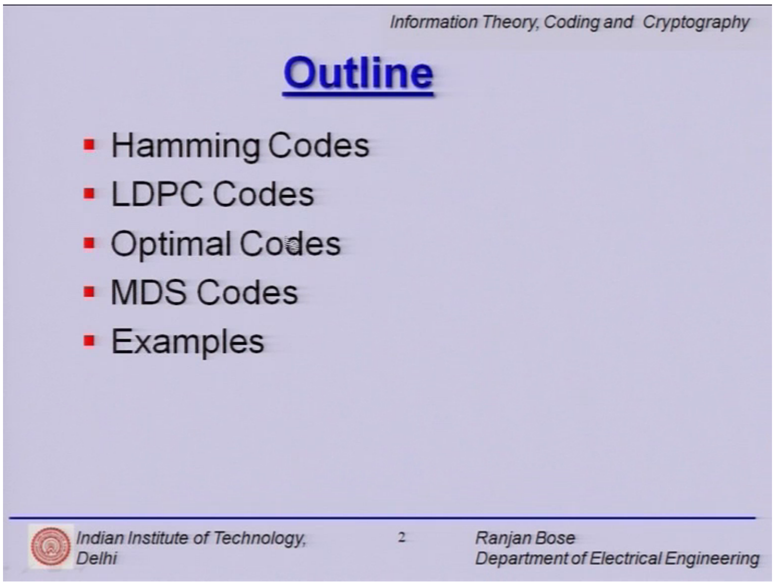
Information Theory, Coding and Cryptography
Dr. Ranjan Bose
Department of Electrical Engineering
Indian Institute of Technology, Delhi

Module - 19
Linear Block Codes
Lecture - 19

Hello, and welcome to our next lecture on Linear Block Codes. Let us start with a brief outline for today's talk.

(Refer Slide Time: 00:38)



Information Theory, Coding and Cryptography

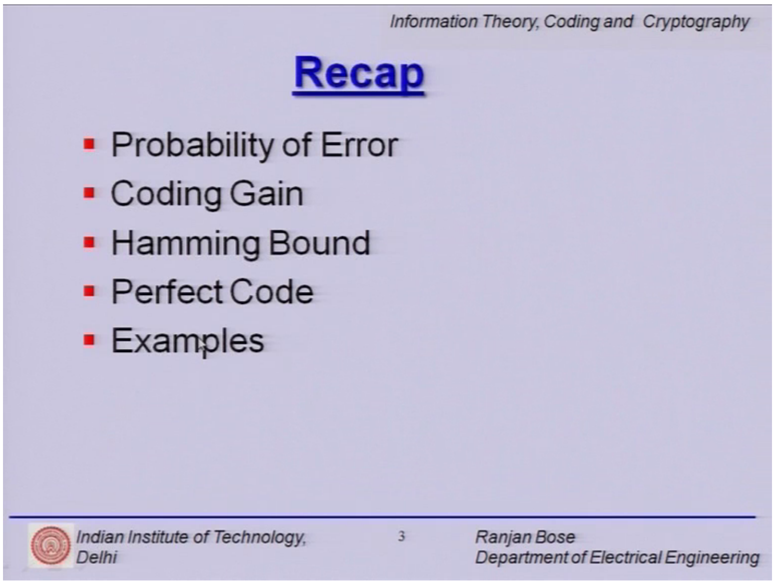
Outline

- Hamming Codes
- LDPC Codes
- Optimal Codes
- MDS Codes
- Examples

Indian Institute of Technology, Delhi 2 *Ranjan Bose, Department of Electrical Engineering*

We will today study Hamming codes followed by low density parity check LDPC codes then we will look at the notion of optimal codes. And finally, maximal distance separable codes also called as MDS codes we will top it up with some examples. So, that is the brief outline for today's talk.

(Refer Slide Time: 00:56)



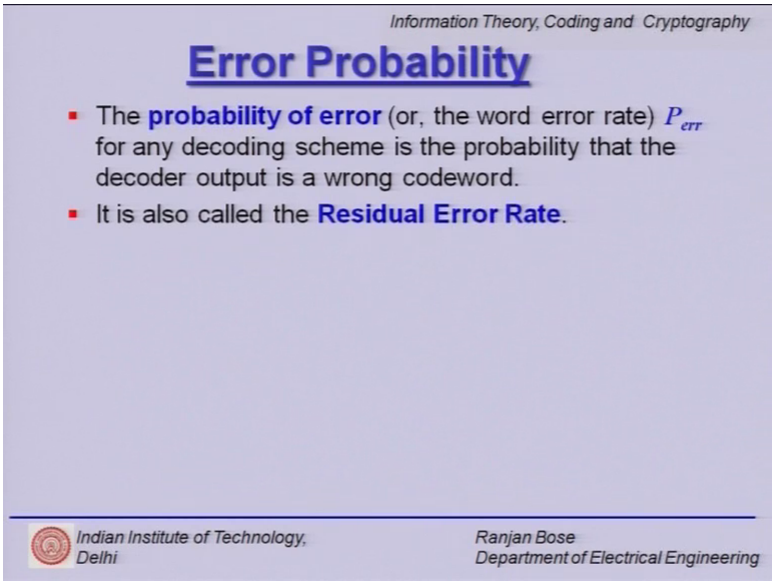
Information Theory, Coding and Cryptography

Recap

- Probability of Error
- Coding Gain
- Hamming Bound
- Perfect Code
- Examples

Indian Institute of Technology, Delhi 3 Ranjan Bose Department of Electrical Engineering

(Refer Slide Time: 00:58)



Information Theory, Coding and Cryptography

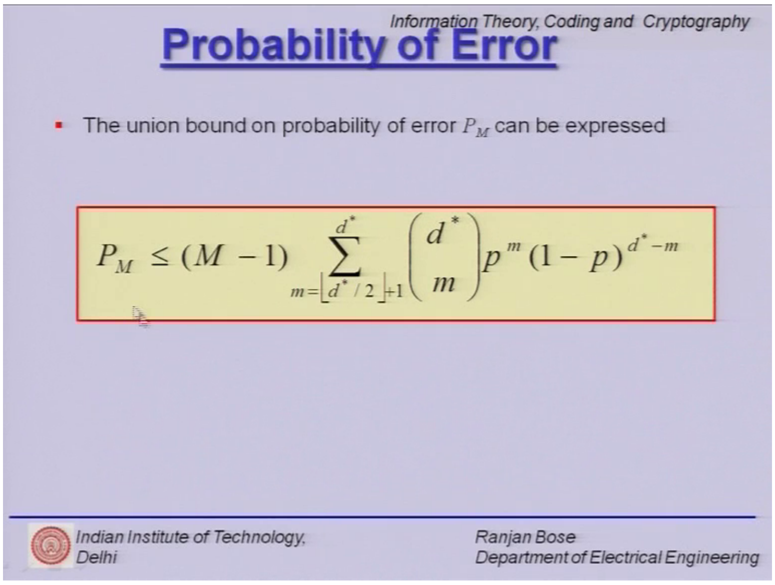
Error Probability

- The **probability of error** (or, the word error rate) P_{err} for any decoding scheme is the probability that the decoder output is a wrong codeword.
- It is also called the **Residual Error Rate**.

Indian Institute of Technology, Delhi Ranjan Bose Department of Electrical Engineering

Let us quickly recap what we learnt in the previous lectures. We have already considered the probability of error P err for any decoding scheme as that probability which the decoder output is a wrong codeword. This is also called as a residual error rate which means that despite applying the error control coding technique you still have a finite non zero residual error rate.

(Refer Slide Time: 01:28)



Information Theory, Coding and Cryptography

Probability of Error

- The union bound on probability of error P_M can be expressed

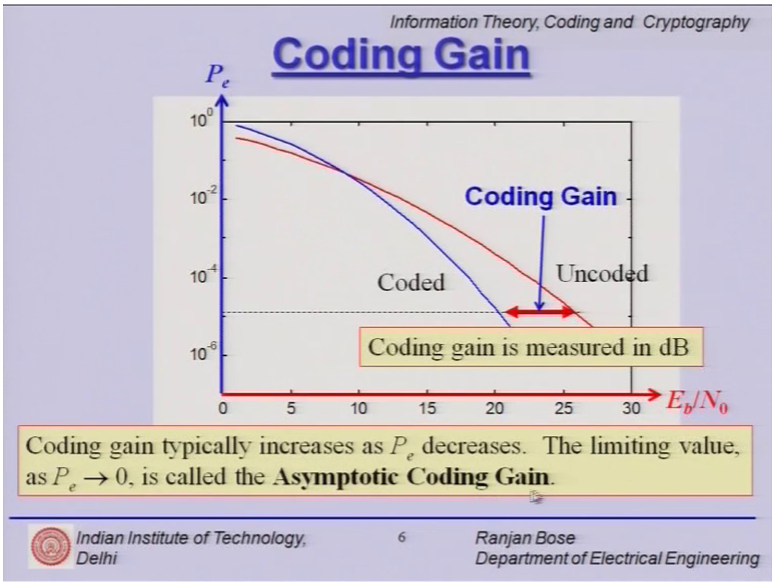
$$P_M \leq (M-1) \sum_{m=\lfloor d^*/2 \rfloor+1}^{d^*} \binom{d^*}{m} p^m (1-p)^{d^*-m}$$

Indian Institute of Technology, Delhi

Ranjan Bose
Department of Electrical Engineering

We also found that the upper bound on the probability of error P M can be expressed as follows. Where, capital M is 2 raised to the power k k being the length of the information word. These turn happens to be the minimum distance of the code and this is our a binary symmetric channel. So, small p is the probability of a bit getting flipped. So, we found this up using a union bound.

(Refer Slide Time: 01:59)



Information Theory, Coding and Cryptography

Coding Gain

Coding gain is measured in dB

Coding gain typically increases as P_e decreases. The limiting value, as $P_e \rightarrow 0$, is called the **Asymptotic Coding Gain**.

Indian Institute of Technology, Delhi

6

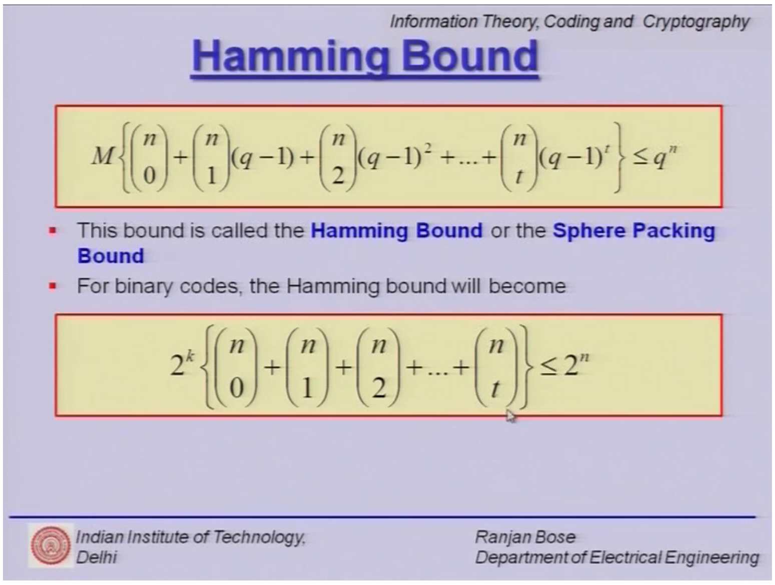
Ranjan Bose
Department of Electrical Engineering

We also found out what do we mean by the term coding gain it helps us compare two systems. So, if you have for example, a coded and an uncoded system and on the y axis we have the probability of error, on the x axis we have E b over N naught which is a measure of the signal to noise ratio in terms of dB for a certain probability of error say in this case approximately 10 raised to the power minus 5 we can say that the coding gain is this decrease in terms of E b over N naught. That is what is the reduced SNR required to give you the same performance in terms of the bit error rate.

Finally, we can sell a system in the market if it adheres to certain probability of error and this access changes depends depending on the application. So, if you have a medical data imaging problem I would rather be around 10 raised to the power 6 or 10 to the power minus 7, whereas if I am just doing simple digital voice transfer 10 to the power minus 4 and 10 to the power minus 5 are good enough. So, this axis actually tells you what is the application level and the quality of service that goes with that application.

So, coding gain changes with the P e coding gain is measured in dB. And typically, it increases as P e decreases the limiting value as P e tends to 0 is called the asymptotic coding gain as the theoretical upper limit.

(Refer Slide Time: 03:36)



Information Theory, Coding and Cryptography

Hamming Bound

$$M\left\{\binom{n}{0}+\binom{n}{1}(q-1)+\binom{n}{2}(q-1)^2+\ldots+\binom{n}{t}(q-1)^t\right\}\le q^n$$

- This bound is called the **Hamming Bound** or the **Sphere Packing Bound**
- For binary codes, the Hamming bound will become

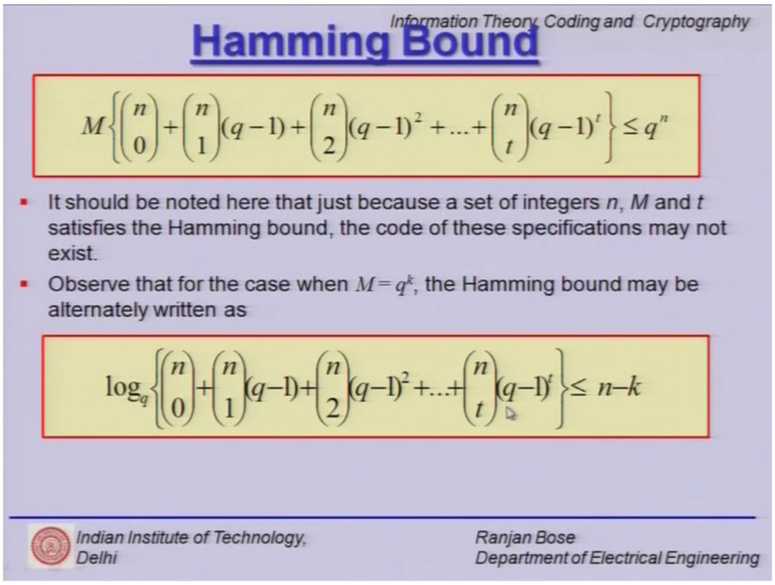
$$2^k\left\{\binom{n}{0}+\binom{n}{1}+\binom{n}{2}+\ldots+\binom{n}{t}\right\}\le 2^n$$

Indian Institute of Technology, Delhi

Ranjan Bose
Department of Electrical Engineering

We also looked at the Hamming bound or the sphere packing bound and if you look at the binary case thus Hamming bound is given by this following expression, where n is the block length, k is the length of the information word and t is the number of errors that it can correct.

(Refer Slide Time: 03:59)



Information Theory, Coding and Cryptography

Hamming Bound

$$M\left\{\binom{n}{0}+\binom{n}{1}(q-1)+\binom{n}{2}(q-1)^2+\ldots+\binom{n}{t}(q-1)^t\right\}\le q^n$$

- It should be noted here that just because a set of integers n, M and t satisfies the Hamming bound, the code of these specifications may not exist.
- Observe that for the case when $M=q^k$, the Hamming bound may be alternately written as

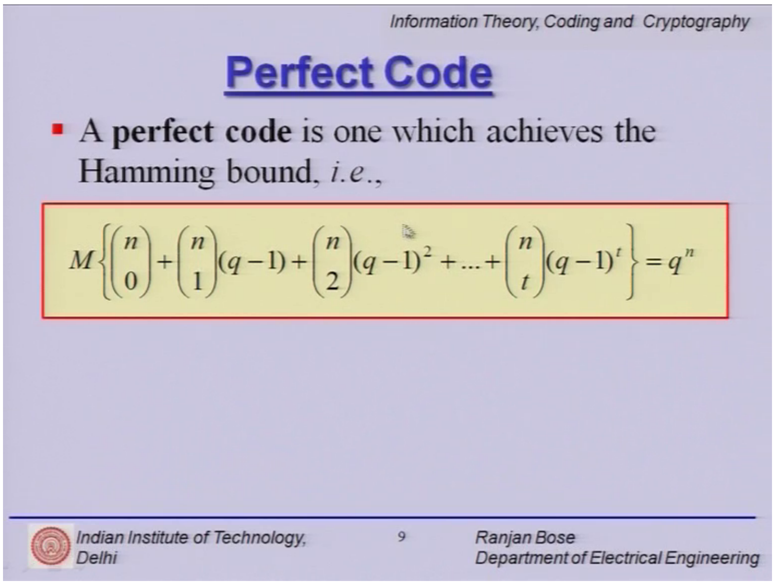
$$\log_q\left\{\binom{n}{0}+\binom{n}{1}(q-1)+\binom{n}{2}(q-1)^2+\ldots+\binom{n}{t}(q-1)^t\right\}\le n-k$$

Indian Institute of Technology, Delhi

Ranjan Bose
Department of Electrical Engineering

Now, we also made an observation that just because we have a set of integers n, M and t which satisfies the Hamming bound it does not necessarily mean that you can find a linear block code that exists for those numbers. And you can alternately write the Hamming bound by substituting M is equal to q raised to the power k as follows. So, please remember the Hamming bound is for a q-ary code, it is not necessarily for binary. Binary is a special case when q is equal to 2.

(Refer Slide Time: 04:35)



Information Theory, Coding and Cryptography

Perfect Code

- A **perfect code** is one which achieves the Hamming bound, *i.e.*,

$$M\left\{\binom{n}{0}+\binom{n}{1}(q-1)+\binom{n}{2}(q-1)^2+\ldots+\binom{n}{t}(q-1)^t\right\}=q^n$$

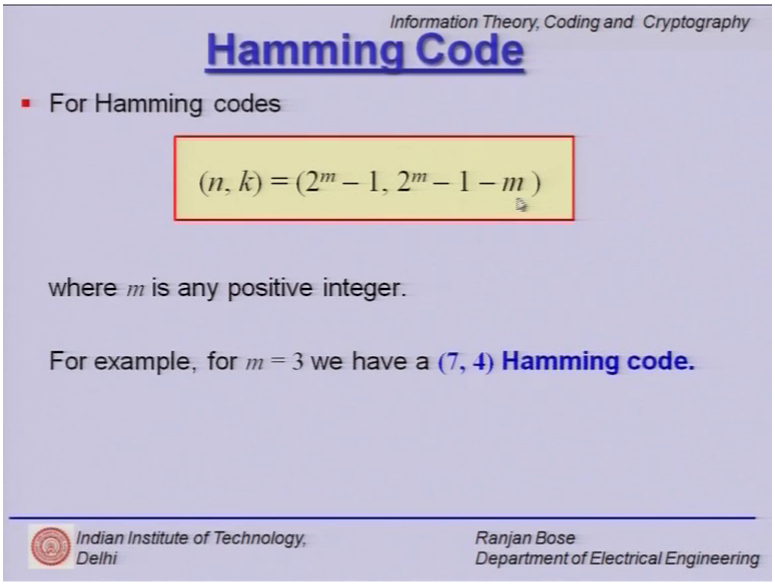
Indian Institute of Technology, Delhi

9

Ranjan Bose
Department of Electrical Engineering

What is a perfect code? Well, a perfect code is one that achieves the Hamming bound and we have put that equality here and this is the condition for a perfect code.

(Refer Slide Time: 04:45)



Information Theory, Coding and Cryptography

Hamming Code

- For Hamming codes

$$\boxed{(n, k) = (2^m - 1, 2^m - 1 - m)}$$

where m is any positive integer.

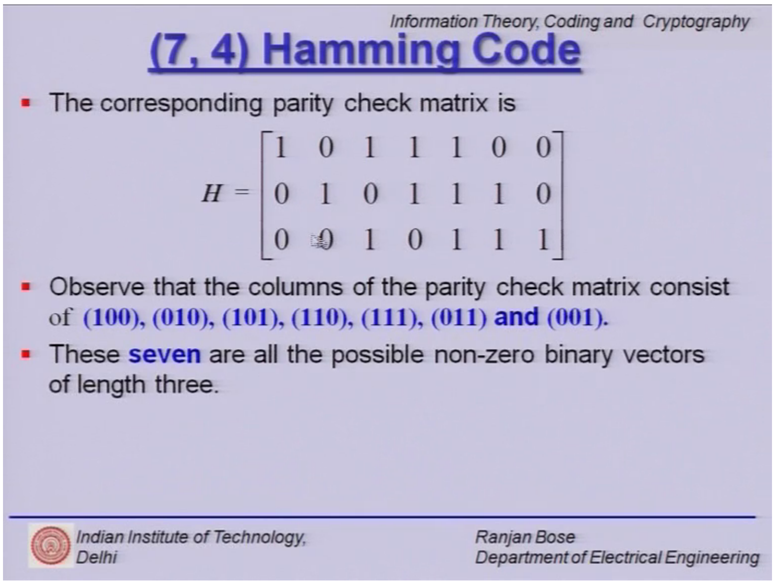
For example, for $m = 3$ we have a **(7, 4) Hamming code.**

Indian Institute of Technology,
Delhi

Ranjan Bose
Department of Electrical Engineering

Now, we move on to a class of codes called Hamming codes. The property of Hamming code is as follows: the n comma k is given by 2 raised to the power m minus 1 comma 2 raised to the power m minus 1 minus m, where m is any positive integer. So, all Hamming codes satisfy this n comma k condition. So, let us say m is equal to 3 and substituting 3 here would give you 7 and substituting 3 here will give you 4. So, 7 comma 4 is a valid Hamming code, alright. So, if you put m is equal to 4 you can have another value say 15 here and 11 here. So, 15 comma 11 is another valid Hamming code.

(Refer Slide Time: 05:42)



Information Theory, Coding and Cryptography

(7, 4) Hamming Code

- The corresponding parity check matrix is

$$H = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

- Observe that the columns of the parity check matrix consist of **(100), (010), (101), (110), (111), (011) and (001).**
- These **seven** are all the possible non-zero binary vectors of length three.

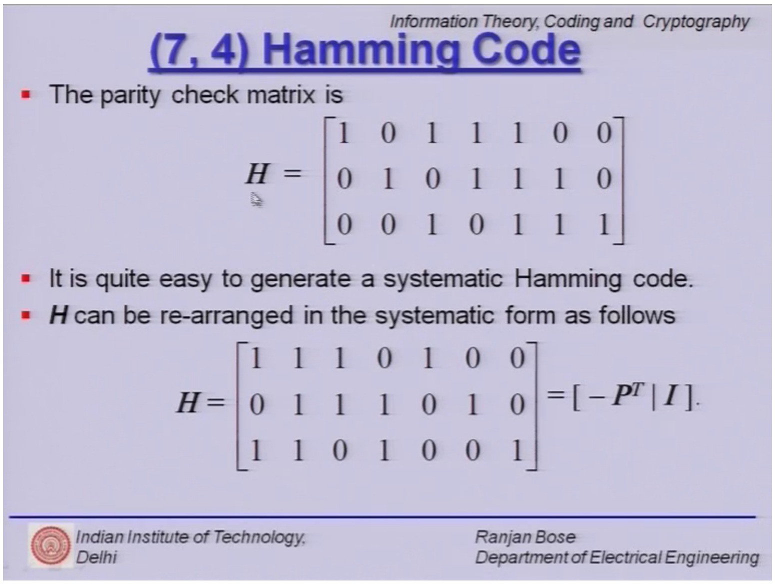
Indian Institute of Technology, Delhi

Ranjan Bose
Department of Electrical Engineering

So, let us look at the generator matrix for the binary 7 comma 4 Hamming code. So, we have the number of rows as four, because clearly k is equal to 4 and number of columns is seven, because n is equal to 7. So, it is a k cross n matrix which is the generator matrix. It forms the basis that generates the entire code space.

Now, you can always find the corresponding parity check matrix. In this case we list out the parity check matrix. Again, please note that the dimension is n minus k cross n. So, n minus k is 3 because n is 7, k is 4. So, 7 minus 4, 3; so the number of rows is three and number of columns is seven, but we make a couple of very interesting observations. The first observation is that all the columns of H right are basically the nonzero vectors the binary vectors of length 3. So, you have 0 0 1 0 1 0 0 1 1 so and so forth, up to 1 1 1. So, all the seven possible combinations barring the 0 0 0, there are seven possible vectors of length 3. All of them are listed in some sequence and this happens to be the check matrix.

(Refer Slide Time: 07:23)



Information Theory, Coding and Cryptography

(7, 4) Hamming Code

- The parity check matrix is

$$H = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

- It is quite easy to generate a systematic Hamming code.
- ***H*** can be re-arranged in the systematic form as follows

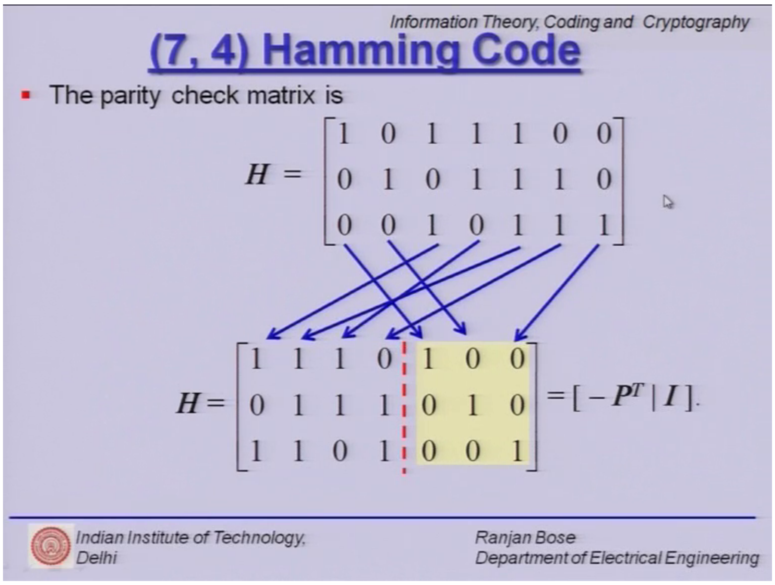
$$H = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} = [-P^T \,|\, I].$$

Indian Institute of Technology, Delhi

Ranjan Bose
Department of Electrical Engineering

So, we know that you can have an equivalent parity check matrix were simply rearranging the columns. So, since I have got these seven vectors and of this I have to identify 1 0 0, 0 1 0 and 0 0 1 to be kept on the rightmost side we can simply rearrange them.

(Refer Slide Time: 07:50)



Information Theory, Coding and Cryptography

(7, 4) Hamming Code

- The parity check matrix is

$$H = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

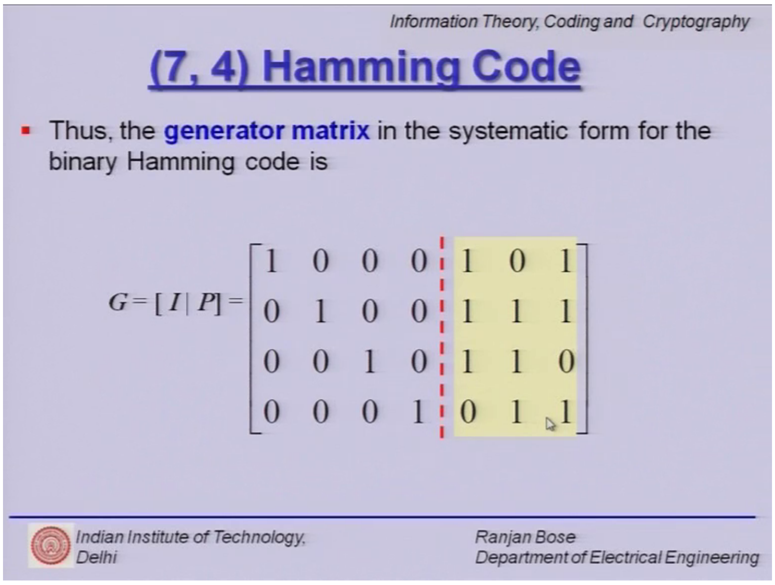
$$H = \left[\begin{array}{cccc|ccc} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{array}\right] = [-P^T \,|\, I].$$

Indian Institute of Technology, Delhi

Ranjan Bose
Department of Electrical Engineering

So, please note, we need it in the systematic form which is of the type minus P T I; I happens to be the identity matrix which is 3 cross 3, right. So, what we want to do is simply rearrange those columns and you get your H matrix and needless to say this can be this is not necessarily unique, you can have many other rearrangements for the first 4 columns and each one will be a valid H matrix, but this gives us a very nice way to put it in this systematic form. So, this is your parity check matrix for a 7 comma 4 binary Hamming code.

(Refer Slide Time: 08:40)



Information Theory, Coding and Cryptography

(7, 4) Hamming Code

- Thus, the **generator matrix** in the systematic form for the binary Hamming code is

$$G = [\,I\,|\,P\,] = \left[\begin{array}{cccc|ccc} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{array}\right]$$

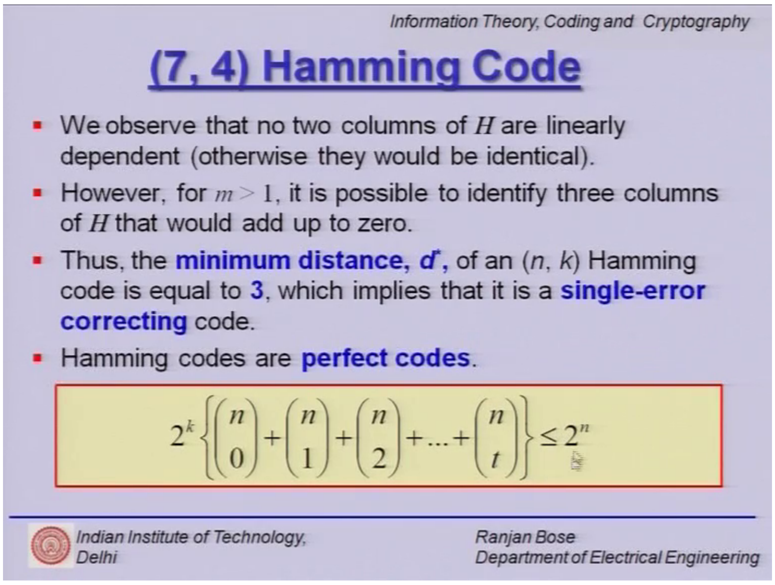
Indian Institute of Technology, Delhi

Ranjan Bose
Department of Electrical Engineering

Now, if we were to look at the generated matrix in a systematic form we have this P transpose. Here for this one please note in binary 1 plus 1 is 0. So, minus 1 is equal to plus 1. So, minus sign is not really making any difference it is as well as minus P T is equal to P T.

So, all you need to do is take this portion and just put a transpose of it and follow it. If we proceed it with an I. So, I have got this identity matrix 4 cross 4 and this is the transpose of that P T which gives me the P matrix and I have a quick rendering of the generated matrix. In a systematic form for the binary 7 comma 4 Hamming code this is your parity matrix and this is the I matrix.

(Refer Slide Time: 09:38)



Information Theory, Coding and Cryptography

(7, 4) Hamming Code

- We observe that no two columns of H are linearly dependent (otherwise they would be identical).
- However, for $m > 1$, it is possible to identify three columns of H that would add up to zero.
- Thus, the **minimum distance, d^*,** of an (n, k) Hamming code is equal to **3**, which implies that it is a **single-error correcting** code.
- Hamming codes are **perfect codes**.

$$2^k\left\{\binom{n}{0}+\binom{n}{1}+\binom{n}{2}+\ldots+\binom{n}{t}\right\}\le 2^n$$

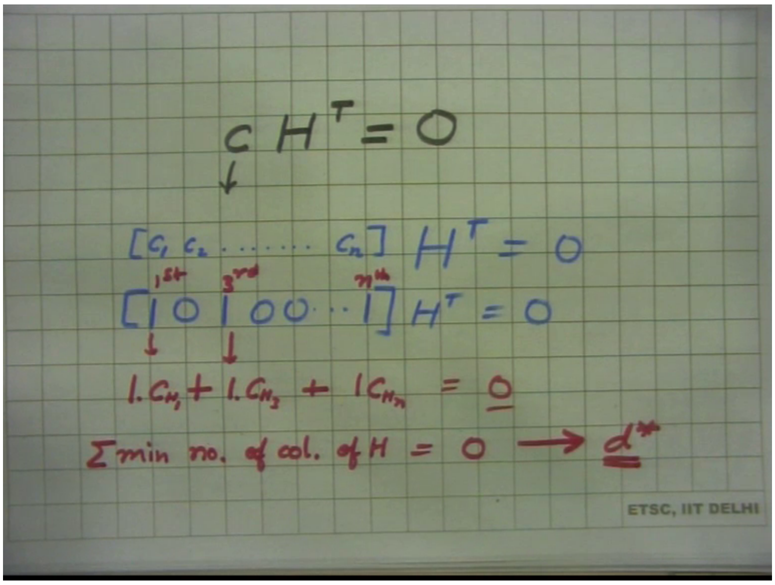
Indian Institute of Technology, Delhi

Ranjan Bose
Department of Electrical Engineering

Now, we observe that no two columns of H are linearly dependent, right because there only seven columns and we have exhausted all the seven vectors which are nonzero to form those columns. Clearly, no two columns of H are linearly dependent, right, but for m greater than 1, it is possible to identify three columns of H which would add up to 0. So, you can always go back to your H matrix and pick up three columns and these three columns would add up to 0.

Now, we established in the last class that that could lead us to the minimum distance of the code. Let us have a quick recap on that.

(Refer Slide Time: 10:40)



$$cH^T = 0$$

$$[c_1\ c_2 \cdots\cdots c_n]\ H^T = 0$$

$$[\overset{1st}{1}\ 0\ \overset{3rd}{1}\ 0\ 0 \cdots \overset{nth}{1}]\ H^T = 0$$

$$1.c_{H_1} + 1.c_{H_3} + 1c_{H_n} = \underline{0}$$

$$\Sigma\, \text{min no. of col. of } H = 0 \longrightarrow \underline{\underline{d^*}}$$

ETSC, IIT DELHI

So, we know that any valid code word c H transpose is equal to 0, that is by construction. So, what is it doing? This is H transpose so, c is actually picking up the columns of H right, because this is the H transpose. So, if you have this c as c 1, c 2, c n multiplied by H transpose equal to 0. All it is doing is the c 1 is picking up the first problem, c 2 is picking up the second column and c n is picking up the n-th column and it is adding up to 0.

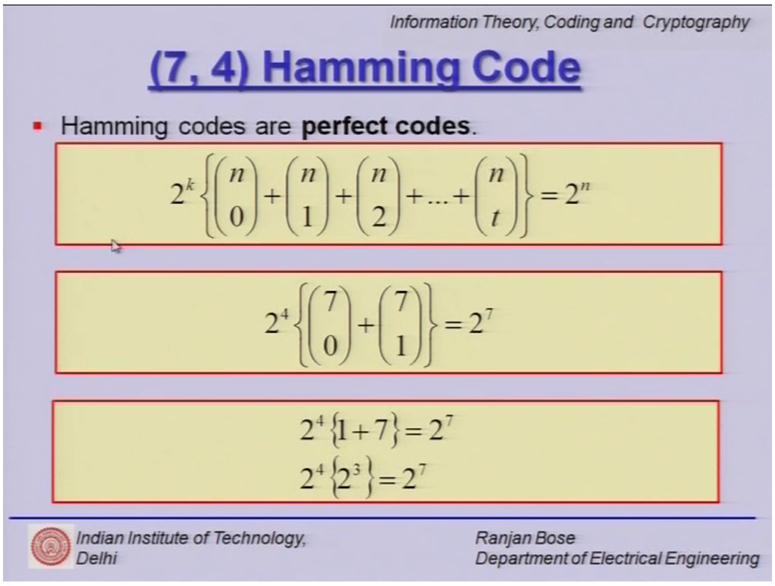
So, if the minimum number of columns of H which add up to 0 that is the minimum weight possible because in a binary case these are the either 1's or 0's; for example: if it were 1 0 1 0 0 1 and you have got this H transpose equal to 0 then what it does is this first guy picks up the column 1 of H. So, 1 into say column one of H, c H 1 and the 0 picks up the second column multiplies it by 0. So, it is negated plus one times c H 3 picks up the third column. So, it picks up the first column third column and n-th column, right and that is 0.

So, essentially the number of nonzero elements in the fact here in our binary case the number of one's here is picking up one column each and that is adding up to the 0 vector. So, the minimum number of columns of H that add up to 0 would give you d star, because how many columns they picked up by the nonzero elements of this code word. And therefore, that directly corresponds to your d star.

So, we come back to a slide and have a look at the H matrix here and we see that you can pick three of those columns and you can ensure that they add up to 0. Any two are linearly independent, but three of them add up to 0, it tells us that the minimum weight of this Hamming n comma k, Hamming code is equal to 0, but we have already seen that for distance d star equal to 3, it can add base to be a single error correcting code because d star should be greater than or equal to 2 t plus 1.

So, you have these Hamming codes even though they are perfect codes because they satisfy the condition for the profit codes are only single error correcting codes nonetheless they are very good they are optimal in certain senses. So, 7 comma 4 Hamming code is not only a good textbook example, but can also be used to give you a single error correction.

(Refer Slide Time: 14:47)



Information Theory, Coding and Cryptography

(7, 4) Hamming Code

- Hamming codes are **perfect codes**.

$$2^k\left\{\binom{n}{0}+\binom{n}{1}+\binom{n}{2}+\ldots+\binom{n}{t}\right\}=2^n$$

$$2^4\left\{\binom{7}{0}+\binom{7}{1}\right\}=2^7$$

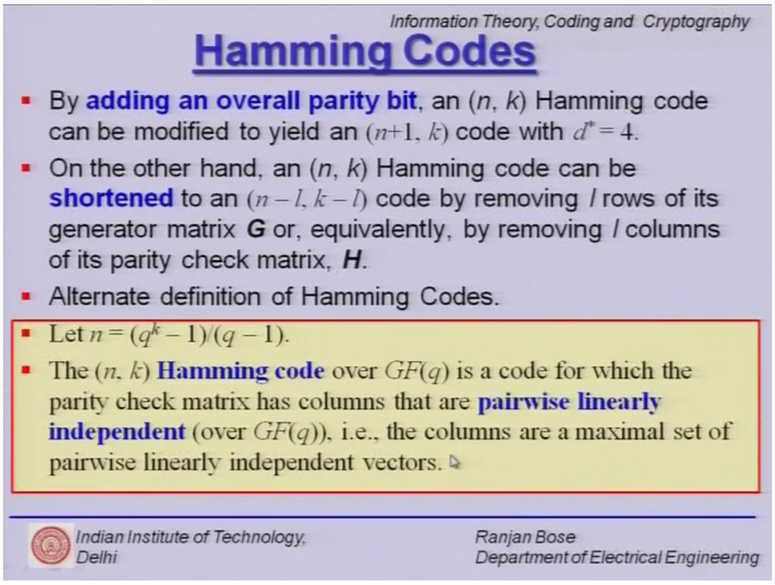
$$2^4\{1+7\}=2^7$$
$$2^4\{2^3\}=2^7$$

Indian Institute of Technology,
Delhi

Ranjan Bose
Department of Electrical Engineering

Now, for perfect codes we need to check this condition because this inequality should be met here. So, if it needs to be equal if it is for perfect code. So, we put this equality. So, 2 raised to the power k up to t error correction should be equal to 2 raised to the power n, but in our Hamming code case we know the 7 comma 4 code n is 7, k is 4, t is 1. So, if you put this condition k is 4, n is 7 and t is 1 so, we only take the first two terms in the summation and substitute them. We quickly can verify that indeed this equality holds and 7 comma 4 Hamming code is a perfect code.

(Refer Slide Time: 15:22)



Information Theory, Coding and Cryptography

Hamming Codes

- By **adding an overall parity bit**, an (n, k) Hamming code can be modified to yield an $(n+1, k)$ code with $d^* = 4$.
- On the other hand, an (n, k) Hamming code can be **shortened** to an $(n-l, k-l)$ code by removing l rows of its generator matrix ***G*** or, equivalently, by removing l columns of its parity check matrix, ***H***.
- Alternate definition of Hamming Codes.
- Let $n = (q^k - 1)/(q - 1)$.
- The (n, k) **Hamming code** over $GF(q)$ is a code for which the parity check matrix has columns that are **pairwise linearly independent** (over $GF(q)$), i.e., the columns are a maximal set of pairwise linearly independent vectors.

Indian Institute of Technology, Delhi

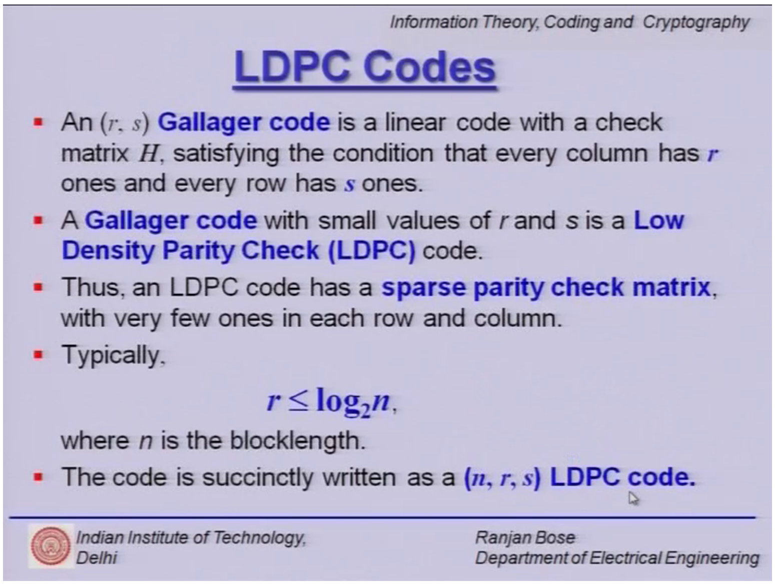
Ranjan Bose
Department of Electrical Engineering

Now, let us see what we can do by tweaking it a little bit. So, already you know you have an n comma k Hamming code you can always modify it to an n plus 1 comma k code with d star is equal to 4. So, by just adding one more bit to the code word you can. So, adding what do you mean by adding one more bit I add a parity bit and I make it n plus 1 comma k code and suddenly my d star becomes 4.

We can go the other way also and we can shorten the n comma k code. Now, how do we do that? Well, n comma k also gives us the idea that G is k cross n. So, I can start deleting rows and columns. So, if I can delete l rows and l columns, I get a modified shortened G matrix for Hamming code, right. So, you can also remove the columns of H and you will get the equivalent H. So, I can play these small tricks to get a smaller or larger code with interesting distance properties, they also an alternate definition of Hamming code. So, let n is equal to q power k minus 1 divided by q minus 1. So, you can have q for example, for binary case q is equal to 2 and k is equal to 4. So, you can substitute these values then an n comma k Hamming code over GF q is a code for which the parity check matrix has columns that are pairwise linearly independent, ok.

So, we are now trying to tell me in a kind of a different way how you can actually construct a Hamming code. We have to get the pairwise linearly independent columns of H and once we have an H we can always construct a G and vice versa. So, that is the columns are a maximal set of pairwise linearly independent vectors, and we saw the simple example for a 7 comma 4 code that each of the columns of pairwise linearly independent, but 3 of them added up to 0.

(Refer Slide Time: 18:01)



Information Theory, Coding and Cryptography

LDPC Codes

- An (r, s) **Gallager code** is a linear code with a check matrix H, satisfying the condition that every column has r ones and every row has s ones.
- A **Gallager code** with small values of r and s is a **Low Density Parity Check (LDPC)** code.
- Thus, an LDPC code has a **sparse parity check matrix**, with very few ones in each row and column.
- Typically,

$$r \leq \log_2 n,$$

where n is the blocklength.

- The code is succinctly written as a **(n, r, s) LDPC code.**

Indian Institute of Technology, Delhi

Ranjan Bose
Department of Electrical Engineering

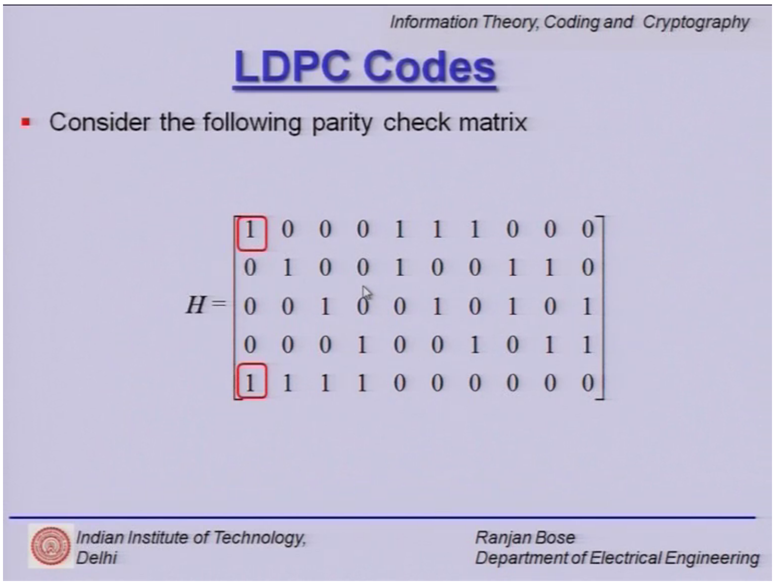
We now look at another class of codes, another class of linear block codes called the low density parity check code, but for that we start with a slightly different definition which is the Gallager code. So, an r comma s Gallager code is a linear code with a check matrix H which satisfies the following condition. So, H matrix has every column has r ones and every row has s ones rested all zeros. So, suddenly if you want to visualize a binary code we are putting constraints on how many ones you can have on every column and how many ones you can have in every rows and if we make this constraint straight that is r and s values are small then my H matrix is pretty much full of zeros,.

So, if a Gallager code with code with small values of r and s is basically a parity check matrix which is mostly zeros and a few ones in the rows and columns as dictated by r and s and by squeezing r and s I can reduce the number of ones and thereby making it a low density; low density of what - ones. So, it is a low density parity check code and we will quickly come to the benefit of this, but first let us understand the job of a parity check code.

What a parity check code does is it takes? So, the received vector in nu, nu H transpose it checks for the syndrome. Now, each time I do nu H transpose, the different elements of nu takes up the rows, the columns of the H matrix and performs computation, but if most of the elements are 0, then the computation load really really goes down. So, question is can we still have good interesting distance properties despite having the parity check matrix highly rare that is the density is pretty low.

So, thus an LDPC code has a sparse parity check matrix with very few ones in each row and column. So, typically LDPC code has r less or equal to log to the base 2 n, where n is the block length. So, if you remember the size of a the parity check matrix is n minus k cross n, there are n columns and number of rows and number of ones in every column has is limited by log to the base 2 n. So, it is kind of a rule of thumb which gives you what is low enough. So, this code can be written as n which is the block length r, where every column has r ones s where every row has s ones; so n comma r comma s LDPC codes. These are the parameters of my LDPC codes.

(Refer Slide Time: 21:35)



Information Theory, Coding and Cryptography

LDPC Codes

- Consider the following parity check matrix

$$H = \begin{bmatrix} \boxed{1} & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ \boxed{1} & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

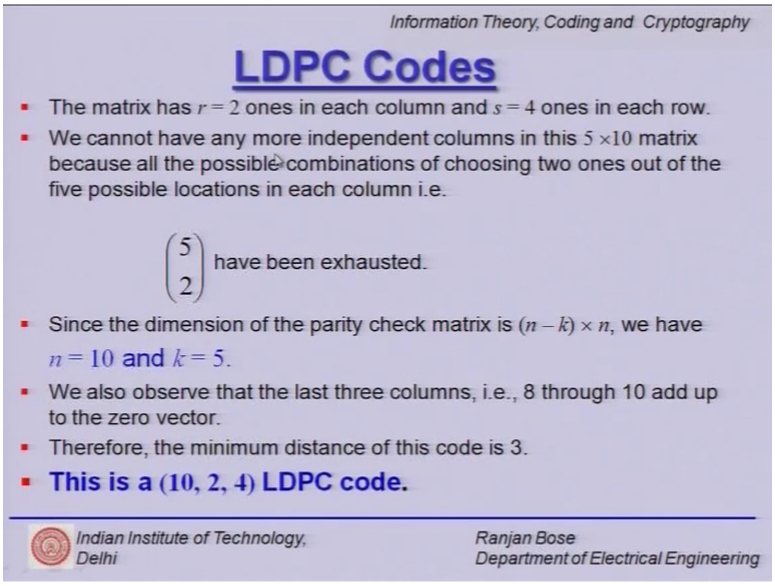
Indian Institute of Technology, Delhi

Ranjan Bose
Department of Electrical Engineering

Let us quickly look at an example. So, let us look at this H matrix which is the parity check matrix. Let us first look at the columns, pick any columns in the third column. I see only two ones, I pick any other 1. Suppose, I pick up the fifth again I see only two ones I pick up any one and again I see only two ones. In fact, I can do this exercise then I can make a sanity check and see that all the columns have at most 2. So, r is equal to 2.

Now, look at the rows I count the number of ones; 1 2 3 4. So, four ones most are zeros again 1 2 3 4 four ones four ones four. So, we say that there at most four ones; in fact, exactly four ones in every row and two ones in every column. So, r is equal to 2 and s is equal to 4 for this parity check matrix and if I count the number of columns 1 2 3 4 5 6 7 8 9 10. So, n is equal to 10, ok. So, this is 10 comma so r; r is 2, 2 comma 4, 10 comma 2 comma 4 LDPC code.

(Refer Slide Time: 23:01)



Information Theory, Coding and Cryptography

LDPC Codes

- The matrix has $r = 2$ ones in each column and $s = 4$ ones in each row.
- We cannot have any more independent columns in this 5×10 matrix because all the possible combinations of choosing two ones out of the five possible locations in each column i.e.

$$\binom{5}{2} \text{ have been exhausted.}$$

- Since the dimension of the parity check matrix is $(n-k) \times n$, we have $n = 10$ and $k = 5$.
- We also observe that the last three columns, i.e., 8 through 10 add up to the zero vector.
- Therefore, the minimum distance of this code is 3.
- **This is a (10, 2, 4) LDPC code.**

Indian Institute of Technology, Delhi

Ranjan Bose
Department of Electrical Engineering

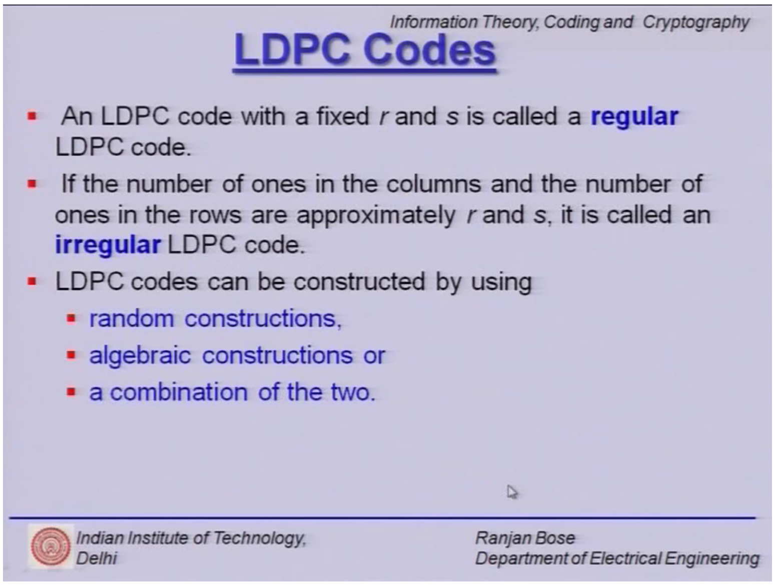
So, we just now verified that r is equal to two ones in each column and s equal to four ones in each rows and clearly we cannot have any more independent columns in this 5 cross 10 matrix, with all possible combinations of choosing two ones out of the five possible locations. 5 choose 2, it says 10 has an exhausted. Therefore, you have n is equal to 10 and k is equal to 5 because the dimension of a parity check matrix is n minus k cross n.

So, we also observed that the last three columns add up to the zero vector; so if you go back and if you look at these things 0 0 0 this 1 plus 1 0 1 plus 1 0 0 0. So, the last three columns if you add them up it is 0 and you can try as much as you want to. You cannot find two columns added up to 0. So, the minimum number of columns that add up to 0 is 3 consequently the d star must be 3 and hence it is a single error correcting code.

So, what does it tell us? It tells us this is a sparse matrix for whatever reason let us believe that it is sparse enough; although, real LDPC codes have pretty much most of the weight as zeros and very few ones. So, this is a toy example, but you will see that despite having lots of zeros you have not compromised on the distance. The minimum distance is 3 and still it is a useful code, it can correct a single error: single error in the code.

So, this is a 10 comma 2 comma 4 LDPC code, this is an example of that.

(Refer Slide Time: 24:55)



Information Theory, Coding and Cryptography

LDPC Codes

- An LDPC code with a fixed *r* and *s* is called a **regular** LDPC code.
- If the number of ones in the columns and the number of ones in the rows are approximately *r* and *s*, it is called an **irregular** LDPC code.
- LDPC codes can be constructed by using
  - random constructions,
  - algebraic constructions or
  - a combination of the two.

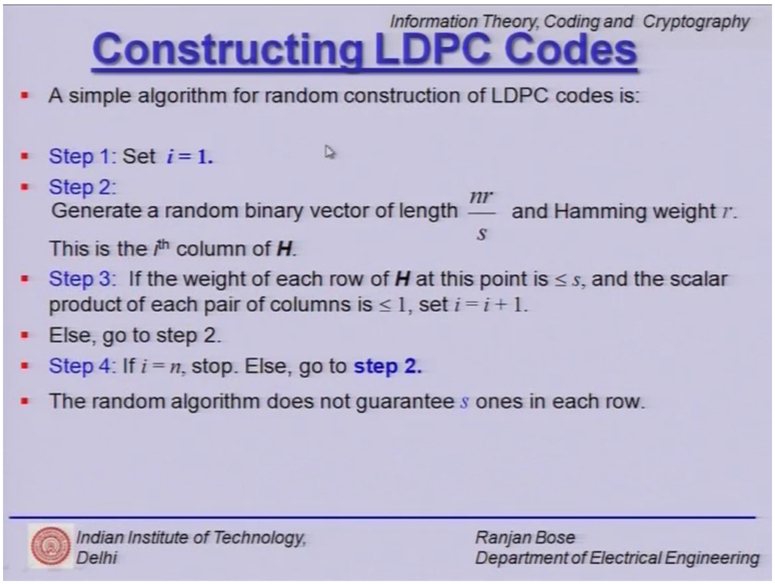
Indian Institute of Technology,
Delhi

Ranjan Bose
Department of Electrical Engineering

Some more definitions an LDPC code with fixed r and s is called a regular LDPC code. So, the example that we looked at just now r was fixed at 2 and s was fixed at 4; so it was a regular LDPC code, but if the number of ones of the columns of number of ones in the rows are approximately r and s. So, we do not restrict it that it necessarily has to be 2, it can be between 2 and 3 and s can be between 4 and 6 we can put those constraints. So, then it is called an LDPC code which is irregular.

So, LDPC codes can constructed by using either a random construction or algebraic construction or a combination of 2. So, I can choose any of these techniques or combine them to form my LDPC codes, but whenever I construct an LDPC code I first get the H matrix and then I get my G matrix.

(Refer Slide Time: 25:58)



Information Theory, Coding and Cryptography

Constructing LDPC Codes

- A simple algorithm for random construction of LDPC codes is:
- Step 1: Set $i = 1$.
- Step 2:
  Generate a random binary vector of length $\frac{nr}{s}$ and Hamming weight r.
  This is the i^{th} column of ***H***.
- Step 3: If the weight of each row of ***H*** at this point is $\leq s$, and the scalar product of each pair of columns is ≤ 1, set $i = i + 1$.
- Else, go to step 2.
- Step 4: If $i = n$, stop. Else, go to **step 2.**
- The random algorithm does not guarantee s ones in each row.

Indian Institute of Technology,
Delhi

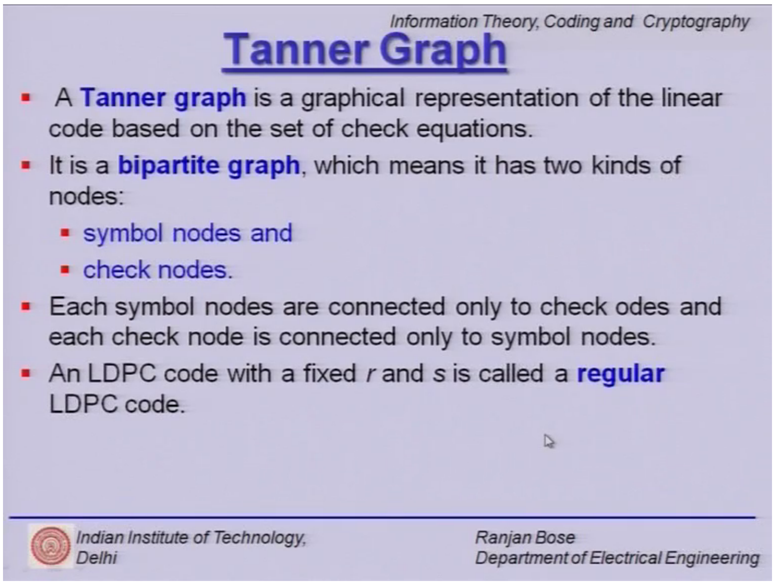
Ranjan Bose
Department of Electrical Engineering

So, a simple algo is given as follows for random construction of LDPC codes. There are four steps in this algorithm set I is equal to one and the second step generate a random binary vector of length n r over s. These are the parameters n is the block length, r is the number of ones in the columns, s is the number of ones in the rows and Hamming weight r.

So, this is i-th column of H. So, I am constructing H I need to get n vectors in place and I get the i-th column, the i-th vector as follows in a random fashion if the weight of each row. So, I put the column in place, but now I talk about the weight if the weight of each 0 of H at this point is less than or equal to s, because s is the constraint I have and the scalar product of each pair of columns is less than 1. Then set is i is equal to i plus 1, that is I go to the next column else go to step 2.

So, we are constructing it right and we are ensuring that you are not exceeding the number of ones in the rows to more than s, and if you can reach I is equal to n that is you have found all the n vectors which satisfy the conditions of the maximum number of ones in the columns and the rows then you have actually successfully constructed the LDPC code. Now, the random algorithm does not guarantee s ones in each row, but it gives you less than or equal to s ones.

(Refer Slide Time: 27:44)



Information Theory, Coding and Cryptography

Tanner Graph

- A **Tanner graph** is a graphical representation of the linear code based on the set of check equations.
- It is a **bipartite graph**, which means it has two kinds of nodes:
  - symbol nodes and
  - check nodes.
- Each symbol nodes are connected only to check odes and each check node is connected only to symbol nodes.
- An LDPC code with a fixed *r* and *s* is called a **regular** LDPC code.

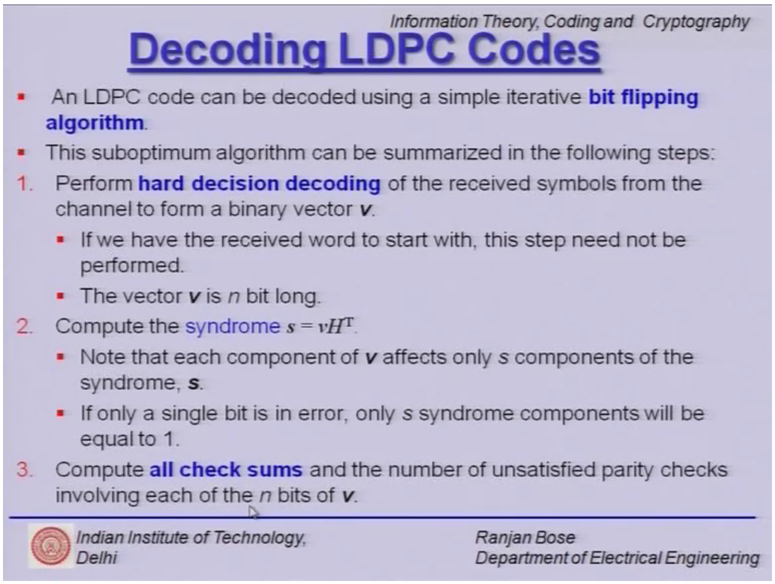
Indian Institute of Technology, Delhi

Ranjan Bose
Department of Electrical Engineering

Now, let us look at something called as a Tanner graph. Now, it is a very interesting graphical representation of the linear code based on based on check equations. So, we again move the focus to the parity check matrix how it works and since parity check matrix does the great job of either telling you whether the code is a valid code word. The receive code received vector is a valid code word or the syndrome associated with it we look at the parity check matrix side.

So, the Tanner graph is a bipartite graph which means it has two kinds of nodes. The symbol nodes and the check nodes and these are connected. So, each symbol node is connected only to the check nodes and each check nodes connected only to the symbol nodes. An LDPC code with fixed r and s is called a regular LDPC code as we have seen earlier.

(Refer Slide Time: 28:42)



Information Theory, Coding and Cryptography

Decoding LDPC Codes

- An LDPC code can be decoded using a simple iterative **bit flipping algorithm**.
- This suboptimum algorithm can be summarized in the following steps:

1. Perform **hard decision decoding** of the received symbols from the channel to form a binary vector ***v***.
   - If we have the received word to start with, this step need not be performed.
   - The vector ***v*** is *n* bit long.
2. Compute the syndrome $s = vH^T$.
   - Note that each component of ***v*** affects only *s* components of the syndrome, ***s***.
   - If only a single bit is in error, only *s* syndrome components will be equal to 1.
3. Compute **all check sums** and the number of unsatisfied parity checks involving each of the *n* bits of ***v***.

Indian Institute of Technology, Delhi

Ranjan Bose
Department of Electrical Engineering

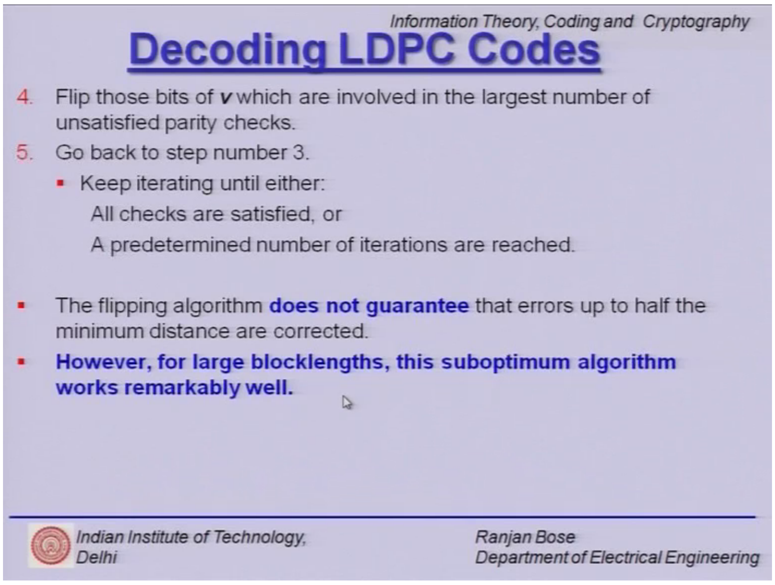
And, how do we do this, decoding part something has to be good about this LDPC code because they are very popular and they are finding place in the newer wireless standards. One of the methods is called the bit flipping algorithm. Let us understand this algorithm and we will follow it up with an example.

So, it is a simple iterative algorithm. The beauty lies in its simplicity and the minimum number of computations required to come to a right conclusion also if you fail to come to a right conclusion you can all always say that, sorry, I tried my best my number of iterations have been exhausted and I could not declare the result. So, it is not that if it fails to give you a correct answer, it will definitely give you the wrong answer. It will raise its flag and say no, I could not complete the decoding process.

So, what is suboptimal algorithm what does it look like well we first perform hard decision decoding on the received symbol to form a received vector nu. So, the first vector nu which is the received vector is n bit long, we do not know whether nu is the valid code word or it is a code word plus error. So, first job is to find that out. So, we find the s as nu H transpose, ok. This is the syndrome decoding strip and I will jump with joy if s comes out to be 0 and I will declare nu to be the correct valid code word. However, if s is nonzero my fun begins and I will try to take a call based on a bit flipping algorithm.

So, note that each component of nu affects only s components of the syndrome s, if only a single bit is in error only s syndrome components will be equal to 1. So, this is an important observation. So, now, we compute all check sums and the number of unsatisfied parity checks involving each of the n bits of nu are figured out. So, what we do is if nu is nonzero I start my action and I try to identify which of the elements of nu are leading to the unsatisfied parity checks.

(Refer Slide Time: 31:19)



Information Theory, Coding and Cryptography

Decoding LDPC Codes

4. Flip those bits of ***v*** which are involved in the largest number of unsatisfied parity checks.
5. Go back to step number 3.
   - Keep iterating until either:
     All checks are satisfied, or
     A predetermined number of iterations are reached.

- The flipping algorithm **does not guarantee** that errors up to half the minimum distance are corrected.
- **However, for large blocklengths, this suboptimum algorithm works remarkably well.**

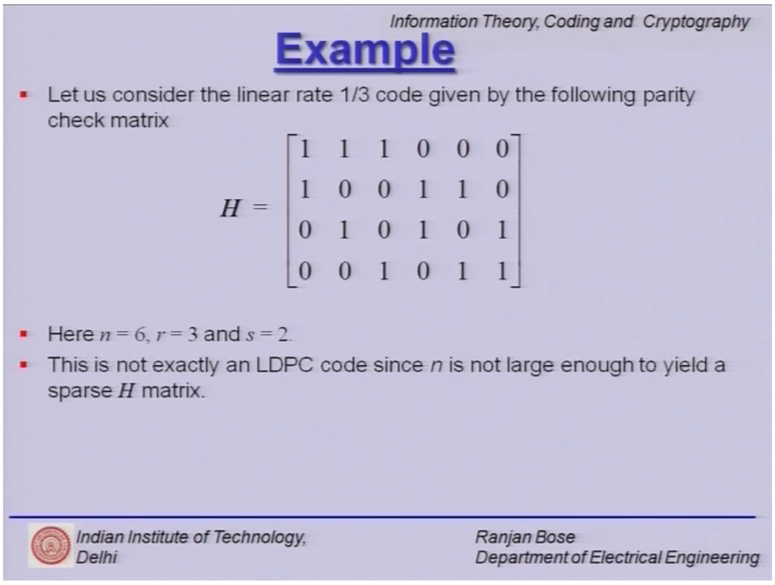
Indian Institute of Technology, Delhi

Ranjan Bose
Department of Electrical Engineering

For those bits of nu which are involved in the largest number of unsatisfied parity check I flip them this is the bit flipping algorithm and then I go back to my step 3 which is again compute the checksum. And again go and see flip those bits of new received vector which are involved in the largest number of unsatisfied parity checks. And I keep doing it till all the checks are satisfied that I get a 0 vector in nu prime H transpose or the maximum number of iterations allowed are reached and I declare that sorry, I could not find the answer.

So, this bit flipping algorithm does not guarantee that errors up to half the minimum distance are corrected. However, for large block lengths this suboptimal algorithm works remarkably well. So, let us look at a very simple example.

(Refer Slide Time: 32:17)



Information Theory, Coding and Cryptography

Example

- Let us consider the linear rate 1/3 code given by the following parity check matrix

$$H = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

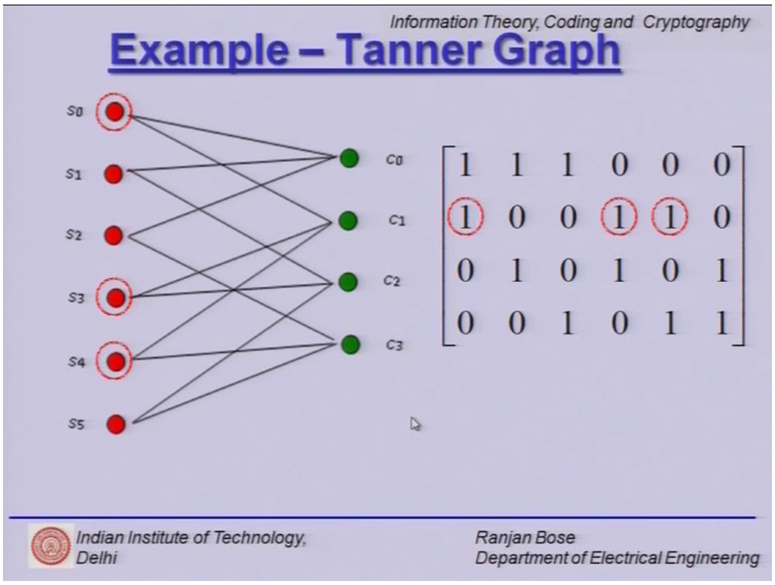
- Here $n = 6$, $r = 3$ and $s = 2$.
- This is not exactly an LDPC code since n is not large enough to yield a sparse H matrix.

Indian Institute of Technology, Delhi

Ranjan Bose
Department of Electrical Engineering

This is your rate 1 by 3 code. So, if you can see that this is n minus k and this is n. So, this n is 6, all right and this is your parity check matrix here every column if you see has r is equal to 2 and if you observe every row s is equal to 3 ones are there. So, for this even though in the truest sense it is not a sparse matrix, but it will at least it fix the slide and we are not going to go for a much larger toy example here. So, m is equal to 6 r is equal to 3 s is equal to 2, but this is we have set this is not n is not large enough to yield a really sparse H matrix what we live with it.

(Refer Slide Time: 33:11)



Information Theory, Coding and Cryptography

Example – Tanner Graph

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ \boxed{1} & 0 & 0 & \boxed{1} & \boxed{1} & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

Indian Institute of Technology, Delhi

Ranjan Bose
Department of Electrical Engineering

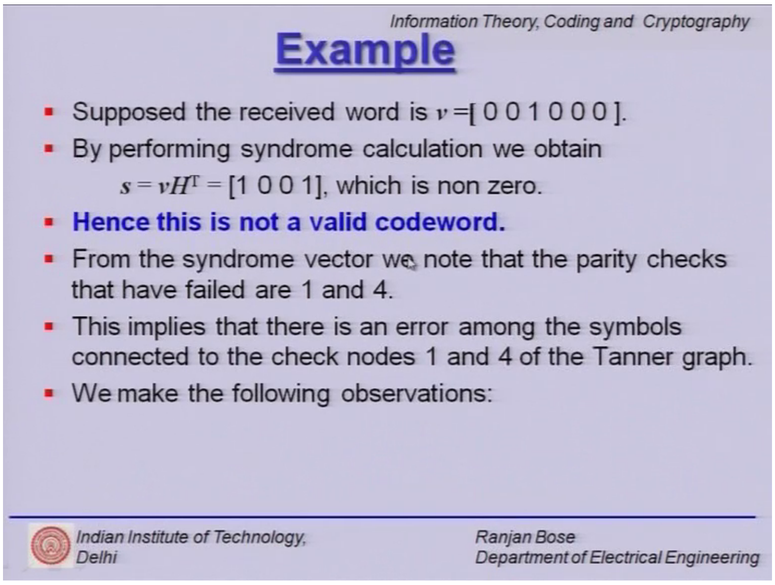
So, we first construct the Tanner graph what do we do with it? On the right you see that is the H matrix is there on the left I have put the Tanner graph. It is a by bipartite graph. Here we have these nodes and this is the check nodes.

Now, please note that 1 1 1 0 0 0. So, this c 0 is connected only to s 0, s 1, s 2 what does these can relate to? These correspond to your new vector that you receive the vector that you receive yes that is of the length n block length and if the errors in s 0, s 1 and s 2 they will relate to this c 0 what is this? Well, this is my syndrome. So, the greens are my elements on the syndrome vector reds are the elements of the received vector. Whenever you put the received vector nu and processor through H transpose you get these as the syndrome.

So, what is the beauty; is that because of the sparsity there are very few connections actually. If I start increasing the number of ones in by parity check matrix the number of interconnection should just go up. So, limiting I can keep increasing my n, but I will limit the number of connections and therein lies the beauty of this low density parity check. So, if my received vector list of syndrome which is 1 0 0 0, it clearly says that this guy which is one should flag and lead me to s 0, s 1, s 2 being the culprits or the suspects.

On the other hand, if I have 1 0 0 1. So, this c 3 also shows a one it means that s 2, s 4 and s 5 which are connected to c 3 one of them are possibly wrong. So, if I look at this guy I have these two and so and so forth. So, you know that whichever element becomes a 1 it has some suspects that it raises up and I will do a voting and see which of the nodes are maximally under suspicion.

(Refer Slide Time: 36:21)



Information Theory, Coding and Cryptography

Example

- Supposed the received word is v =[0 0 1 0 0 0].
- By performing syndrome calculation we obtain

  $s = vH^T$ = [1 0 0 1], which is non zero.
- **Hence this is not a valid codeword.**
- From the syndrome vector we note that the parity checks that have failed are 1 and 4.
- This implies that there is an error among the symbols connected to the check nodes 1 and 4 of the Tanner graph.
- We make the following observations:

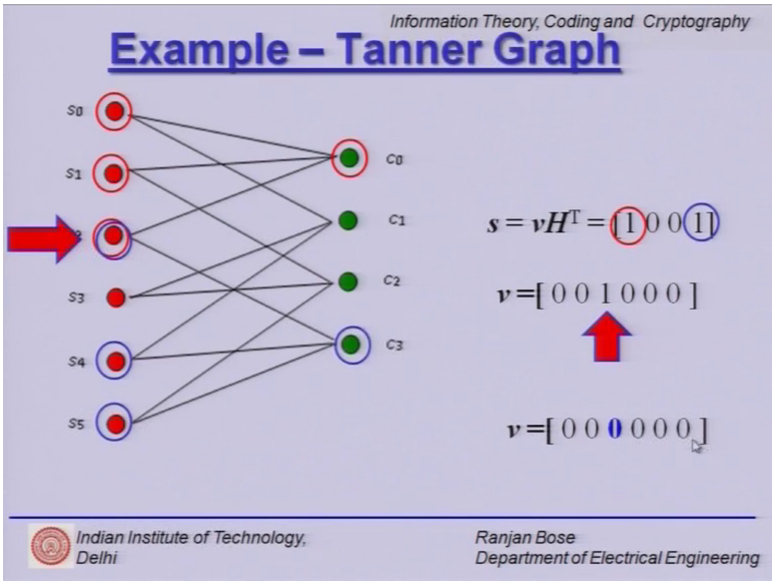
Indian Institute of Technology, Delhi

Ranjan Bose
Department of Electrical Engineering

So, let us understand this with a simple example. Suppose, we happen to receive vector 0 0 1 0 0 0 at the receiver and I wish to do whether this is a valid code word, whether I have made an error? If yes, then what is the corrected version? So, first stop is to find out the syndrome I quickly take my H matrix and I perform nu H transpose and I get 1 0 0 1 which is a non zero. So, clearly I declare that this received vector nu is not a valid code word, but then what is that code word which resulted in this error I need to do the correct decoding.

Let us do this bit flipping algorithm for this LDPC code to do. So, first observation is that the syndrome has the first element and the last element as ones and rest are zeros, ok; 0 0 0 0 would say nobody has made any errors. These two might lead to which of these have made an error. So, this implies that there is an error among the symbols connected to the check nodes 1 and 4. These are the check nodes and two flags have gone up in the Tanner graph.

(Refer Slide Time: 37:45)


Information Theory, Coding and Cryptography

Example – Tanner Graph

$s = vH^T = [1\ 0\ 0\ 1]$

$v = [0\ 0\ 1\ 0\ 0\ 0]$

$v = [0\ 0\ 0\ 0\ 0\ 0]$

Indian Institute of Technology, Delhi

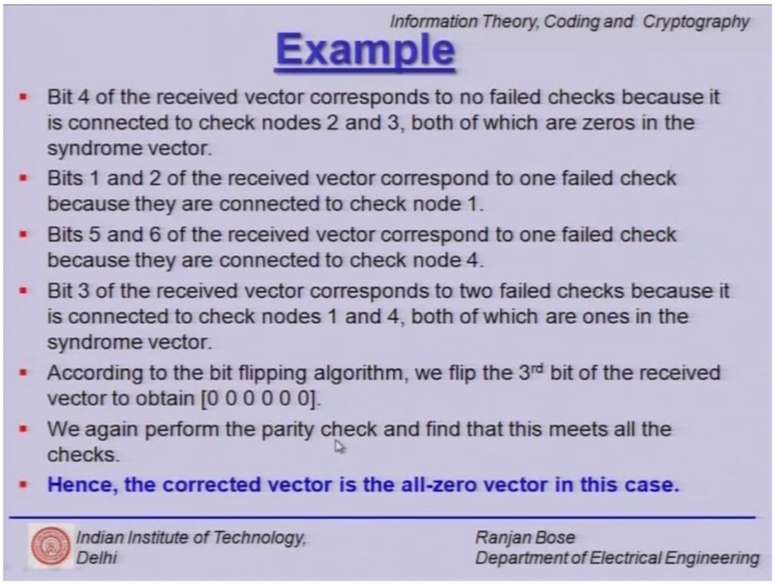
Ranjan Bose
Department of Electrical Engineering

So, we make the following observation, ok. So, we pull up the Tanner graph and we look at the syndrome and we identify the c 0 the first element of my syndrome and the last c 3. They are the ones whose flags have been raised and now, I have to go back and see which of these received vector which of these elements need to be flipped or changed, so that I can get a 0 0 0 0 here that is the logic.

So, first I look at this first guy; first one it is only connected to these three. So, first guess is that any one of these three s 0, s 1, s 2 could have resulted in this one. Either this is flipped or this is flipped or this is flipped, but I do not know. So, I look at the second guy and I say well c 3 is connected to s 2, s 4, s 5, right. Now, s 2 is raising the maximum suspicion because not only is it connected to c 0, but it is also connected to c 3.

So, what I will do is I pull back my nu the received vector and I said which is that number 3 position, this is the guy which is connected to two of the check nodes which are in error. So, I flip it is a bit flipping algorithm. So, I am not sure whether it is right or not I am I am taking a guess it is an intermediary step. So, I flip this having flip this I again perform this new prime H transpose, but this time I will get a 0 0 0 vector. So, the syndrome will be 0. So, this bit flipping has led me to a decoded a correctly decoded vector nu which is the all zero vector that was sent.

(Refer Slide Time: 40:03)



Information Theory, Coding and Cryptography

Example

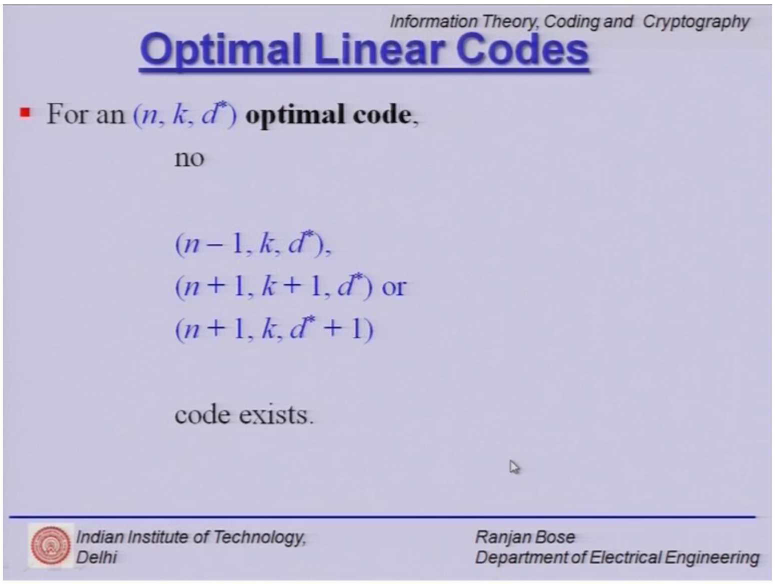
- Bit 4 of the received vector corresponds to no failed checks because it is connected to check nodes 2 and 3, both of which are zeros in the syndrome vector.
- Bits 1 and 2 of the received vector correspond to one failed check because they are connected to check node 1.
- Bits 5 and 6 of the received vector correspond to one failed check because they are connected to check node 4.
- Bit 3 of the received vector corresponds to two failed checks because it is connected to check nodes 1 and 4, both of which are ones in the syndrome vector.
- According to the bit flipping algorithm, we flip the 3rd bit of the received vector to obtain [0 0 0 0 0 0].
- We again perform the parity check and find that this meets all the checks.
- **Hence, the corrected vector is the all-zero vector in this case.**

Indian Institute of Technology, Delhi

Ranjan Bose
Department of Electrical Engineering

So, this is a simple example that tells me how a bit flipping example algorithm works and this is exactly what is written in this slide. So, bit 4 of the received vector corresponds to no field checks bit 1 and 2 of the received vector corresponds to check node 1. Similarly, bits 5 and 6 to check node 4. So, we follow this up and look at the conclusion that third bit could be possibly flipped we do that flipping and check for the parity check matrix. Hence, the corrected vector is the all zero vector in our case.

(Refer Slide Time: 40:34)



Information Theory, Coding and Cryptography

Optimal Linear Codes

- For an (n, k, d^*) **optimal code**,

  no

  $(n-1, k, d^*)$,
  $(n+1, k+1, d^*)$ or
  $(n+1, k, d^*+1)$

  code exists.

Indian Institute of Technology, Delhi

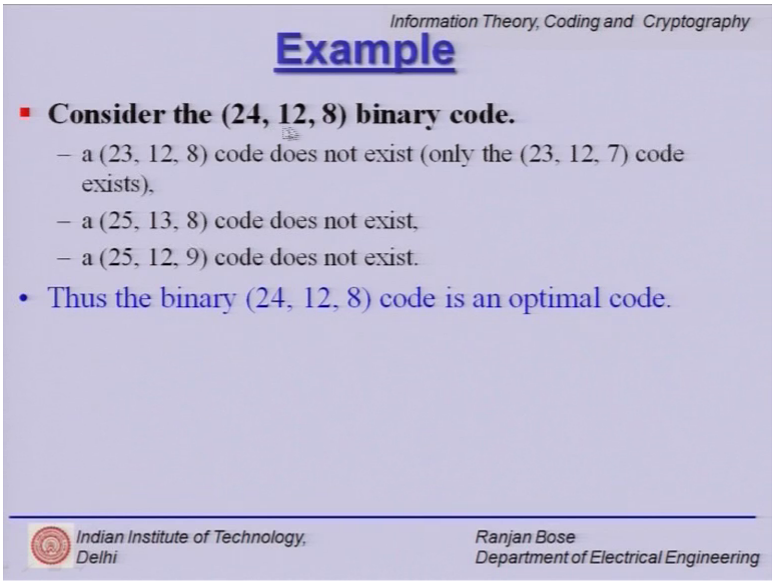
Ranjan Bose
Department of Electrical Engineering

Now, let us move a little bit ahead and we start defining an optimal code. So, what is an optimal code? An optimal n comma k comma d star code is such that none of it cannot be really improved how can you improve it well the three parameters n, k, d star n minus k represents the overhead. So, I can improve this n comma k comma d star code by ensuring that even if the d star is the same k is the same I can have less overhead. So, I can have n minus 1 comma k comma d star. So, if I can find the code which is n minus 1. So, it is a shorter code dock length is reduced without changing the number of information word and it stills gives me the same minimum distance, then I have improved upon the code or I said look d star I am not touching I can increase n to n plus 1 and increase k to k plus 1.

So, what does it mean that my code rate has become k plus 1 over n plus 1 and I can keep going and I can make it k plus 2 n plus 2. So, you can improve the code rate, right because that fraction goes closer to 1 by this do you agree that k plus 1 over n plus 1 is a closer to unity than k over n like 1 by 2 2 by 3, 3 by 4 so and so forth. So, your code rate keeps improving without decreasing your d star. So, again, you have an improvement or I said look I have n plus 1, I do not touch k, but that increase of 1 gives me d star plus 1 just as we did for that Hamming code that we had a 7 comma 4 code and we added a parity and the d star become larger or not.

So, if I can do so, if we have an n comma k comma d star code for which no such codes exists then this is an optimal code, ok; so none of the nearest improvements hold water such as an optimal code.

(Refer Slide Time: 43:24)



Information Theory, Coding and Cryptography

Example

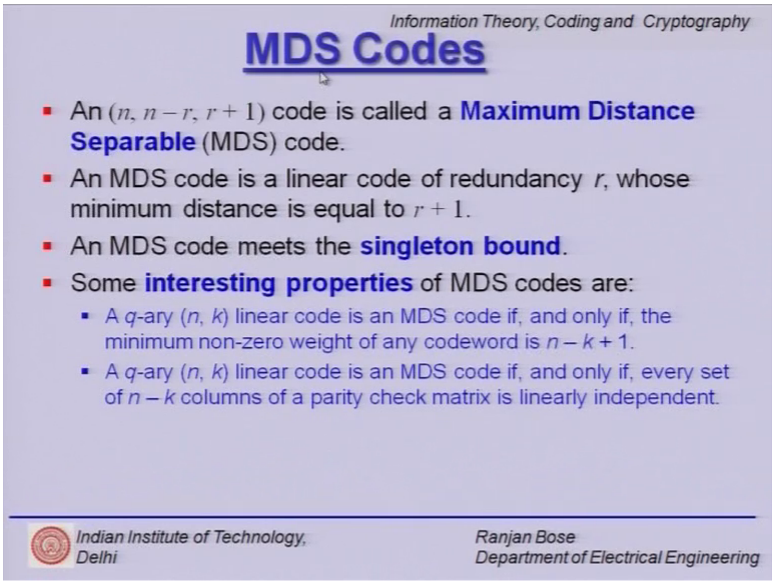
- **Consider the (24, 12, 8) binary code.**
  - a (23, 12, 8) code does not exist (only the (23, 12, 7) code exists).
  - a (25, 13, 8) code does not exist.
  - a (25, 12, 9) code does not exist.
- Thus the binary (24, 12, 8) code is an optimal code.

Indian Institute of Technology, Delhi

Ranjan Bose
Department of Electrical Engineering

For example, this 24 comma 12 comma 8 is a binary code which is optimal because you can check that if you reduce this n you cannot find you can disprove that 23 comma 12 comma 8 code it does not exist right this 25, 13. So, you increase it this code also does not exist because 24 plus 1, 25; 12 plus 1, 13 or 8. So, this also does not exist and this n plus 1 25, 12 and d is d star is written 9. This code also does not exist, you can take your time, prove it anyway, either by construction or you can prove in terms of the violation of the singleton bound whatever you want to do and you can show that these codes do not exist. And therefore, 24 comma 12 comma 8 is indeed an optimal code.

(Refer Slide Time: 44:21)



Information Theory, Coding and Cryptography

MDS Codes

- An $(n, n-r, r+1)$ code is called a **Maximum Distance Separable** (MDS) code.
- An MDS code is a linear code of redundancy r, whose minimum distance is equal to $r+1$.
- An MDS code meets the **singleton bound**.
- Some **interesting properties** of MDS codes are:
  - A q-ary (n, k) linear code is an MDS code if, and only if, the minimum non-zero weight of any codeword is $n-k+1$.
  - A q-ary (n, k) linear code is an MDS code if, and only if, every set of $n-k$ columns of a parity check matrix is linearly independent.

Indian Institute of Technology, Delhi

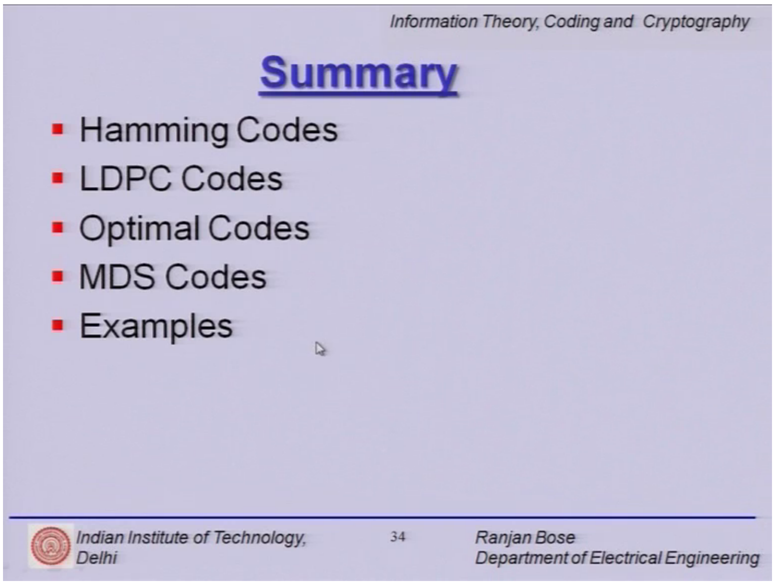
Ranjan Bose
Department of Electrical Engineering

Now, we come to one final definition which is the maximum distance separable codes which is the MDS code. How does this work? Well, we start from the redundancy, then finally, we are trying to correct the certain number of errors by adding a certain number of redundancy.

So, we say for a given redundancy r therefore, my code is n comma n minus r ok, k is equal to n minus r because the redundancies r we are adding r bits to k to make it n, but whose maximum whose minimum distance is equal to r plus 1. So, adding r redundancies you are having a minimum distance r plus 1 such a code is the MDS code, you can easily see that MDS codes meets the singleton bound by substituting this.

Some important in an interesting properties of MDS codes are that a q-ary n comma k. Linear code is an MDS code if and only if the minimum nonzero weight of any code word is n minus k plus 1 is one interesting property. Also, a q-ary n minus k linear code is an MDS code if and only if every set of n minus k columns of the parity check matrix is linearly independent. So, we can verify these two properties of MDS codes. So, these are so, we will see that binary MDS codes rarely exists we will have to go to q-ary codes for getting meaningful MDS codes and when we talk about Reed-Solomon codes we will find that they are indeed maximum distance separable codes.

(Refer Slide Time: 46:16)



Information Theory, Coding and Cryptography

Summary

- Hamming Codes
- LDPC Codes
- Optimal Codes
- MDS Codes
- Examples

Indian Institute of Technology, Delhi

34

Ranjan Bose
Department of Electrical Engineering

So, we come to the end of this lecture. Let us summarize what we have learnt today. We have covered Hamming codes and then we looked at the interesting class of codes called low density parity check codes, we then defined what are optimal codes and MDS codes, we also looked at some examples.

With that we come to the end of this lecture.