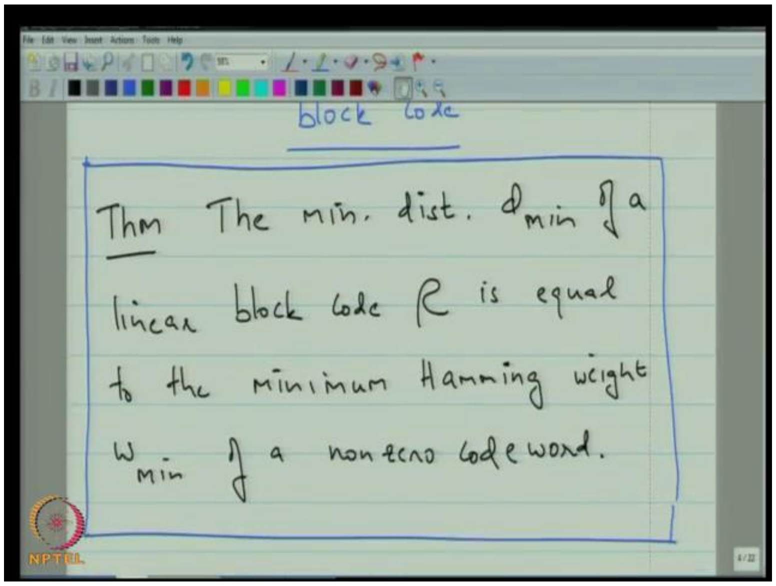
Error Correcting Codes

Dr. P. Vijay Kumar
Department of Electrical Communication and Engineering
Indian Institute of Science, Bangalore

Lecture No. # 12
Bounds on the size of a Code

Good afternoon, and welcome back; so we will begin today our twelfths lecture in this series. We just browse quickly through our last lecture, where I work was dealt with finding initially the minimum distance of a code.

(Refer Slide Time: 00:54)

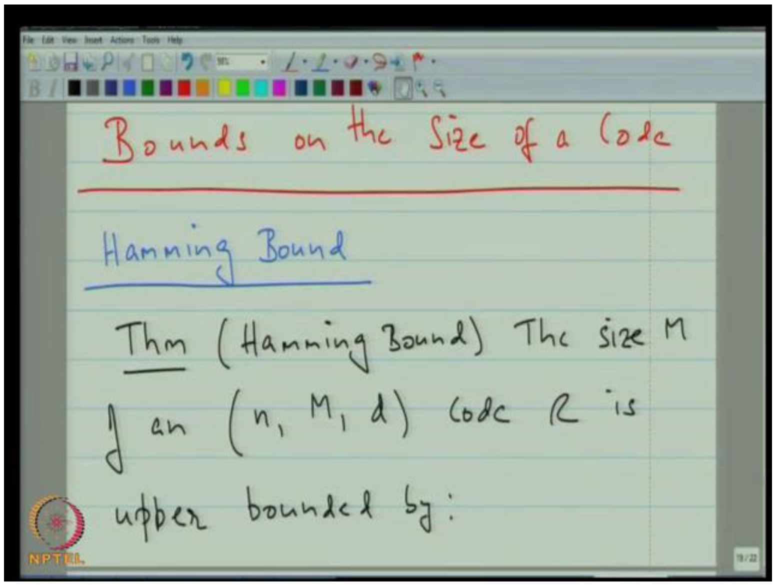


block code

Thm The min. dist. d_{min} of a linear block code $\mathcal{C}$ is equal to the minimum Hamming weight w_{min} of a non zero code word.

Let me do this, and I told you that is equal to the minimum Hamming weight; we prove that we looked at examples, and then we define an alternative means of determining the minimum distance using a parameter S, and we prove the theorem that d min S plus 1. Then we should that the Hamming code is S is 2, and this also let us to define the general Hamming code. Then we prove the Singleton bound, which is the straight forward consequence of what we have done before? And codes, that meet the Singleton bound is called MDS codes. We look at two examples, and I quickly mention that there are no other known families, and towards the end of the lecture.

(Refer Slide Time: 01:46)



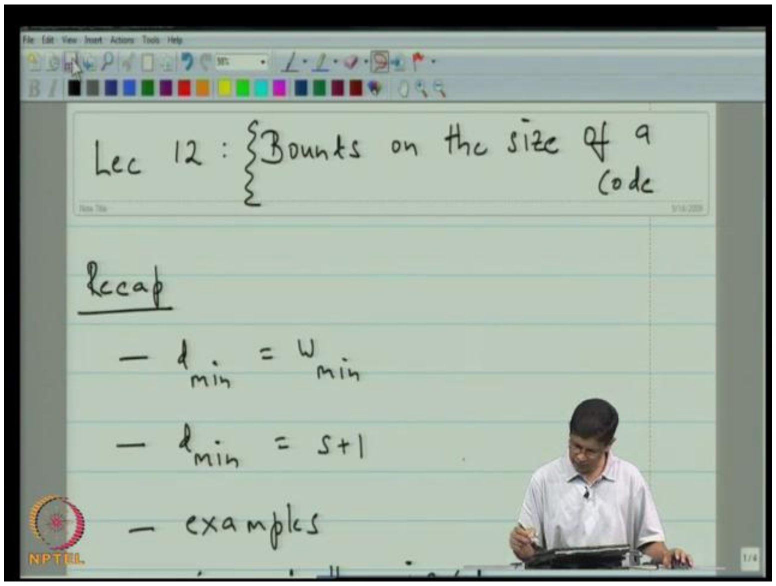
Bounds on the Size of a Code

Hamming Bound

Thm (Hamming Bound) The size M of an (n, M, d) code $\mathcal{C}$ is upper bounded by:

We started looking at bounds on the size of a code, and we started out with the Hamming bound. And what we showed is that the Hamming bound, we wrote down the expression with Hamming bound, we did not actually prove it. And I just about got started proving this, so what I am going to do is, I am going to copy this page over here on to our next lecture.

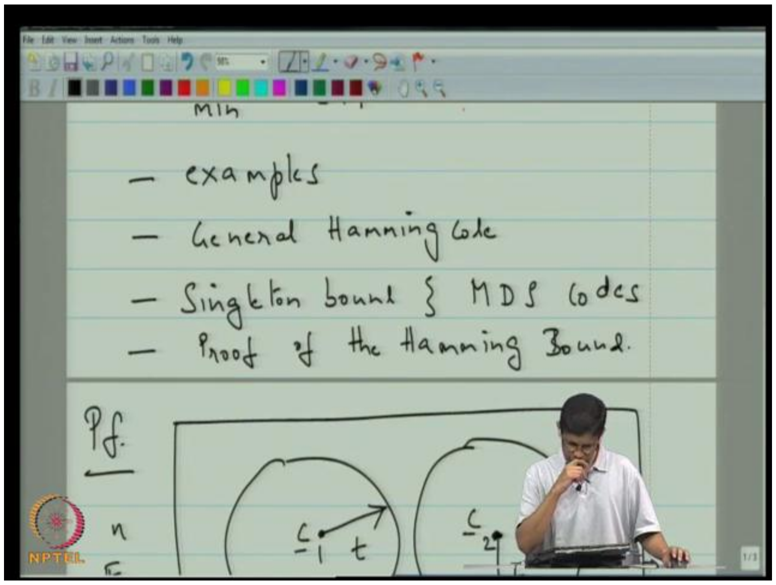
(Refer Slide Time: 02:44)



Lec 12 : Bounts on the size of a code

Recap

- $d_{min} = w_{min}$
- $d_{min} = s+1$
- examples


Min

— examples

— General Hamming Code

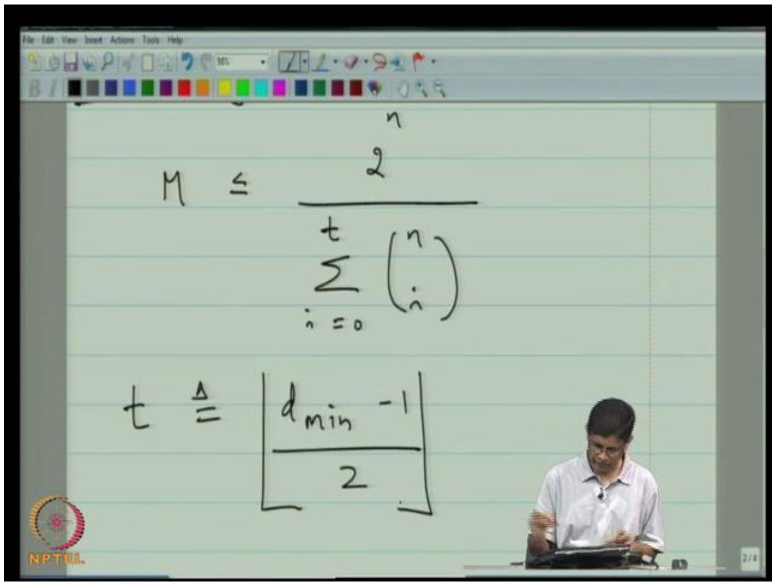
— Singleton bound & MDS Codes

— Proof of the Hamming Bound.

Pf.

I am going to this lecture twelve as I just mentioned. We are going to call this, Bounds on the size of a code; Just is a quick recap, we looked at we showed last time that d min equals w min that d min was equal to s plus 1. Let us to define that we looked at examples of course and in both case you looked at examples, and they went on to show. We talked about the general Hamming code, then we talk about the Singleton bound, and MDS codes, and then towards the end of the last lecture, we will looking at proof of the Hamming bound. So, the Hamming bound says that, we just get it of this extra page that is introduced here.

(Refer Slide Time: 05:30)

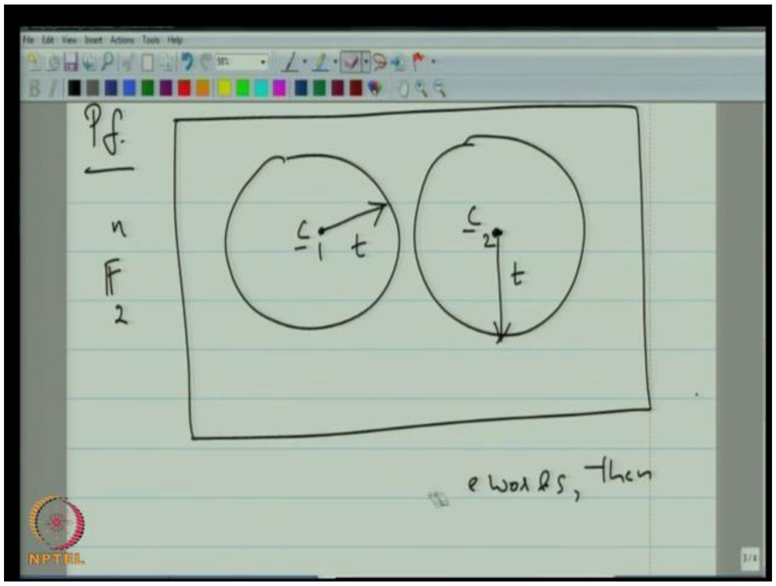


$$M \le \frac{2^n}{\sum_{i=0}^{t} \binom{n}{i}}$$

$$t \triangleq \left\lfloor \frac{d_{min} - 1}{2} \right\rfloor$$

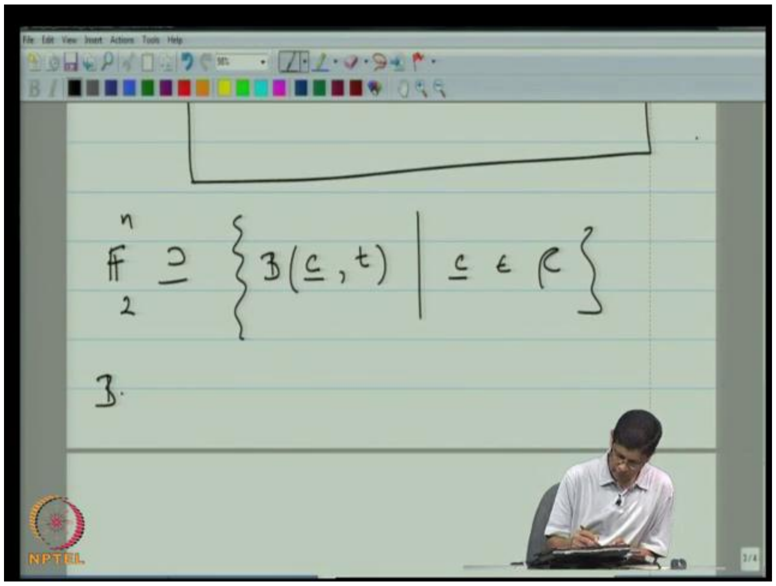
The Hamming bound said that the following that, the size of the code is less than or equal to 2 to the n divided by the sum i is equal to 0 to t n choose i, where t was defined as d min minus 1 upon 2.

(Refer Slide Time: 06:02)



We start it proving this using the following argument. Now, looking just erase this last part of the here, I am going to phrase it. If I can that is, I am going, so in the cleangenius say it out in words then after that well I write on the proof. In words put it actually saying is that if you look at the set of all n topples. Then inside that set of n topples are contain the code words, and around each code word you can actually define a ball of radius t, and these balls must also we contained in the set of all n topples, but it terms out that these balls are the destine, there is no overlap. What that means is that, if you cannot the number of code words multiply that by the number of vectors in a ball then that must be less than the total number of vectors, and that for the Hamming bound constraints. So, let us put down here.

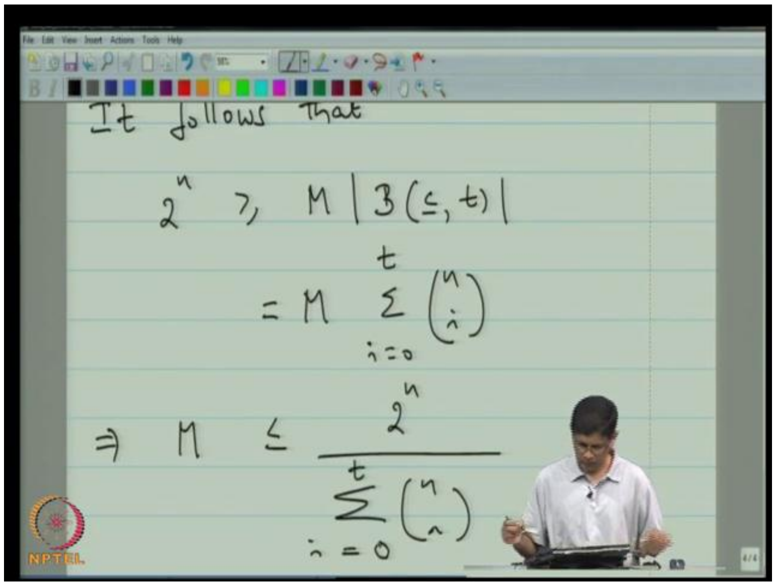
(Refer Slide Time: 07:16)



$$\mathbb{F}_2^n \supseteq \left\{ B(\underline{c}, t) \;\middle|\; \underline{c} \in \mathcal{C} \right\}$$

$B.$

(Refer Slide Time: 08:06)



It follows that

$$2^n \geq M\,|B(\underline{c}, t)|$$

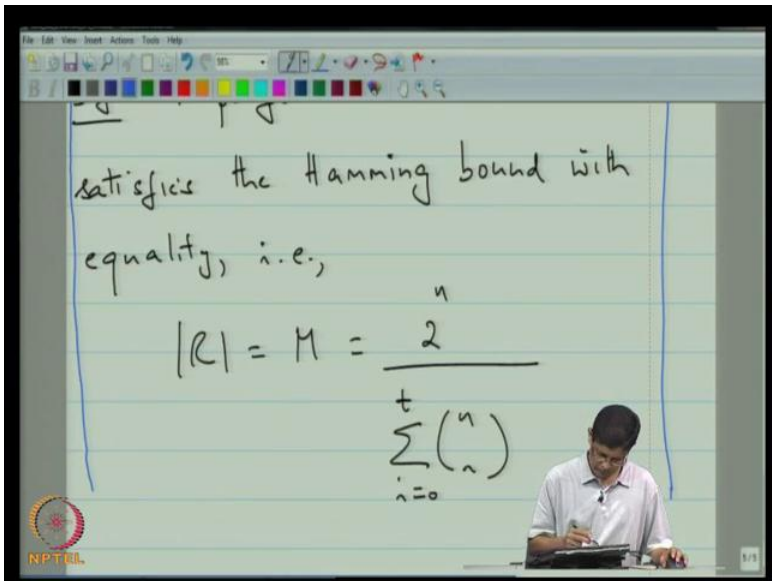
$$= M \sum_{i=0}^{t} \binom{n}{i}$$

$$\Rightarrow M \leq \frac{2^n}{\sum_{i=0}^{t} \binom{n}{i}}$$

So, F 2 to the n contains the balls of radius t, center it around all of the code words, but these balls are disjoint; meaning that B (c, t) is p. Now, we just going to count, you going to say that, since once that contains the other, the cardinality of the first set must be greater than or equal to the cardinality of the second.

It follows that 2 to the n is greater than or equal to the number of code words M times, the size of the number of vectors in a ball of radius t, which is nothing but. So from this the Hamming bound follows. That proves the theorem.

(Refer Slide Time: 09:53)

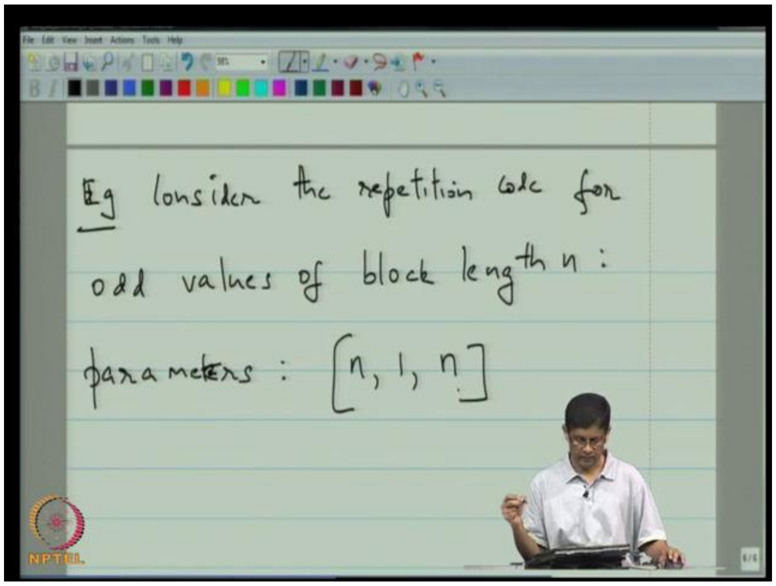


satisfies the Hamming bound with equality, i.e.,

$$|C| = M = \frac{2^n}{\sum_{i=0}^{t} \binom{n}{i}}$$

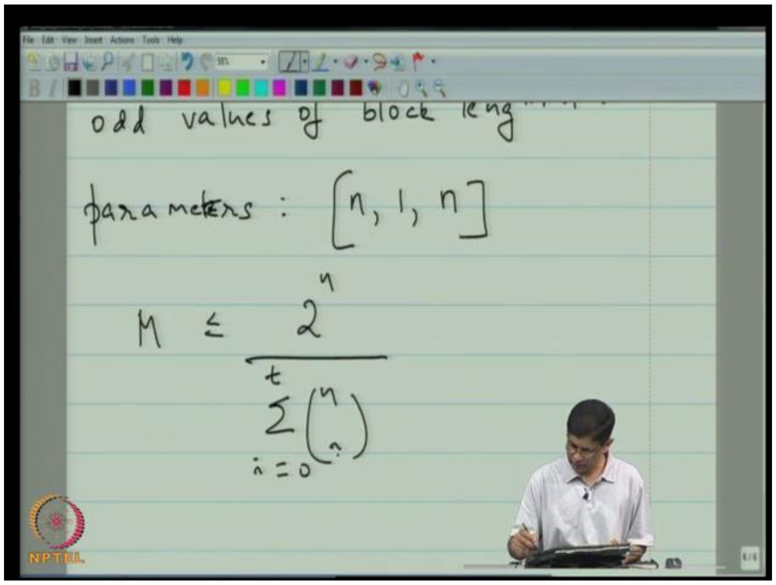
Now, let as look at examples, and see where we stand with respect to this bound, but first I need a definition a perfect code; perfect code is the code that satisfies the Hamming bound with equality, i, e; that is size of the code is equal to 2 to the n divided by the sum power i n choose i.

(Refer Slide Time: 11:43)



Eg Consider the repetition code for odd values of block length n :

parameters : $[n, 1, n]$



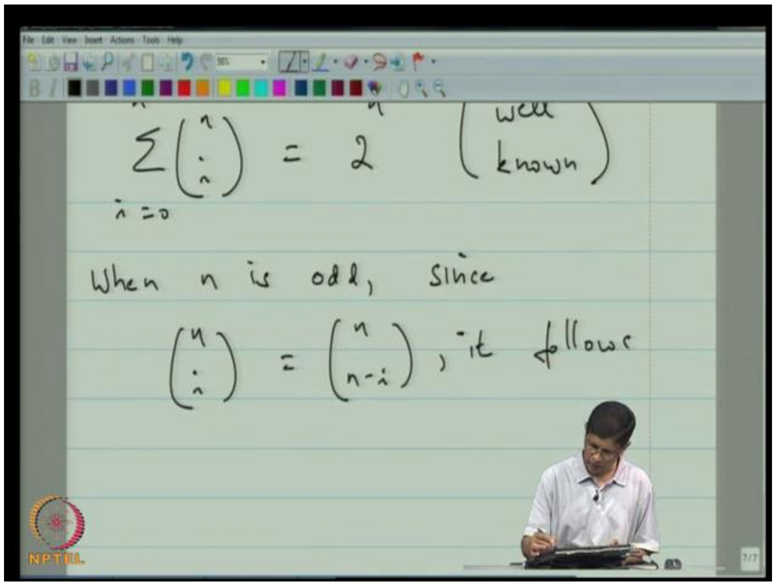
odd values of block length

parameters : $[n, 1, n]$

$$M \leq \frac{2^n}{\sum_{i=0}^{t} \binom{n}{i}}$$

Of course, a natural question to ask, is well to perfect codes exist example one, consider the repetition code for odd for odd values of block length n. Then you know that the parameters of this code are n, the dimension is one, and the minimum distance is also n. Now, what is the Hamming bound say? The Hamming bound says that M in this case is less than or equal to 2 to the n divided by the sum i goes from 0 to t n choose i.

(Refer Slide Time: 13:11)



$$\sum_{i=0}^{n} \binom{n}{i} = 2^n \quad (\text{well known})$$

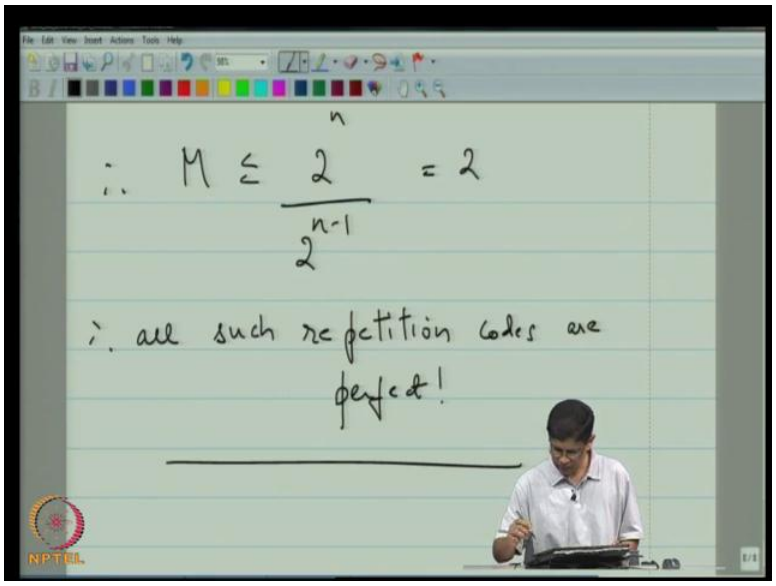
When n is odd, since

$$\binom{n}{i} = \binom{n}{n-i}, \text{ it follows}$$

Now, we know that if you add I goes from zero to n. If, you add up all the binomial coefficients you get two to the n. This is well known result in common networks or from the binomial theorem, and when n is odd. Since, n choose i is equal to n choose n minus i. It follows that, this is in fact equal to 2 to the n minus 1, because there is the Asymetrix between the binomial coefficients.

If, you sum all of them, then you get two to the n, but let say you start from left, and when you start from half way, then you will get two to the n minus 1. So, that is what let us what this saying? Now we ready to plug this in, so we want to use this, and plug it into the Hamming bound, so here is the Hamming bound, so we want to replace the denominator here, but what we just derived.

(Refer Slide Time: 15:00)

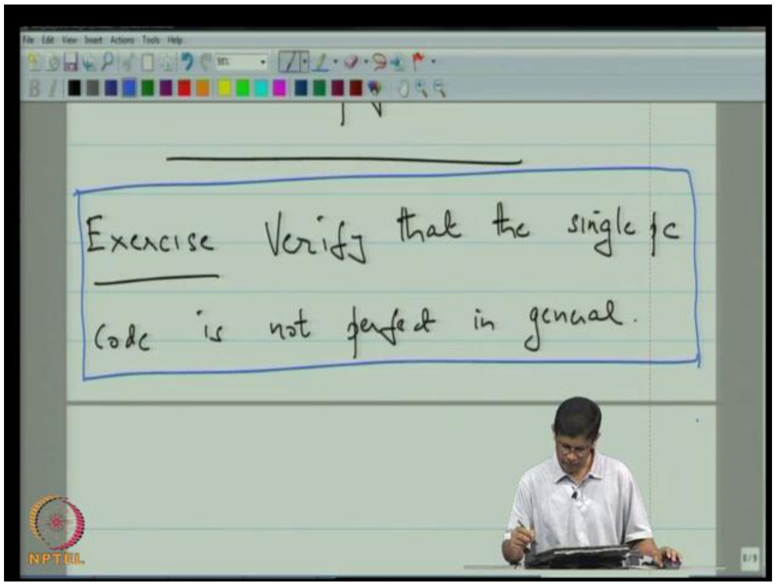


$$\therefore \; M \le \frac{2^n}{2^{n-1}} = 2$$

$\therefore$ all such repetition codes are perfect!

And therefore, M is less than or equal to 2 to the n divided by 2 to the n minus 1, which is 2; therefore all such repetition codes are in fact perfect. And actually you must be wondering well how about the single parity check code are they also perfect.

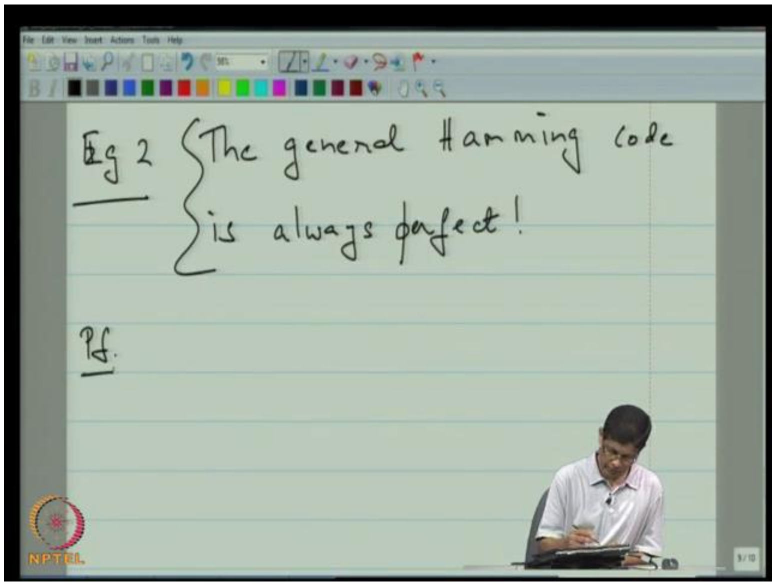
(Refer Slide Time: 15:42)



Exercise Verify that the single pc code is not perfect in general.

So, you can verify, I will leave this is in exercise verify that with the single parity check code is not perfect in general. Any other perfect codes are the situation, then we were talking about MDS codes are pointed out that the news is in some sense the bad news, because the only non MDS codes are the trivial codes, which are either the repetition code or the single parity check code. In the case of perfect codes, however the news is little bit better there is one excuse me family of codes that meets the Hamming bound the quality, and not surprisingly, this is the class of general class of Hamming codes.

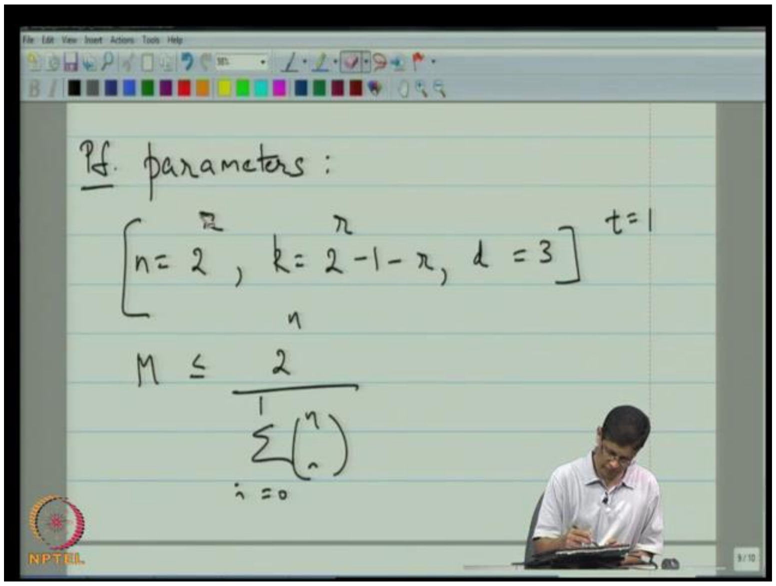
(Refer Slide Time: 17:05)



Eg 2 The general Hamming code is always perfect!

Pf.

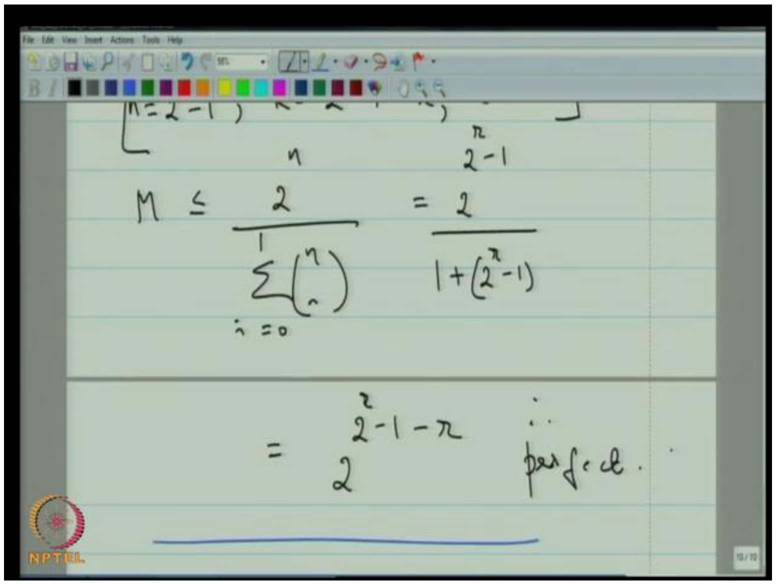
(Refer Slide Time: 17:40)



Pf. parameters :

$$\left[n = 2^{r}, \; k = 2^{r} - 1 - r, \; d = 3 \right] \quad t = 1$$

$$M \le \frac{2^{n}}{\sum_{i=0}^{1} \binom{n}{i}}$$

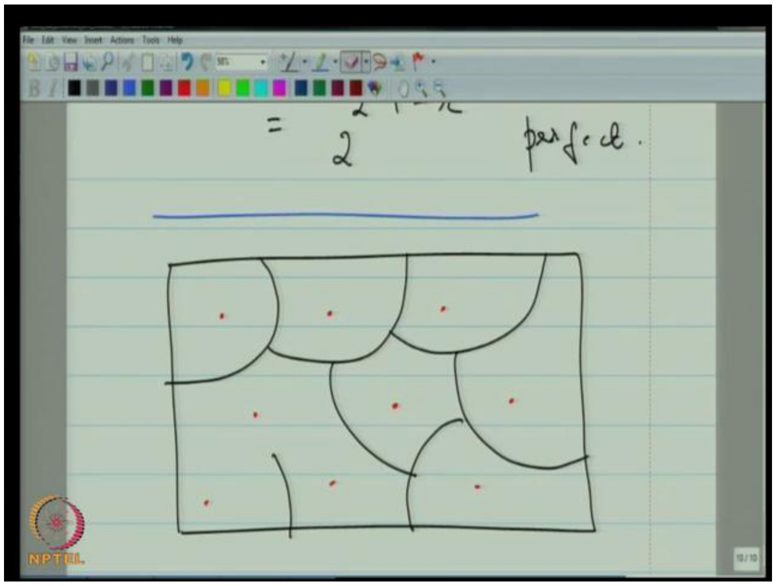


$$M \le \frac{2^n}{\sum_{i=0}^{1}\binom{n}{i}} = \frac{2^{2^r-1}}{1+(2^r-1)}$$

$$= 2^{2^r-1-r} \quad \therefore \text{ perfect.}$$

The general Hamming code is always perfect. Proof: Remember that the general Hamming code has these parameters are n equals 2 to the r, k is 2 to the r minus 1 minus r, and d or d min is 3. Now, if you check the Hamming bound m is less than or equal to 2 to the n divided by the sum i is equal to 0. Here, t is equal to 1 0 to 1 n choose i, so this becomes by the way, this is the small type of here, and is not two to the n or its n is 2 to the r minus 1. So, here apply this, then this is 2 to the r minus 1 divided by 1 plus 2 to the r minus 1, which reduces to 2 to the 2 to the r minus 1 minus r, which is precisely the dimension of the code, therefore it is perfect.

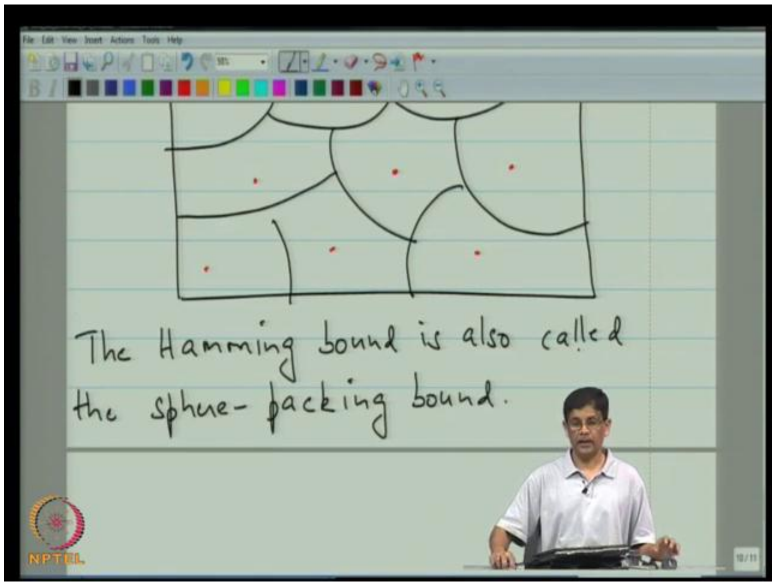
(Refer Slide Time: 19:30)



$$= 2 \quad \text{perfect.}$$

I should also to say that, the hamming code is also known as the sheer packing bound, because you can think of a picture like this. Where you have the code words, and you have these balls of radius sheers of hamming t of radius t, and these perfectly cover the space. It packs the space in the sense that there are no vectors left over, where is apart from the code words, and the balls surrounding them. There are no other vectors these no empty space, so it is speak; for this reason is also called first we a packing bound.

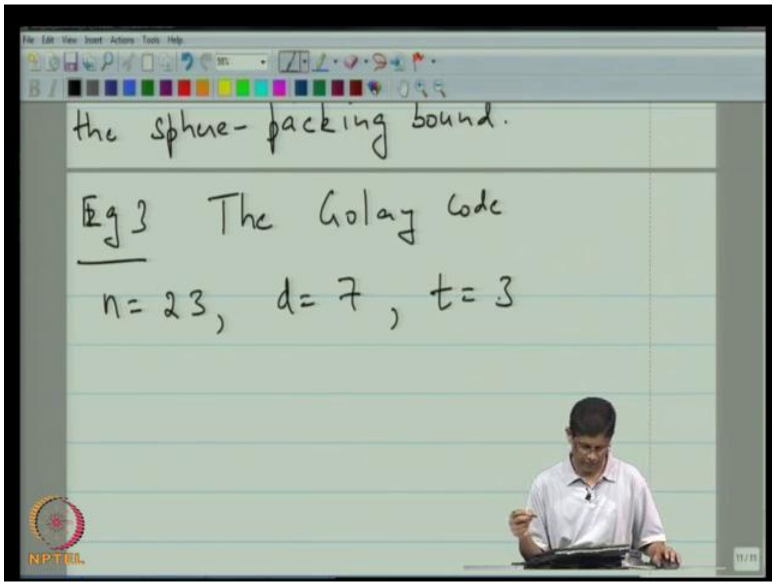
(Refer Slide Time: 20:27)



The Hamming bound is also called the sphere-packing bound.

We have seen that our repetition codes of odd length are perfect, when every Hamming code is perfect, at the any other perfect codes. In the story here is interesting, there was this is the coding theorist marshal go lay, who actually was perusing the same question, and he made the following observation is at look. Let as assume that, there is a linear code which is perfect. Then he looked at the Hamming bound, and we made a certain observation. He notices that, the sum of certain binomial coefficients had to equal a power of two, and he said this is new and then this gets into numerology, because what you are saying now is… I am just going to look at numbers in see, if I can get the sum of certain consecutive binomial coefficients to equal a part of 2. Once I find that, then I actually start looking for perfect codes, it seems like a fare best idea, but actually it worked, so here is what he did many years ago?

(Refer Slide Time: 22:16)



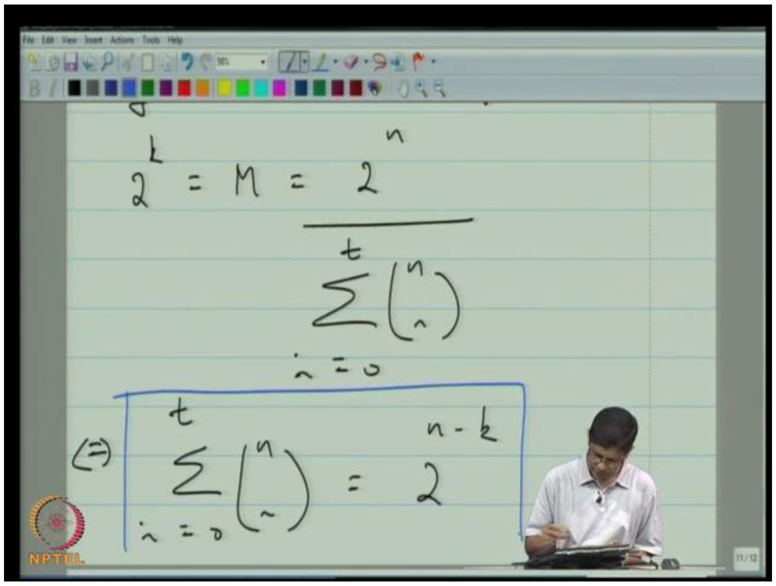
the sphere-packing bound.

Eg 3 The Golay code

$n = 23, \quad d = 7, \quad t = 3$

This class of code is called the Golay code, and there is from the following calculation supposing you take n to be 23, and you take d to be 7, in which case t is 3. So, Golay's observation was that so perhaps, the since this is in a side a let me a do one thing, I am going to a let me introduce align a certain page here, we will come back to Golay code.

(Refer Slide Time: 23:29)



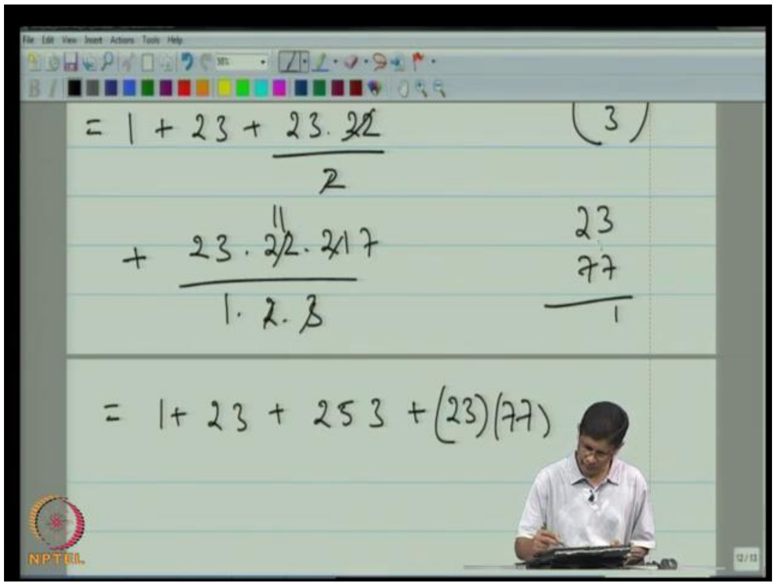
$$2^k = M = \frac{2^n}{\sum_{i=0}^{t} \binom{n}{i}}$$

$$\Leftrightarrow \boxed{\sum_{i=0}^{t} \binom{n}{i} = 2^{n-k}}$$

Let as put down Golay's observation. First, if a linear code is perfect then you must have that the size of the code must equal 2 to the n are another words, the sum i is equal to 0 to t n choose i must equal 2 to the n minus k. This is what our telling you about earlier, there is observation were basically that if you to the sum of certain the first few binomial coefficients. Then you must end of pair of two, and these all precisely the binomial coefficients in equation, it is called this equation 1. And his search for the suppose to searching for perfect code to actually look a different values of n, and then try to see when it is true that you can actually get a power of two.

And in terms or that it is not very often that this happens. Wherever, certain specific cases in which does happen, and then he was able to rule out the existence of perfect code, and some of those thus any codes 23 was a case in which he could not rule out the existence. So, then they said about constructing a code, and I think he was a he was a brilliant coding here is and he did manage to come up a back this many years ago, so many years ago with the perfect code.

(Refer Slide Time: 25:50)

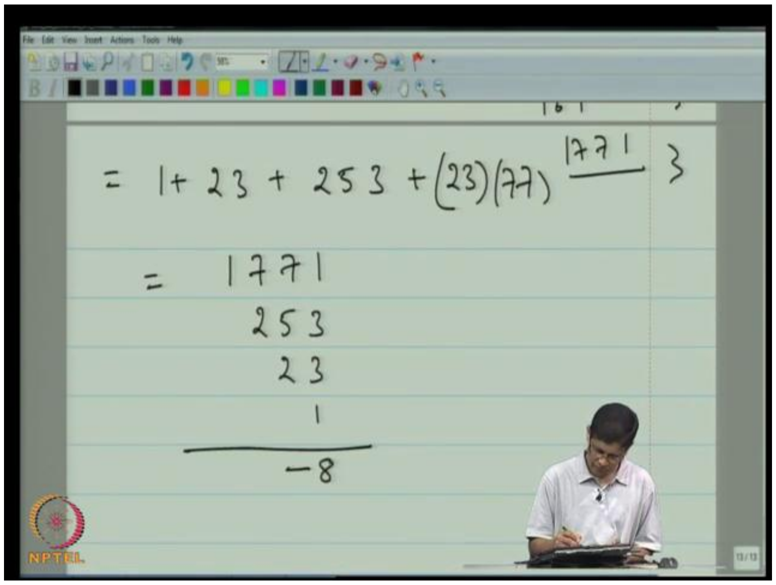


$$= 1 + 23 + \frac{23 \cdot \cancel{22}}{\cancel{2}}$$

$$+ \frac{23 \cdot \overset{11}{\cancel{22}} \cdot \cancel{21}7}{1 \cdot \cancel{2} \cdot \cancel{3}}$$

$$\begin{array}{r} 23 \\ 77 \\ \hline 1 \end{array}$$

$$= 1 + 23 + 253 + (23)(77)$$

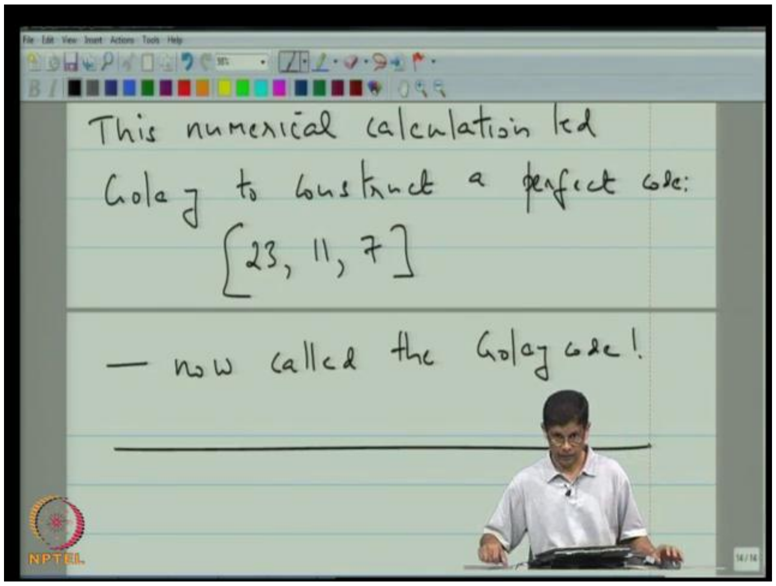


$= 1 + 23 + 253 + \underbrace{(23)(77)}\ \overline{1771}\ 3$

$$\begin{array}{r} = \quad 1771 \\ 253 \\ 23 \\ 1 \\ \hline -8 \end{array}$$

Equation one in this case, will actually read like, so will ask as to compute the sum 23 choose i for i going from 0 to 3, which is 23 choose 0 plus 23 choose 1 plus 23 choose 2 plus 23 choose 3. Now, this is 1 plus 23 plus 23 into 22 divided by 2 plus 23 into 23 into 21 divided by 1 into 2 into 3. This is 1 plus 23 plus 253 from multiplying 11 times 23 plus 23 times 77. Let us do that computation on the side. So, this is, therefore 1771 plus 253 plus 23 plus 1; heard it you get 8 14 0 2, and this is precisely 2 to the 11. That was it discovery, and that let him to a consider the possibility that are code which such parameters exist, so you want to go any further into this, so this let him.

(Refer Slide Time: 27:32)



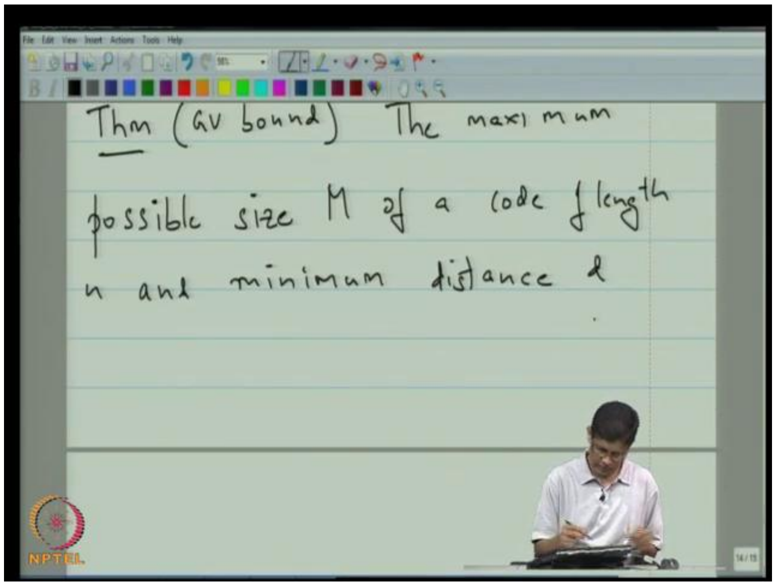
This numerical calculation led Golay to construct a perfect code:

$[23, 11, 7]$

— now called the Golay code!

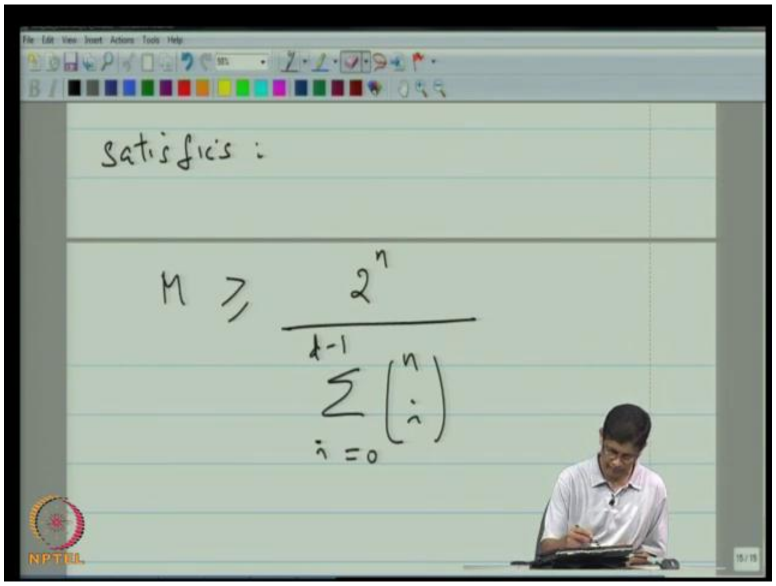
This numerical calculation led Golay to construct a perfect code of the parameters of this code by 23, the dimension is 11, and the min distance is Hamming. This code is now called the go lay code not surprisingly. Now, this may actually, led you to ask to keep question a little bit for the answer. Now, I know that the repetition code fall link the perfect, I know that every Hamming code is perfect, and I have just I have just seen the parameters of a Goley code or the any others, and it terms out that there are no other perfect codes that are known; all though that took some time to prove when involved several coding theory is, including a finish coding theory is (( )), so that took a longer to proof; that these are all the known, these are all the only possible perfect codes. There are perfect codes other than this, but they are those a turnery, and they are also called Goley codes to distinguish between you call them the turnery Golay code, but apart from this no perfect codes or known as upper bounds on the size of code.

(Refer Slide Time: 30:25)



Thm (GV bound) The maximum possible size M of a code of length n and minimum distance d

(Refer Slide Time: 31:40)



satisfies :

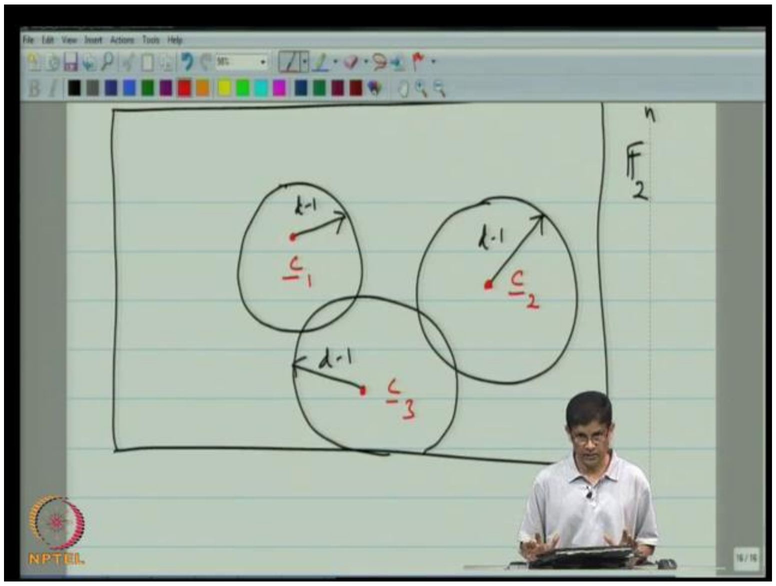
$$M \geq \frac{2^n}{\sum_{i=0}^{d-1} \binom{n}{i}}$$

So, now let us turn or a tension to a lower bound, and this particularly low bound is called a Gilbert Varshanov low bound. I think it is words noting the bounds that, we have stating whole for an any code regardless of that are linear or not necessarily linear. However, we examples there are known are linear codes which are interesting theorem, and will abbreviate Gilbert varshanov, and just write GV and the GV bound? The code size perhaps, I need to mean for little bit the error, the maximum the maximum possible size M of a binary code of length n, and minimum distance d satisfies the equation M is (( )) I need a low bound.

I need to any quality to go in the opposite direction, the M is greater than or equal to 2 to the n divided by the sum I goes from zero, and at this point you must be thinking this looks just like the Hamming bound, well it does with there are two difference, one is of course there is a low bound not in upper bound, and second leaving where as earlier. In the case of the Hamming bound you would have t here, 0 to t you have 0 to d minus 1. If, these are the two differences, how do you prove this?

Proof is via a greedy algorithm for code construction. It proceeds like this what we do is, we take a code word, so we building up a code by picking code word one at a time. And of course we have to keep in mind that our code must have block length n, and minimum distance d, we have to keep this parameter d in minus we go along, so you begin like this, take any vector at random this is pick any vector here.

(Refer Slide Time: 33:52)

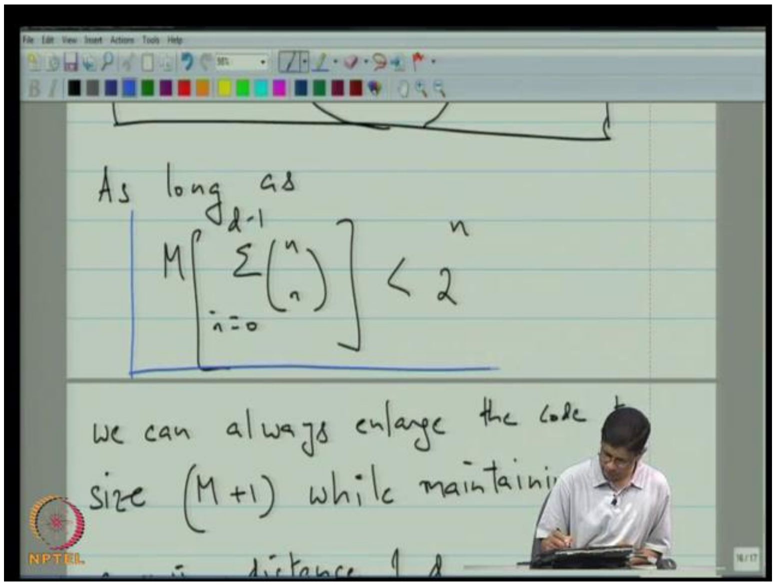


Let us, make that our first code word now, because you want minimum distance d in your code a force you cannot pick another code words which belongs to a ball of radius d minus 1, around this code word you say that is fine, you draw this ball, and the radius of the ball is d minus 1. And so when you pick a next code word you say, I am as avoid this ball F in fact what you do is from the set of all n topples, which is F 2 n you throughout all the vectors in this ball, after picking a first code vector and you throughout the code vector two, you do not want to pick the same code vector twice.

Then you pick another vector. So, let say you pick a next code vector and there is somewhere here. There is two similarly, this a ball of radius d minus 1 surrounding that, so you pick the code word and throughout all the vectors in that ball, and then you pick a next a vector in that could be c 3 and again what you do as you throughout all the vectors in a balls in a radius d minus one. Now, there are a couple of circle point about this algorithm; one is that now you see this intersection of circles I mean your bother sees wait a minute, you are not suppose have to intersection you have doing something wrong. This nothing wrong with this, because these balls can have intersection is just that note to code words must be the length Hamming distance d of each other. What you are doing is that, whenever you pick a code word you are always going to through away all the vectors which line a ball of radius d minus 1 surrounding that code word, now some of these may already have been thrown away.

Because if there was intersection, then these a balls in these intersections would already have been thrown away do not need to throw them, away twice is keep there in mind, but certainly regard less of that.

(Refer Slide Time: 36:32)



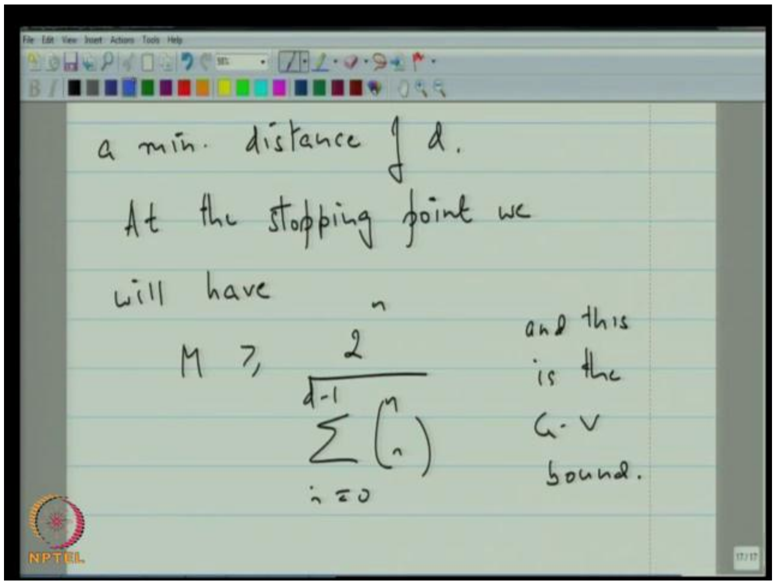
As long as

$$\boxed{M\left[\sum_{i=0}^{d-1}\binom{n}{i}\right] < 2^{n}}$$

we can always enlarge the code

size $(M+1)$ while maintaini

distance d



a min. distance of d.

At the stopping point we will have

$$M \geq \frac{2^n}{\sum_{i=0}^{d-1} \binom{n}{i}}$$

and this is the G.V bound.

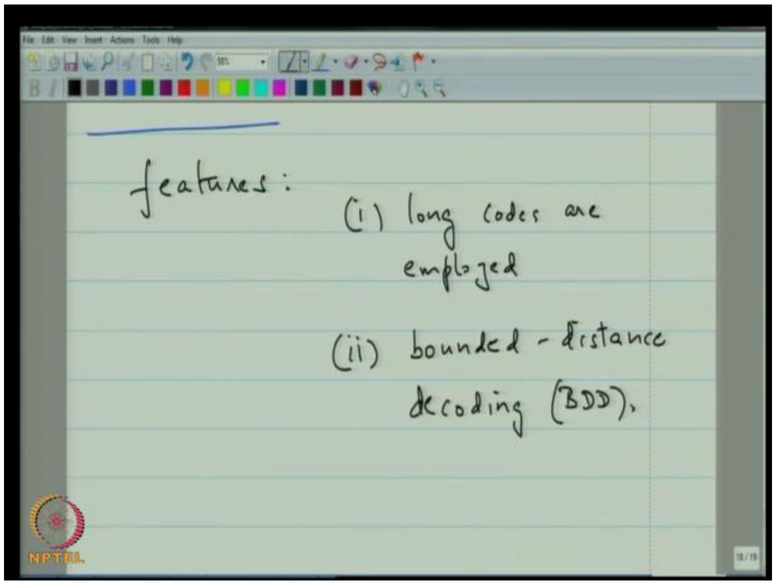
It certainly true that as long as M times the sum in choose i to 0 minus 1, as long as this is less than two to the n. We can we can always enlarge the code to size M plus 1, while maintaining a minimum distance of d. So, again just in case that is not clear, let us go back here what we have saying is that it may be that, you thrown out the less than sum in choose an I throw to d minus 1, but we will assume the worst case will actually (( )) I can assume that, you thrown out that you have to thrown out all of them.

Even if, you did that and still the number of vectors is less than 2 to the n; that means that some where line in this box, there is another code word that you can take. So, you can continue this process. Now, this part you might want just to pay get attention make carefully, so the argument is that as long as this equation, so may be, I will call thus equation 2, call this equation 2. As around this equation two is satisfied, you can enlarge the code. Now at some point equation two will fail to be satisfy, and that is when you cannot enlarge the code any further. However, you will your (( )) to reach a point, where this equation fails. So, we will just simply say at this trapping point, where this process cannot be continued at the stopping point we will have M greater than or equal to 2 to the n divided by the sum, and this is Gilbert Varshanov bound.

I am referring to this is the greedy bound. Greedy, I would interpret in the following away. There is that what you do this you do not look at long term consequence, you only look at short term consequences. Once you picked the code word, then your overriding concern is that the next code word, you pick must be it is this d y from that is all that you occupies a might, so given that you pick unique codeword and so on it is, the short term view point algorithm and it produces a code of this guarantee size.

Now have given you the Hamming bound, and the Gilbert Varshanov. This is the Gilbert varshanov bound, and very early we looked at the Hamming bound. Now, what will do next is that law examine the same two bounds expect that we will look at them as of n very, very very big, we look at very long codes, that is going to let some calculations. Now, you will ask now, where are you interested in long codes, I mean a from will short code good enough, and I will a try to interest you in long codes by giving you a long code approach to achieving reliable communication. I will explain, would I will mean by that.

(Refer Slide Time: 41:26)

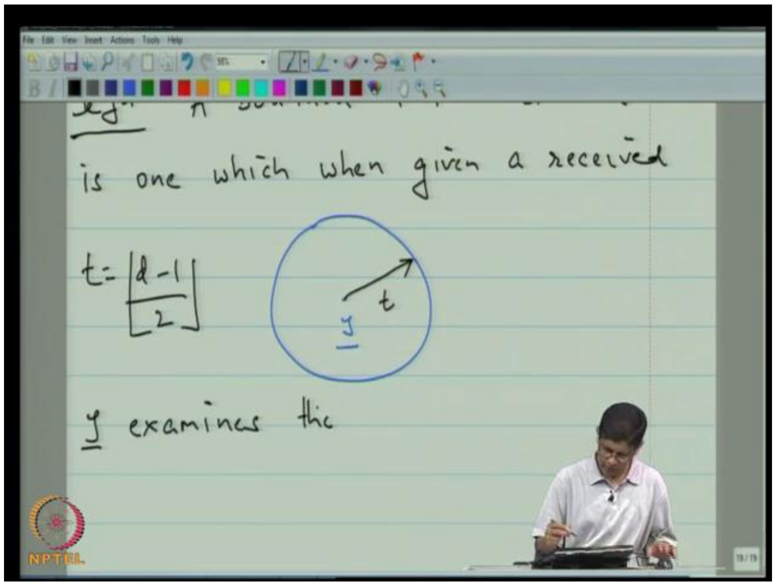

features:

(i) long codes are employed

(ii) bounded - distance decoding (BDD).

So, I call these an approach to attaining reliable communication. Now, this approach has two features, one is it employees let me perhaps let me write down put down features here. So, features one long codes are employed, long codes meaning codes of long block length, second you assume bounded distance decoding.

(Refer Slide Time: 43:23)

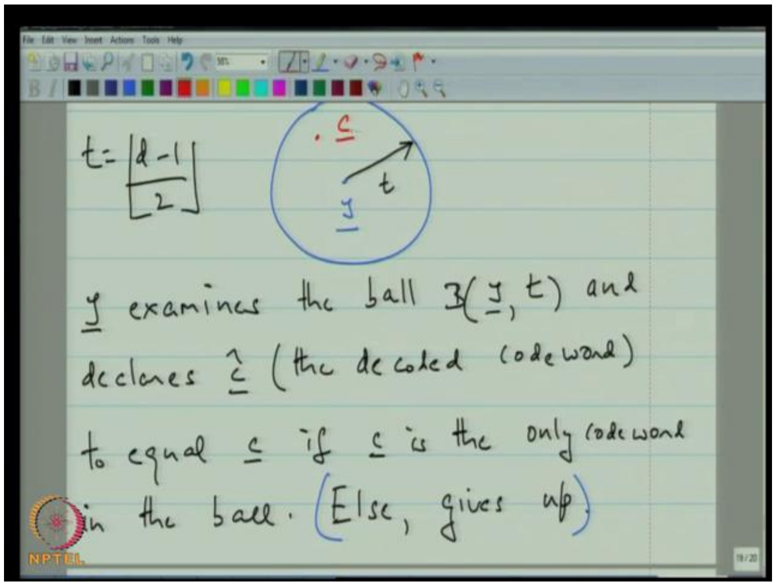


is one which when given a received

$t = \left\lfloor \frac{d-1}{2} \right\rfloor$

$\underline{y}$ examines the

(Refer Slide Time: 44:13)



$t = \left\lfloor \frac{d-1}{2} \right\rfloor$

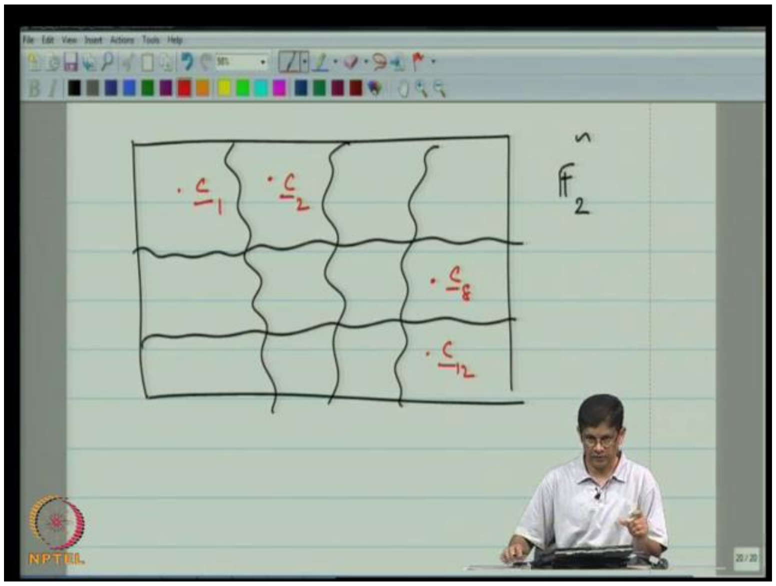
$\underline{y}$ examines the ball $B(\underline{y}, t)$ and declares $\hat{\underline{c}}$ (the decoded codeword) to equal $\underline{c}$ if $\underline{c}$ is the only codeword in the ball. (Else, gives up)

The second point here, let us put down a formal definition of this bounded distance decoding definition a bounded distance decoder is one which, when given a received of some also going to draw picture here in the middle of this definition.

Just to make things clear a and t is going to be d minus 1 upon 2 has always, when given a received vector y examines the ball B y t, which have shown above and declares c hat which is the decoded code word. If, c c hat declare chat to equal c, if c is the only code word, in the ball else gives up that gives up else gives up is parentheses is not very interesting, but what is interesting, as what it does length it finds out that within this ball, there is the single code vector c.

Now of course, if there are if there are two code words, then course if bound know what to do? Sorry actually (( )), it is not possible that there are two code words that simply not possible, because then the Hamming distance between the two code words should the no larger than 2 t, which is less than the minimum distance. So, what could happen is they could be no code word within this ball in which case just gives up. So, such a decoder is called a bounded distance decoder, now you might say well, what else is there and I mean what are other kinds of decoder are there, these terms for the logical to me.

(Refer Slide Time: 47:00)

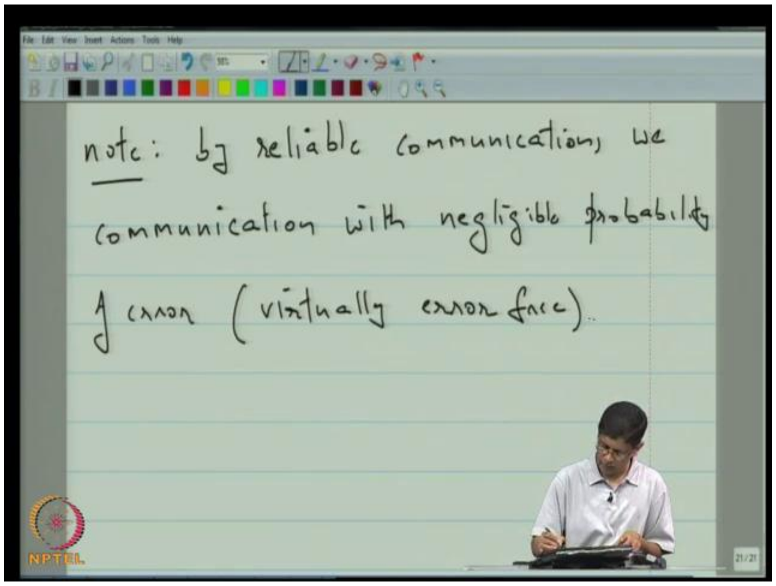


So, what in alternative decoder might do this, and will actually be coming to this soon, an alternative would be to do the following instead of what it my do, is it my just partition the set of all centuples. Let us say that this is F 2 to the n into regions, and say that look if the received vector is the here. I am going to decode c 1, and on it is going to partition the region, and then to every region it is like a tertiary, and if you are received vector.

Let us, say happens to fall if here then you simply this side decode c it these decision reasons are based on Hamming distance as you might expect, but the difference is that earlier, you would only confine yourself long in a ball of radius t around the received vector whereas here, you are actually saying look and just going to look at, for the nearest codeword. I am going to decode toward in way it is a simple of philosophy and as will see the better one, but the bounded distance decode a decoding applies in many situations for reasons.

I think which will become clearer a little later, if we are clear about what bounded distance decoded as they are going to now. Therefore, long at in approach to making communication reliable by using long codes and applying this bounded distance decoding algorithm. Now, this word reliable what do you mean by reliable these actually.

(Refer Slide Time: 49:00)



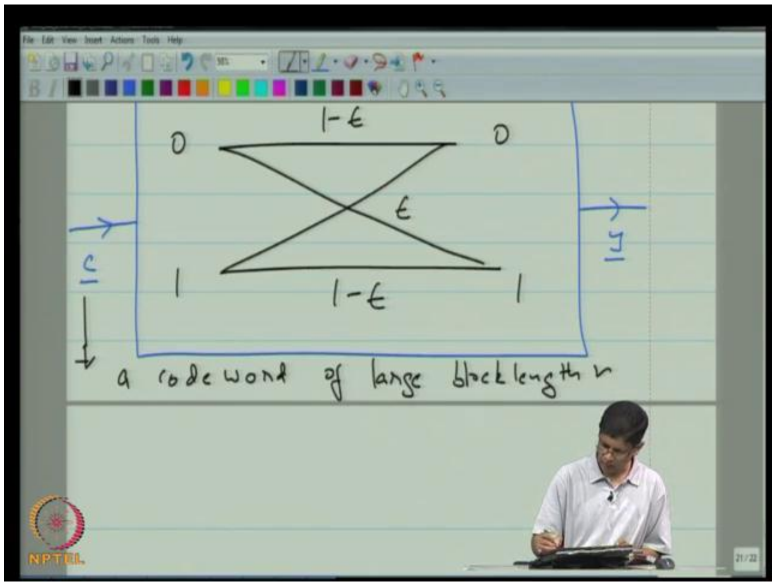
note: by reliable communication we communication with negligible probability of error (virtually error free).

So, what we mean by reliable communication? Is communication in which the probability of error is negligible? So, note by reliable communication you mean communication with negligible probability of error. It can also think of as virtually being error free. It is important, we are not satisfied with making the probability of errors small, they actually want to make it negligible, and it is not easy to do that, because in the reason why this is of interested.

Because tells that there is possible, what is says as that look, if you are trying to communication across of communication channel? It always be the case that, the channel is not perfect it is going to introduce some distortion and is as the distortion on the channel does not prevent, you from achieving reliable communication everything.

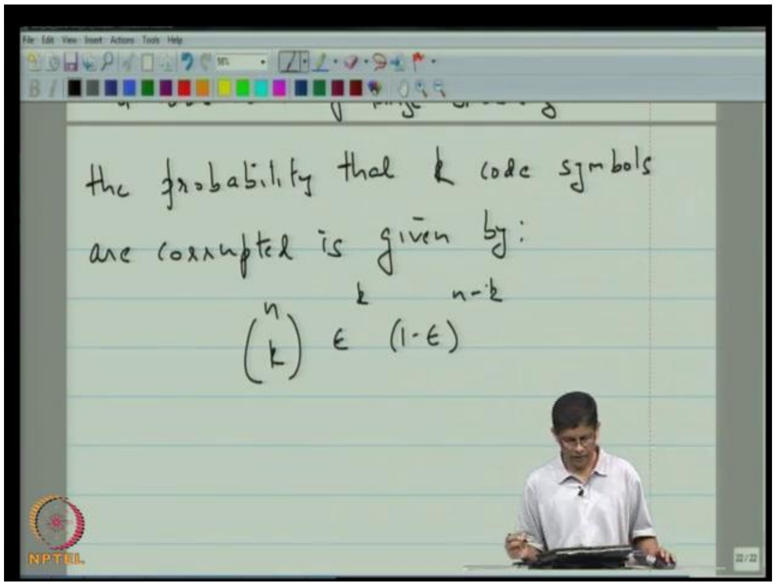
Then it will actually does is that it limits the maximum rate at which you can communication, what done is their approach. So, we start of by saying we are when approach towards retaining a reliable communication. It is going to use long codes, and that is where interested in the Hamming bounds for larger n, so your approach is this.

(Refer Slide Time: 51:24)



We going to say that consider the binary symmetric channel when put 0 or 1 put is 0 1, and the cross over probability is epsilon when and what we are going to do is across this channel across this channel, were going to sent a codeword and what is going to come out of this is the received vector y. Now, this code has large block length n, we going to send therefore n symbols across the channel, and the binary symmetric channel is going to corrupt some of those codes words.

(Refer Slide Time: 52:50)


the probability that k code symbols are corrupted is given by:

$$\binom{n}{k} \epsilon^{k} (1-\epsilon)^{n-k}$$

The probability that k code symbols are corrupted is given by n choose k epsilon to the k 1minus epsilon to the n minus k, and we know that. Now, this comes from the binomial distribution. It is well known that when n is large, this distribution tends towards the Gaussian distribution tends to become Gaussian and Gaussian with parameters. Let us add that here with parameters, mean equals n times epsilon, and the way if the standard deviation at which is equal the root n epsilon 1 minus epsilon, where only about a minute left just to summarize, what we have looked at is, we look at bounds on these size of a code, we look at the Hamming bound then we talked about perfect codes, we talked about the Hamming code the Goley code, then we look at the Gilbert varshanov bound. And I am looking at explain try to explain y, z that long codes are of interest from the point of view, it retaining the reliable communication. So, we will pick on this, and on the next lecture, thank you.