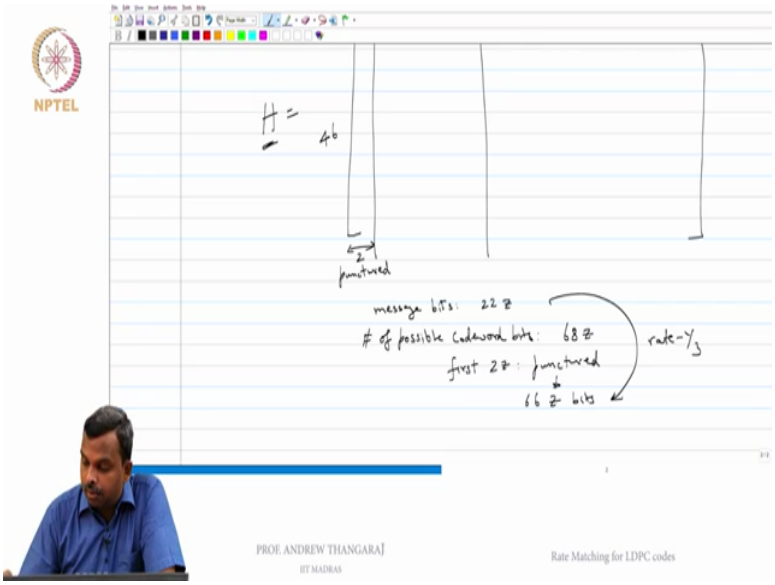
LDPC and Polar Codes in 5G Standard
Professor Andrew Thangaraj
Department of Electrical Engineering
Indian Institute of Technology, Madras
Rate Matching for LDPC codes

(Refer Slide Time: 00:18)



Okay so if you remember the parity check matrix has this form okay so if you take the first base graph for instance this is going to be 46 rows and 68 columns and first 22 are message rows and the remaining 46 are parity rows okay and there is an expansion factors that right so that is okay so whatever the expansion factor is it will finally come into the picture and the first two columns are punctured always right okay.

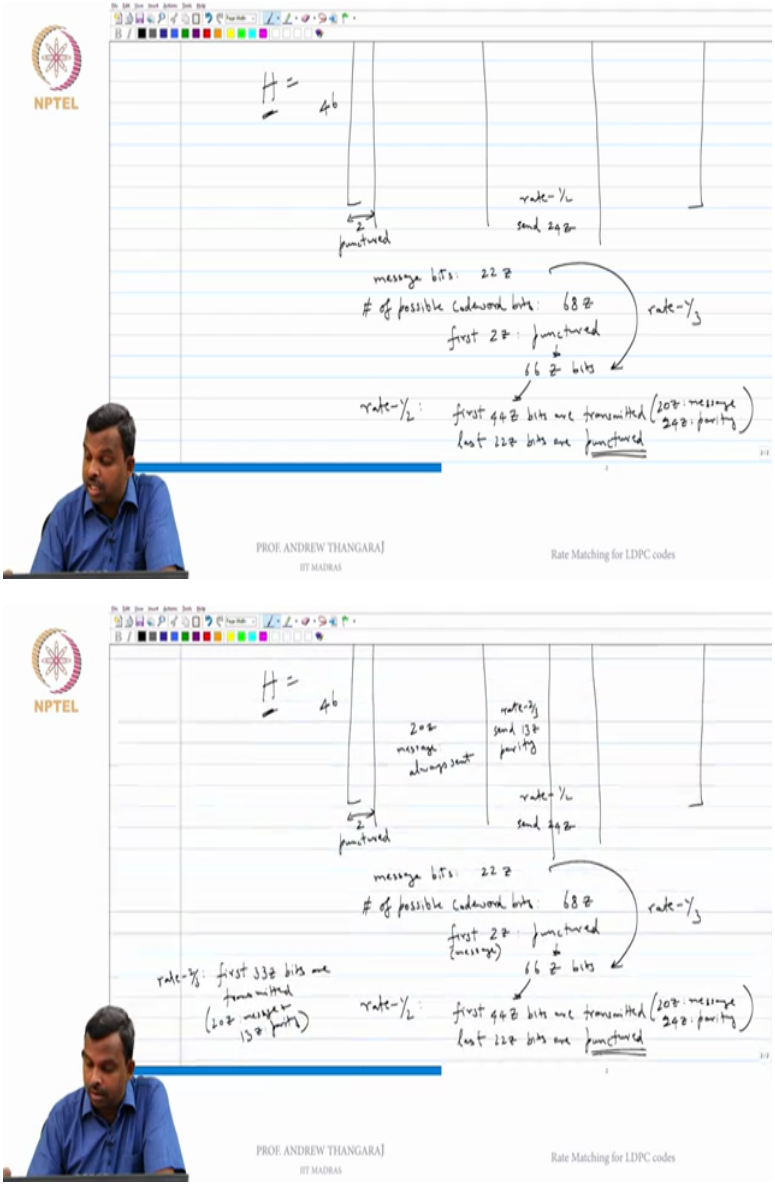
So once you fix the z, once you fix the expansion factor and once you fix the parity check matrix okay, your number of message bits becomes 22 times z okay, so now so you can take various examples for instance you can take base graph 1 which I have done here and maybe z equals 16 or z equals 24, your number of message bits is going to be 22 into 16, 22 into 24 okay.

Now the number of message bits may not be exactly that, you might be short by a little bit, so now the way the standard picks the z is to make sure that you do not have too much of a short fall and when you have a short fall it will invariable be only within one z block okay and then within that z block you make you shorten you make few of the bits to be 0 okay so that is what you do if you are not exactly a multiple of z okay.

I am not going to talk too much about that, we will not worry about that too much but what about the number of code word bits, so, so far if you look at the number of codeword bits, number of possible codeword bits, it is actually 68 into z okay so you have 22 into z you multiplying by, computing all the parities you will get 68 into z, two z, first two z are punctured okay so after the puncturing this results in 66 z bits okay.

So potentially from here to here you have a rate 1 by 3 okay so this is a lowest rate at the base graph 1 supports, the similar number for base graph 2 is 1 by 5 okay so if you remember it is 42 by 52 okay 10 z will be the number of message bits and 52 z will be the total number of codeword bits, you puncture the first two bits you get 50 z, so 10 z to 50 z is 1 by 5, so that is base graph 2 and base graph 1 you have rate 1 by 3 okay.

(Refer Slide Time: 03:10)



And so what if I want a higher rate so suppose I want rate half, what do I do? Okay so that is where the rate matching sort of comes in for rate half the message bit is 22 z you transmit only first 44 z bits are transmitted, last 22 z bits are punctured, punctured meaning see remember when you puncture those parities they are actually non zero values and you are not transmitting them okay so that is something important to remember.

So then you get rate half okay so out of these 46, so you have the first 22 and out of these 46, you will only send the next 22 okay so for rate half, so remember it is not a total of 22, you are puncturing something in the middle right, so you are sending only 20 of these things so you can do 24 okay so you send 24, okay out of the 66 total you are sending, so 2 got

punctured out of 68 so you got 66 and out of the 66, the first 20 z, 20 z are message the remaining 24 z are parity okay so remember how will you do this?

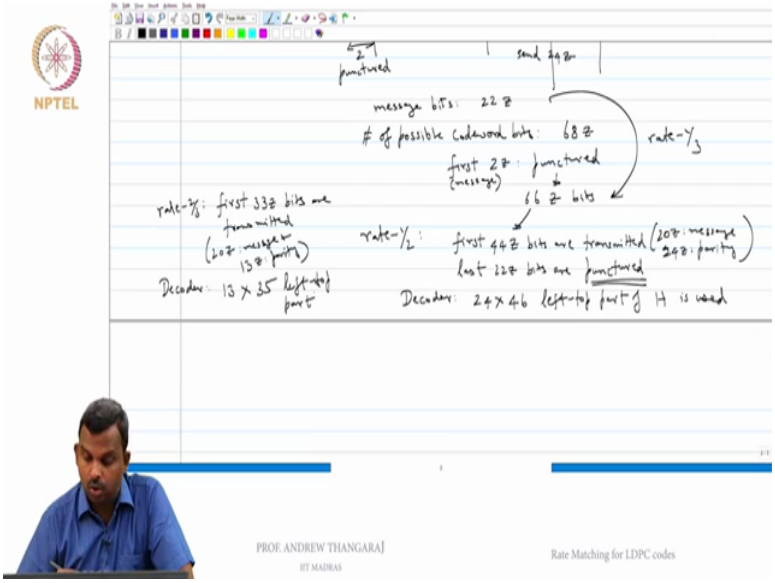
First 44 z's are transmitted, 20 z is message and 24 z is parity okay remember the first 2 z message blocks got punctured okay, so this is message and you puncture them okay, so 24 z you send to rate 1 half, what if you want a higher rate, supposing you want rate let us say 2 by 3 okay so what do you do for a 2 by 3, let me write that down here, supposing you want to write 2 by 3 from 22 z I should be only sending 33z total right.

So that will give me rate 2 by 3 bits are transmitted so that means 20 z is message plus 13 z is parity okay so for rate 2 by 3 so you will have something over here okay 20 is that message is always sent okay so 22 z is the total number of message, 20 bits of message you always send, how many parities you send depends on the rate that you want to accomplish, rate 2 by 3 you send only 13 z, rate half you send only 24 z, higher and higher rate, different rates you can decide how many bits you want to sink okay so that is the rate matching strategy from a high level.

So how do you do the decoding now, so supposing if you do rate half how should you decode, now that is an important question as well right so you have only 24 z okay so since you have only 24 z here okay the total that you are sending is 44 z okay so if you look at it there will be this block, there will be this 22, 24 by 24 block here okay let me draw like that okay so this is a 24 by 24 part of it okay so nothing to the right of this gets transmitted.

So your parity check matrix will just be this okay so the first 24 z, 24 block rows is what you will use in the parity check matrix okay so you sent 24 z parities okay which means only 24 blocks you will use in the decoder okay so only this part, only this part that I blocked out here is used in the rate half code okay.

(Refer Slide Time: 07:50)

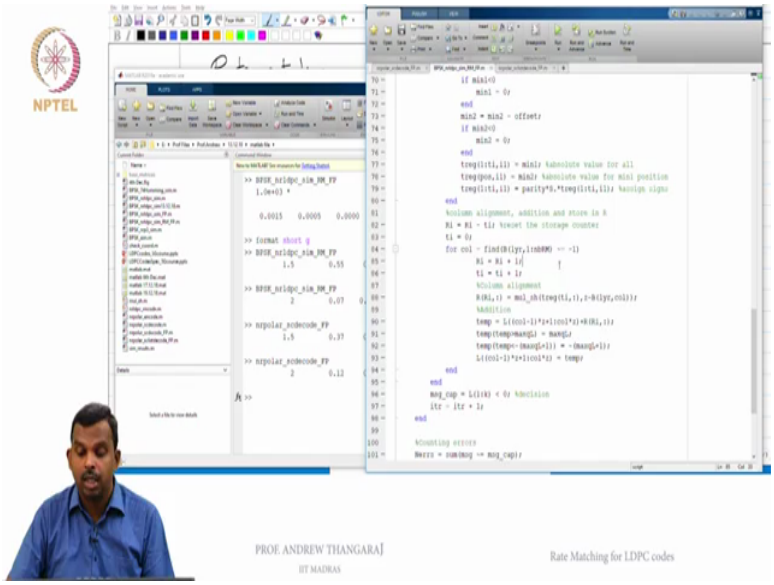


So let me complete that here, okay so in the decoder for the rate half code you use 24 by the total here is 24 plus 22 just 46, 24 by 46 left top part of h is used okay so the other part should not be using even it is not valid to use the other parts they do no work okay, so once again let me repeat you had 22 z total number of message bits, two got punctured but the punctured part you have to use in the decoder, 20 z is what you sent.

If you want a overall rate of half, you have to send totally 44 z code word bits so another 24 z you can send which means 24 block rows, okay block columns here and 24 block rows of the parity check matrix actually got used in the computation, so this is the valid parity check matrix that is one part of it okay.

So now what about the rate 2 by 3, okay in the rate 2 by 3 you using only 13 okay so you will be doing 13 by 35 okay so in the decoder you will be using 13 by 35 left top part okay so the others you should not use okay so that is something to be careful about okay so that is one part of it, so once you do this with the LDPC code you can send any rate you want not only the lowest rate of 1 by 3 you can send 1 by 2, 2 by 3, etc. okay.

(Refer Slide Time: 09:33)



So let me show you how I modified the MATLAB code to accomplish this okay, so this is what I have done here, so this is the rate matched version of the code, the first thing you will see is I have added this new thing called NBRM okay, so supposing I desire a rate of half, I have to figure out how many blocks to send okay so that is going to be KB divided by rate plus 2, 2 is the punctured position and then you have KB divided by rate okay.

So for instance for the base graph 1 KB is going to be 22, if rate is half you will have this first part giving you 22 by half which is 44 plus 2 is 46 okay so if you remember 46 is the total number of blocks that will be valid in your parity check matrix okay, so the total N is NB into RM, that is NBRM into z, okay this is after expansion and then this is the number of rows of the parity check matrix.

Number of columns part of parity check matrix, this is just expansion, I do not have to write any code for it, this is number of rows, okay and you have to subtract KB for it okay, so MBRM the rate matched version of the parity check matrix all the rows will not apply, they will not be valid parity checks you have to subtract KB from it, only MBRM minus KB will be valid.

So if you look at rate half for instance NBRM became 46 so NBRM minus 22, 46 minus 22 will be 24, so we saw just now 24 needs to come out okay so the first KB part of it needs to be subtracted from this okay, so hopefully this is clear this plus 2 and this is because of the puncturing, if we were not to puncture this plus 2 would not come okay so this is how it looks okay, so this is one change.

So now your decoder was using the entire parity check matrix and I have to make suitable changes there also, the changes here are not too bad for every iteration the for loop for the layer will run only till MBRM, it would not take the whole parity check matrix only till MBRM and the column will stop at NBRM okay, the columns here will not use the entire thing MBR, other than that nothing has to change, everything else will remain the same, the min-sum operation after you have accumulated all the rows together and align them as the same and then here again when you do the addition the final addition you stop with NBR okay so there is a change here to say MBR okay so that is it.