ET4394
Wireless Networking

802.11 Dynamic Rate Control Algorithm

Students	Manasa Dattatreya	4749634
	Pradeep Venkatachalam	4735323

TU Delft

Contents

1. Introduction

IEEE 802.11 wireless systems have rate control algorithms which are built to adapt the transmission rate between channels in response to changing channel conditions.

The algorithm implemented has it's principles loosely derived from the **Minstrel Rate Control Algorithm**. Most code bases rely on the assumption that packets sent as slow data rates have a higher chance of success than one at higher data rate. However, in reality, packets at a lower rate has a higher probability of being shot down by some other node sending a packet. The Minstrel selects rates that will give an approximation to the best available throughput, while delivering packets with the highest probability achievable given link condition.

Rate Control Algorithms are typically built on analyzing the link performance metrics and the channel quality. Under the former, we have SNR and Bit Error Rate. From the model given, we vary the Modulation and Coding Scheme (MCS) based on the properties of the underlying propagating channel. We wish to maximize the link throughput through a closed loop feedback scheme.

The original algorithm estimates the SNR values of the received packet, compares it against a pre-defined threshold and identifies the suitable Modulation and Coding Scheme for transmission of the next packet.

From experimentation and observation, we derive that the Bit Error Rate depends on the Channel conditions, SNR and the MCS. Since the relationship between the environmental variables is not very well defined, the results of the performed experiment are obtained after taking a slightly heuristic approach, based on experimentation, evaluation and trail and error methods.

2. Implementation

This project involves taking into consideration the Bit Error Rate of the transmitted packet along with the SNR to compute the appropriate Modulation and Coding Scheme to be used. The Bit Error rate is a measure of success-fullness of packet transmission. Thus, it should play an important role in adjusting the Modulation and Coding scheme used .

We have introduced an offset variable in case the previous packet triggers an error. This affects the Modulation and Coding Scheme of the next packet. The implementation is shown below.

```
    val = ber(numPkt);
if ber(numPkt) > 0
    offset = 1;
else
    offset = 0;
end

% Compare the estimated SNR to the threshold, and adjust the MCS value
% used for the next packet. Apply offset of the BER if applicable

MCS(numPkt) = cfgVHT.MCS; % Store current MCS value
increaseMCS = (mean(y.EstimatedSNR) > snrUp((snrInd==0)+snrInd)+offset);
decreaseMCS = (mean(y.EstimatedSNR) <= snrDown((snrInd==0)+snrInd)+offset);
if snrInd >= 1
snrInd = snrInd-decreaseMCS
end
snrInd = snrInd+increaseMCS

if snrInd < 9
cfgVHT.MCS = snrInd-1;
else
    cfgVHT.MCS = 8;

end
```

```
end

%Certain Boundary checks had to be done in some transmissions,
%the MCS value exceeded the allowed limit.
```

3. Observations

Four sets of values are taken and tested against the original algorithm and the implemented algorithm. Each set was checked with Models C-F with bandwidth 20, 40, 80 MHz.

- The **Default Model 0** presented, with mean SNR = 22 and variation in SNR = 14
- **Default Model 1** with a jump of 0.25
- **Default Model 2** with a jump of 1.0
- **Model 3** with mean SNR = 20, variation in SNR = 10

The results have been tabulated below.

- Improvements in the performance by the new algorithm are indicated by **Bold characters**
- Results where there the performance is worse compared to the original algorithm is indicated by *Italic Characters*
- Results where there is tradeoff in the output observed is indicated by ***Bold Italic Characters***
- The values for which comparision is not possible or if the performance obtained is the same has no special highlighting and are represented by normal characters.

Default Model 0 With Original Algorithm (Mean = 22, Amplitude = 14)

Statistics	Model (Throughput (Mbps) , Error Rate)			
Bandwidth (MHz)	C	D	E	F
20	No Value	14.281/0.06	13.3308/0.13	NoValue
40	32.5183/0	22.577/0.08	26.589/0.08	27.7734/0.04
80	37.0574/0.04	35.9141/0	36.7388 / 0.02	32.8472 / 0.22

Default Model 0 With Implemented Algorithm (Mean = 22, Amplitude = 14)				
Statistics	Model (Throughput (Mbps) , Error Rate)			
Bandwidth (MHz)	C	D	E	F
20	22.0694/0.01	*13.8 / 0.08*	**13.7455/0.1**	14.6927 / 0.1
40	**33.889 / 0**	**22.9/0.05**	**28.2616/0.05**	***28.11/0.06***
80	***36.4306/0.03***	35.9141/0	36.7388 / 0.02	**36.6342 / 0.14**

Model 1 With Original Algorithm (Mean = 22, Amplitude = 14) and Jump 0.25

Statistics	Model (Throughput (Mbps) , Error Rate)			
Bandwidth (MHz)	C	D	E	F
20	No Value	14.4/0.05	13.8/0.1	NoValue
40	33.06/0	23.3143/0.06	29.4233/0.02	26.0001/0.08
80	36.781/0.06	37.2076/0	35.79/0.05	34.36/0.19

Model 1 With Implemented Algorithm (Mean = 22, Amplitude = 14) and Jump 0.25				
Statistics	Model (Throughput (Mbps) , Error Rate)			
Bandwidth (MHz)	C	D	E	F
20	21.9/0.01	***13.7/0.05***	***13.7/0.09***	16.2561/0.1
40	*32.4/0.01*	**23.4/0.04**	29.4233/0.02	**28.4105/0.07**
80	**39.2/0.03**	37.2076/0	**36.1478/0.04**	**36.5923/0.14**

Model 2 With Original Algorithm (Mean = 22, Amplitude = 14) and Jump 1.0

Statistics	Model (Throughput (Mbps) , Error Rate)			
Bandwidth (MHz)	C	D	E	F
20	No Value	No Value	No Value	NoValue
40	33.1/0	20.0/0.15	27.6619/0.05	25.3/0.06
80	37.0/0.03	36.79/0.02	35.6/0.05	31.5/0.26

Model 2 With Implemented Algorithm (Mean = 22, Amplitude = 14) and Jump 1.0				
Statistics	Model (Throughput (Mbps) , Error Rate)			
Bandwidth (MHz)	C	D	E	F
20	23.1/0.01	14.8/0.08	13.5/0.07	13.9/0.09
40	*30.7/0.01*	**22.94/0.08**	***26.52/0.03***	**30.3/0.02**
80	**37.195/0.02**	***35.5/0.02***	35.6/0.05	**35.09/0.17**

Model 3 With Original Algorithm (Mean = 20, Amplitude = 10)

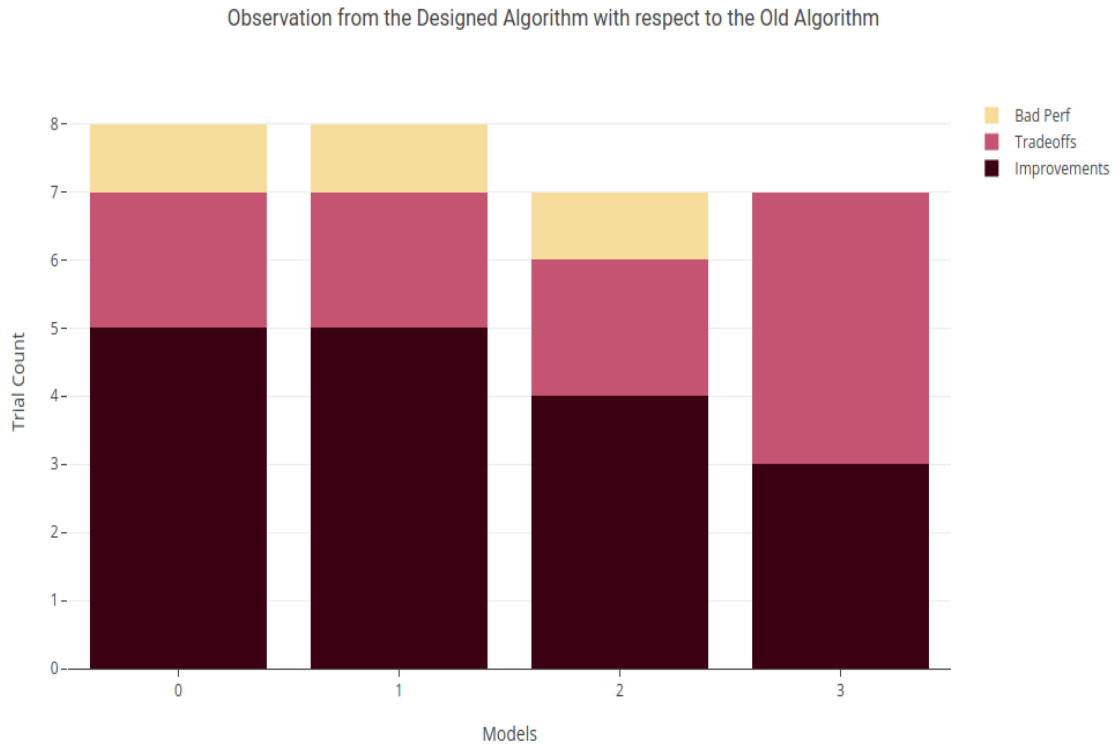
Statistics	Model (Throughput (Mbps) , Error Rate)			
Bandwidth (MHz)	C	D	E	F
20	19.9416/0	13.72/0.05	11.97/0.14	15.3165/0.07
40	32.2774/0	22.9/0.05	28.2/0.03	27.8992/0.03
80	37.35/0.02	35.05/0	35.01/0.05	38.9/0.05

Model 3 With Implemented Algorithm (Mean = 20, Amplitude = 10)				
Statistics	Model (Throughput (Mbps) , Error Rate)			
Bandwidth (MHz)	C	D	E	F
20	19.9416/0	***13.6619/0.04***	**13.4066/0.08**	15.3165/0.07
40	32.2774/0	**23.237/0.04**	***28.3419/0.04***	***27.9352/0.04***
80	**39.1885/0.02**	35.0595/0	***34.4398/0/04***	38.9/0.04

4. Inferences and Conclusions

We have observed a well defined increase in performance for most of the test cases that have been executed. Changing the MCS value based on the Bit Error Rate thus proved to be better in increasing the throughput of the transmission without affecting (or reducing) the Bit Error Rate.

Thus, through this algorithm, we are introducing a variation based on the probability of success of the pre-



ceding packet.

The original Minstrel Algorithm collects statistics from all packets transmitted. It also sends a certain portion of it's packets at rates that are considered to be non optimal, in order to gather more information. Our implementation only takes into consideration consecutive packets, but a better implementation would record the degree of success of a group of packets being transmitted and make an appropriate evaluation on the optimal Coding Scheme.