We can observe that the results with 200 packets improved. However there are still specific setups in which Matlab does better. This could be improved by further fine tuning our SNR thresholds. Also, we can observe how the EMA succesfully ensures a low BER which performs way much better than the Matlab example. It is important to mention that we could fine tune this EMA threshold to allow for more throughput at the cost of a higher BER that could still be below the Matlab results.

6 Final Remarks

By means of this assignment we learned the role that rate control techniques play in the system performance. However, we believe that a much longer time should be dedicated in the construction of the tables than the one we were able to give it in order to obtain a substantial improvement over our simplified implementation. Each situation should be considered to construct a specific table. In addition, this tables should be constructed by using thousands of packets to improve accuracy. However this represented a limitation to our current timing and computation power.

Finally, even though we managed to obtain some improvement, one drawback that we observe on this approach is the necessity of building efficiency tables beforehand, since we believe it is very unlikely that these values, which are built on a specific setup , would work equally well on other environments.

References:

[1] Efficient Adaptation of Modulation and Coding Schemes in High Quality Home Networks. Hendrik Koetz and Ruediger Kays Communication Technology Institute, TU Dortmund University, 44221 Dortmund