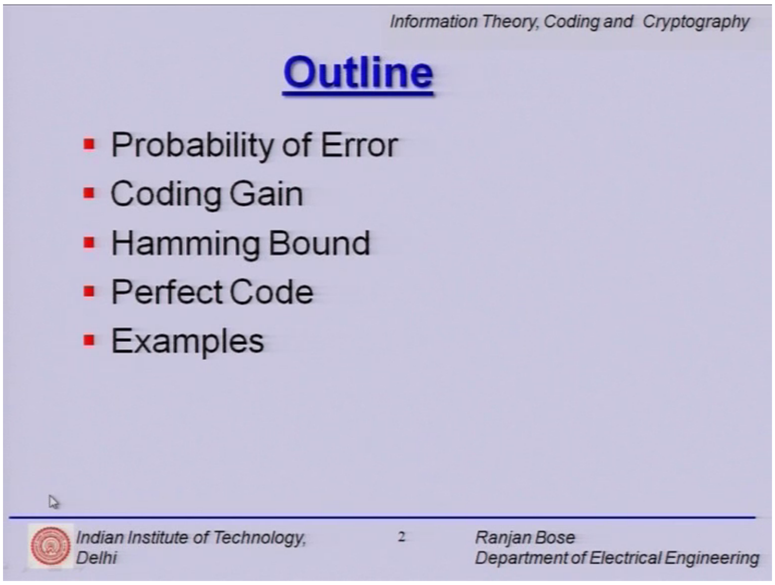
Information Theory, Coding and Cryptography

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

Module – 18
Linear Block Codes
Lecture – 18

Hello, and welcome to our next module on linear block codes.

(Refer Slide Time: 00:31)



Information Theory, Coding and Cryptography

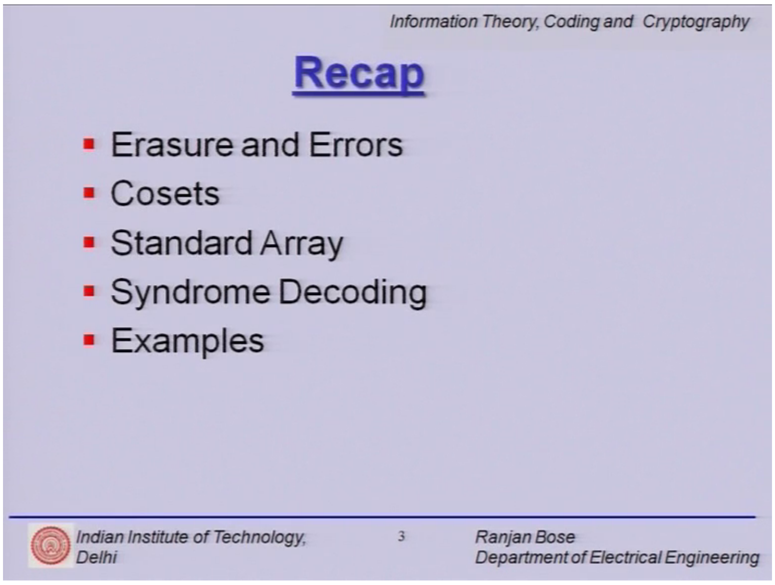
Outline

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

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

Let us start with a brief outline of today's talk. We will discuss a very important and practical aspect of linear block codes which is the probability of residual errors, we will also talk about coding gain; we look at hamming bound and look at something called as the notion of perfect codes.

(Refer Slide Time: 00:58)



Information Theory, Coding and Cryptography

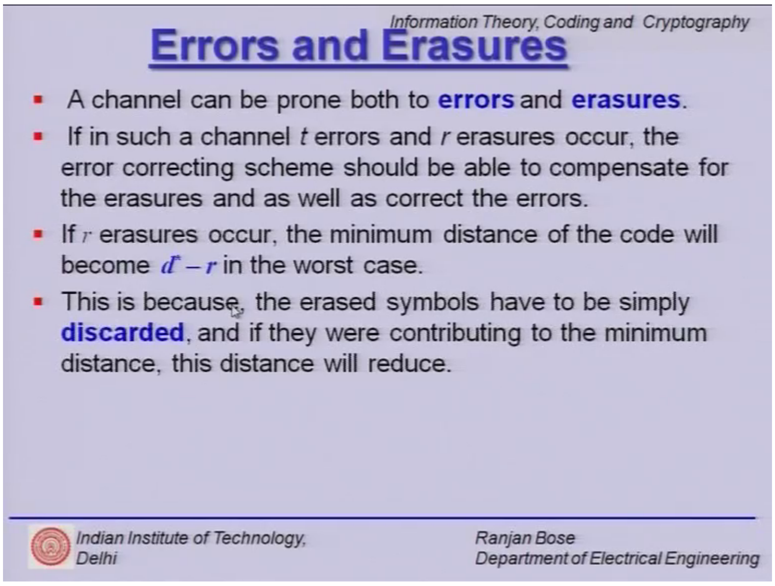
Recap

- Erasure and Errors
- Cosets
- Standard Array
- Syndrome Decoding
- Examples

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

But let us start with a quick recap what we have done so far. We have covered in the last lecture topics related to erasures, errors, cosets, standard arrays, syndrome decoding and we looked at several examples.

(Refer Slide Time: 01:12)



Information Theory, Coding and Cryptography

Errors and Erasures

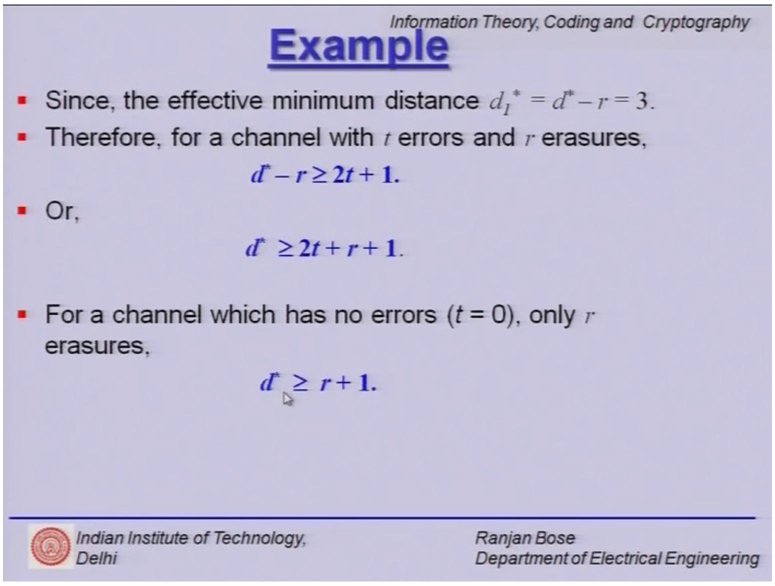
- A channel can be prone both to **errors** and **erasures**.
- If in such a channel t errors and r erasures occur, the error correcting scheme should be able to compensate for the erasures and as well as correct the errors.
- If r erasures occur, the minimum distance of the code will become $d^* - r$ in the worst case.
- This is because, the erased symbols have to be simply **discarded**, and if they were contributing to the minimum distance, this distance will reduce.

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

So, very quickly we have realized that a channel can have both errors and erasures and you can model such a channel with t errors and r erasures occurring for every codeword and question is how many of these can you actually correct ? So, if r erasures occur what it effectively does is it reduces the minimum distance of the code. You have r fewer symbols to work with and consequently in the worst case you are effective d star the minimum distance becomes d star minus 1.

So, the erased symbols have to be simply discarded.

(Refer Slide Time: 01:54)



Information Theory, Coding and Cryptography

Example

- Since, the effective minimum distance $d_I^* = d^* - r = 3$.
- Therefore, for a channel with t errors and r erasures,

$$d^* - r \geq 2t + 1.$$

- Or,

$$d^* \geq 2t + r + 1.$$

- For a channel which has no errors (t = 0), only r erasures,

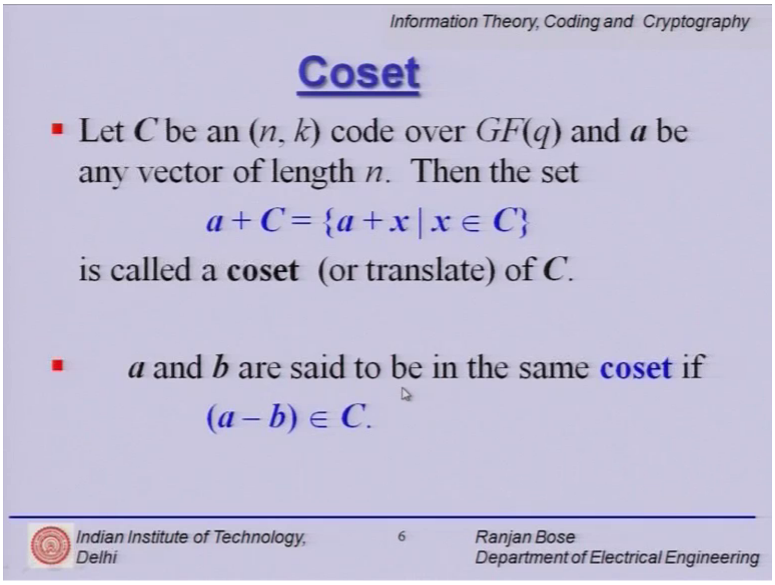
$$d^* \geq r + 1.$$

Indian Institute of Technology, Delhi

Ranjan Bose
Department of Electrical Engineering

We looked at a simple example where we saw that in order to correct for t errors you now must have d star minus r which is the effective minimum distance greater than or equal to 2 t plus 1 or you can have d star greater than or equal to 2 t plus r plus 1. So, your erasures eat into the number of errors you can correct, ok. Suppose, your channel does not have any errors it has only r erasures then d star should be greater than r plus 1.

(Refer Slide Time: 02:32)



Information Theory, Coding and Cryptography

Coset

- Let C be an (n, k) code over $GF(q)$ and a be any vector of length n. Then the set

$$a + C = \{a + x \mid x \in C\}$$

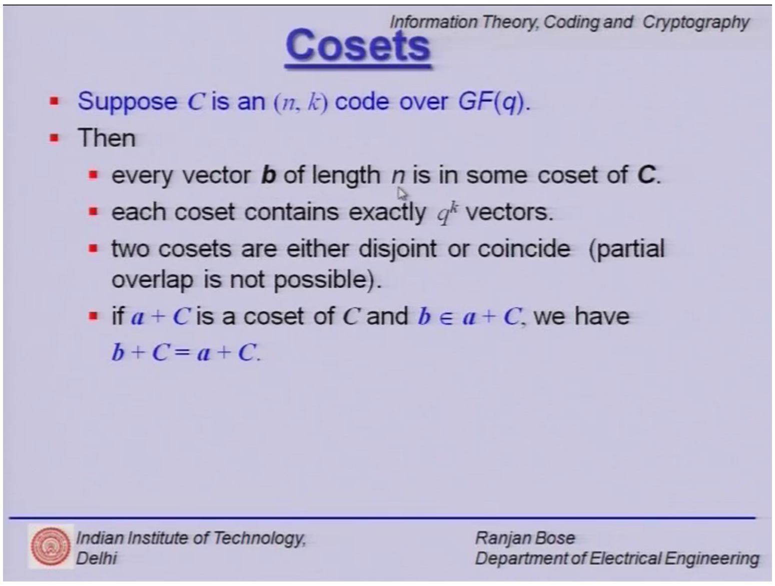
is called a **coset** (or translate) of C.

- a and b are said to be in the same **coset** if $(a - b) \in C$.

Indian Institute of Technology, Delhi 6 Ranjan Bose Department of Electrical Engineering

Now, we also looked at cosets and we defined for an n comma k code over GF q if a is any vector of length n; what is n? n is the codeword length also called as the block length of the linear block code then a plus C is called the coset or the translate of C and a and b are said to in the same coset if a minus b is an element of C.

(Refer Slide Time: 03:02)



Information Theory, Coding and Cryptography

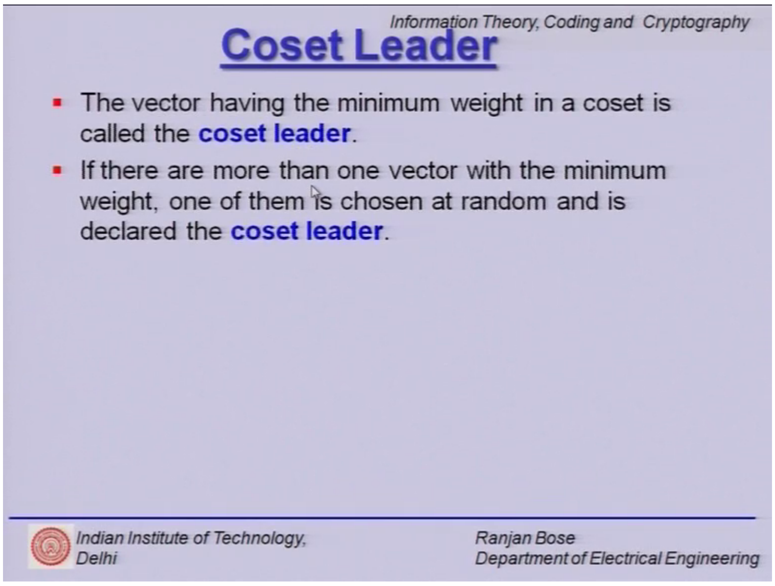
Cosets

- Suppose C is an (n, k) code over $GF(q)$.
- Then
  - every vector $\boldsymbol{b}$ of length n is in some coset of $\boldsymbol{C}$.
  - each coset contains exactly q^k vectors.
  - two cosets are either disjoint or coincide (partial overlap is not possible).
  - if $a + C$ is a coset of C and $b \in a + C$, we have $b + C = a + C$.

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

So, we have looked at some properties of the cosets and every vector b of length n is in the same coset of C. If we have each coset contains exactly q raised to the power k vectors 2 cosets are either disjoint or coincide and a plus C is a coset of C, if b is an element of a plus C then we have b plus C is equal to a plus C. So, we have already looked at these properties.

(Refer Slide Time: 03:15)



Information Theory, Coding and Cryptography

Coset Leader

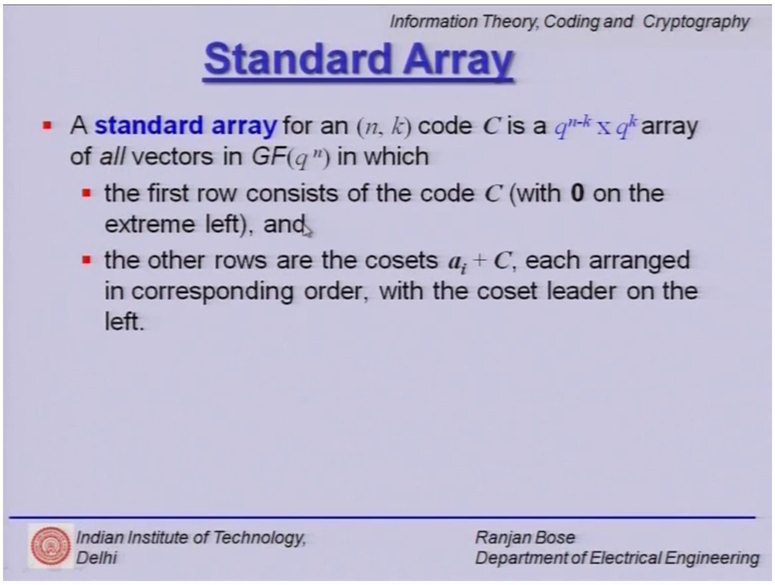
- The vector having the minimum weight in a coset is called the **coset leader**.
- If there are more than one vector with the minimum weight, one of them is chosen at random and is declared the **coset leader**.

Indian Institute of Technology, Delhi

Ranjan Bose
Department of Electrical Engineering

We then define the notion of a coset leader. What is a coset leader? It is a vector which has the minimum weight in a coset and it is called the coset leader. We will identify the error with that if there more than one vectors with minimum weight then one of them is chosen as random.

(Refer Slide Time: 03:52)



Information Theory, Coding and Cryptography

Standard Array

- A **standard array** for an (n, k) code C is a q^{n-k} x q^k array of *all* vectors in $GF(q^n)$ in which
  - the first row consists of the code C (with **0** on the extreme left), and
  - the other rows are the cosets $a_i + C$, each arranged in corresponding order, with the coset leader on the left.

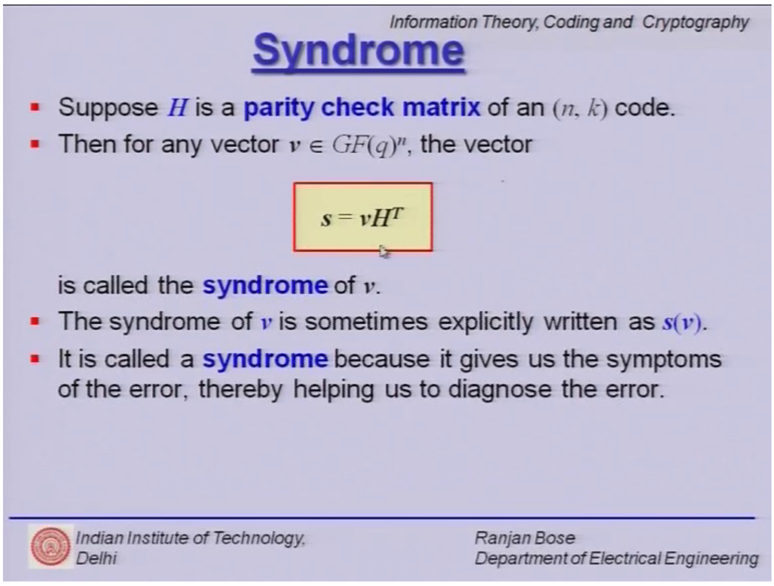
Indian Institute of Technology, Delhi

Ranjan Bose
Department of Electrical Engineering

What do we do with a coset? We form a standard array based on cosets. So, what is the standard array? The standard array for an n comma k code is nothing, but a table. How big is the table? It is q raised to the power n minus k cross q raised to the power k. So, this table the size of this table increases exponentially as k and n minus k increases.

So, the first row of the standard array consists of the actual codewords starting with the all 0 codeword on extreme left and all other rows are cosets.

(Refer Slide Time: 04:31)



Information Theory, Coding and Cryptography

Syndrome

- Suppose H is a **parity check matrix** of an (n, k) code.
- Then for any vector $v \in GF(q)^n$, the vector

$$\boxed{s = vH^T}$$

is called the **syndrome** of v.

- The syndrome of v is sometimes explicitly written as $s(v)$.
- It is called a **syndrome** because it gives us the symptoms of the error, thereby helping us to diagnose the error.

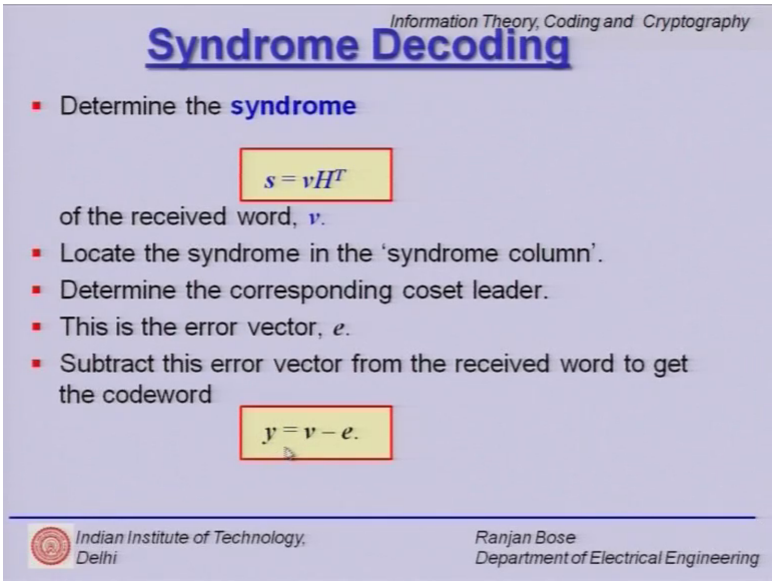
Indian Institute of Technology, Delhi

Ranjan Bose
Department of Electrical Engineering

.

Then we talked about something called as a syndrome decoding, for that we defined the syndrome first where if H is the parity check matrix of an n comma k code then any vector nu which is element of the GF q n is defined as s is equal to nu times H transpose. This s is called the syndrome of nu and sometimes it is explicitly written as s of nu. It is called a syndrome because it gives us the symptoms of the error, thereby diagnosing the error, ok. If you remember if this nu was the real codeword C the valid codeword C then CH transpose would be 0. So, the syndrome is 0 by definition for an actual codeword.

(Refer Slide Time: 05:28)



Information Theory, Coding and Cryptography

Syndrome Decoding

- Determine the **syndrome**

$$s = vH^T$$

of the received word, v.
- Locate the syndrome in the 'syndrome column'.
- Determine the corresponding coset leader.
- This is the error vector, e.
- Subtract this error vector from the received word to get the codeword

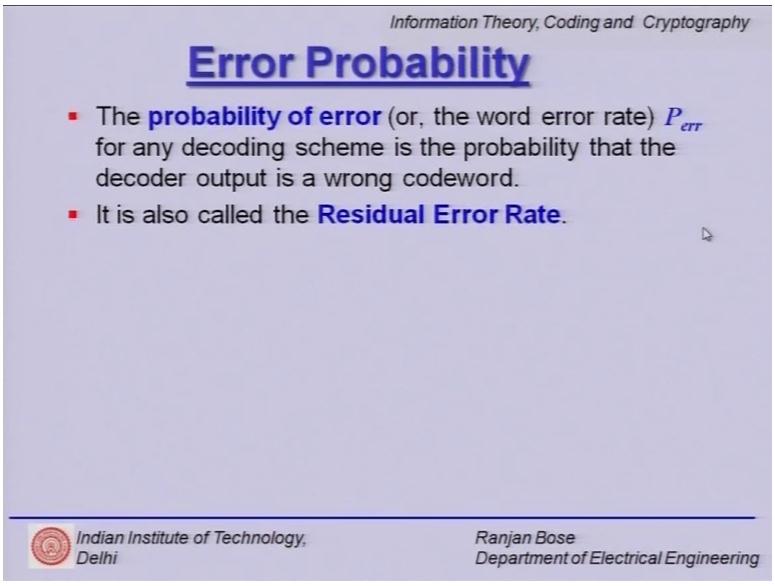
$$y = v - e.$$

Indian Institute of Technology, Delhi

Ranjan Bose
Department of Electrical Engineering

What do we mean by syndrome decoding? Well, first we all find out the syndrome of any received vector nu and then we locate the syndrome in the syndrome column and determine the corresponding coset leader which is supposed to give the error vector, e. We subtract this error vector e simply from the received vector to recover from the errors and give our correctly decoded word y.

(Refer Slide Time: 06:01)



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**.

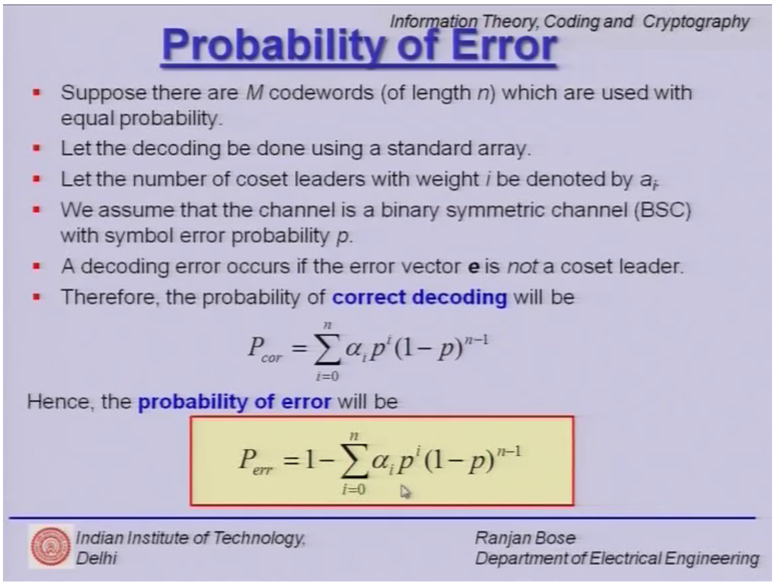
Indian Institute of Technology, Delhi

Ranjan Bose
Department of Electrical Engineering

Now, we talk about the probability of error. Now, please note the aim of linear block codes is to recover from errors, but they are not foolproof. They can correct for errors for which they have been defined for, they have been designed for a certain number of errors and they can only correct up to those many number of errors. So, there is a probability that more number of errors occur in a codeword more than what it has been designed for. In that case your error correcting code cannot recover and hence it leads to some residual probability of error.

So, the probability of error or P err for any decoding scheme is the probability that the decoder output is a wrong codeword. Despite our safety net, despite this mathematical insurance that we have put across we still can make errors this is called the residual error rate. What does it mean? Despite applying error control coding we still get errors and that is the residual error rate.

(Refer Slide Time: 07:10)



Information Theory, Coding and Cryptography

Probability of Error

- Suppose there are *M* codewords (of length *n*) which are used with equal probability.
- Let the decoding be done using a standard array.
- Let the number of coset leaders with weight *i* be denoted by a_i.
- We assume that the channel is a binary symmetric channel (BSC) with symbol error probability *p*.
- A decoding error occurs if the error vector **e** is *not* a coset leader.
- Therefore, the probability of **correct decoding** will be

$$P_{cor} = \sum_{i=0}^{n} \alpha_i p^i (1-p)^{n-1}$$

Hence, the **probability of error** will be

$$\boxed{P_{err} = 1 - \sum_{i=0}^{n} \alpha_i p^i (1-p)^{n-1}}$$

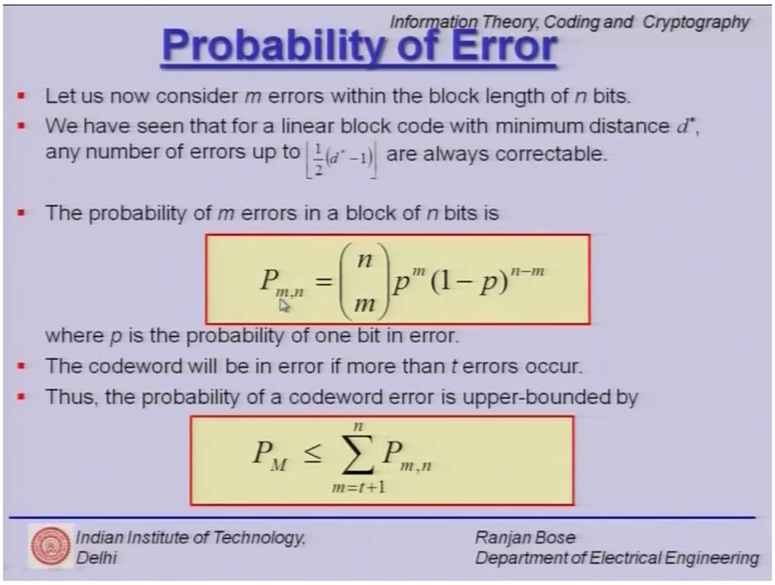
Indian Institute of Technology, Delhi

Ranjan Bose
Department of Electrical Engineering

Now, let us look at the probability of error. Suppose, there are M codewords each one of length n, we are talking about a linear block code of block length n these codewords are used with equal probability, then suppose you do a standard array based decoding and please note this analysis will hold true for syndrome decoding as well. Let the number of coset leaders with weight i be denoted by a i we assume that there is a binary symmetric channel with symbol error probability small p. So, our decoding error occurs if the error vector is not a coset leader because that is how we defined error correction based on standard arrays that the coset leader is this is designated as that error and we correct for that error.

So, the probability of correct decoding will be given by P correct as a summation i is equal to 1 through n alpha i p i 1 minus p n minus 1. So, probability of incorrect detection the probability of error is just 1 minus the probability of correct detection.

(Refer Slide Time: 08:26)



Information Theory, Coding and Cryptography

Probability of Error

- Let us now consider *m* errors within the block length of *n* bits.
- We have seen that for a linear block code with minimum distance d^*, any number of errors up to $\left\lfloor \frac{1}{2}(d^*-1) \right\rfloor$ are always correctable.
- The probability of *m* errors in a block of *n* bits is

$$\boxed{P_{m,n} = \binom{n}{m} p^m (1-p)^{n-m}}$$

where *p* is the probability of one bit in error.

- The codeword will be in error if more than *t* errors occur.
- Thus, the probability of a codeword error is upper-bounded by

$$\boxed{P_M \le \sum_{m=t+1}^{n} P_{m,n}}$$

Indian Institute of Technology, Delhi

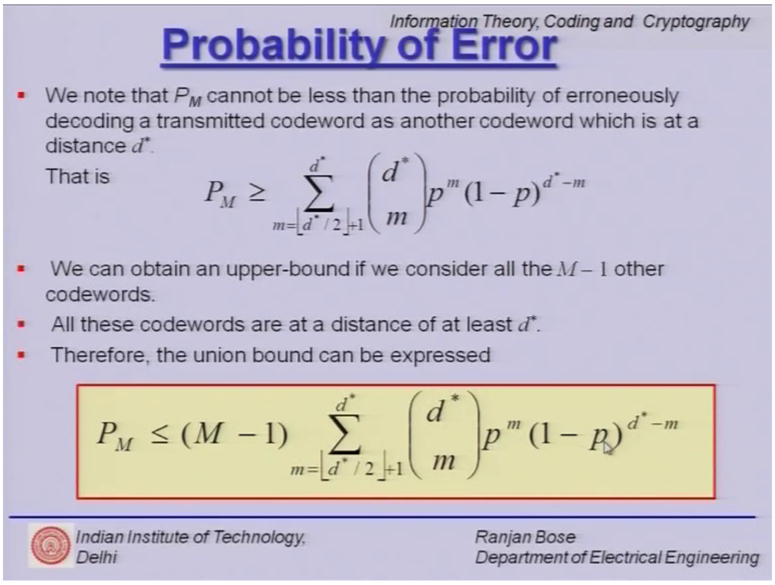
Ranjan Bose
Department of Electrical Engineering

So, let us now have m errors in the block length of n bits. So, n is necessarily less than or equal to n. So, how many errors can be correct? Well, it depends on the minimum distance of the code d star. So, we know that the number of errors that are correctable are up to this integer less than or equal to half d star minus 1 this comes from the fact that d star should be greater than or equal to 2 t plus 1 for correcting t errors. So, we will flip it around and we will put that number of correctable errors.

The probability of m errors in a block of n bits is given by n choose m because m errors are there this is the probability making one error. So, we have m errors. So, p raised to the power m and at the same time the n minus m symbols must not being error. So, probability of not being error is 1 minus p. So, you have 1 minus p raised to the power n minus m this is the probability m comma n which means the probability of making m errors out of the n possible symbols.

The codeword will be in error if more than t errors occur with t is the designed error correcting capability of that code. So, how do we upper bound the probability of codeword error, it is given simply by all the cases, where m is t plus 1 or more. Here we are found out what is the probability that m errors occur in a codeword of length n. Now, up to m is equal to t errors can be corrected, but t plus 1 up to n all such errors cannot be corrected by our linear block code whose d star is such that d star is greater than or equal to 2 t plus 1. So, we have this upper bound by summation over all m's from t plus 1 up to n.

(Refer Slide Time: 10:48)



Information Theory, Coding and Cryptography

Probability of Error

- We note that P_M cannot be less than the probability of erroneously decoding a transmitted codeword as another codeword which is at a distance d^*.

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

- We can obtain an upper-bound if we consider all the $M-1$ other codewords.
- All these codewords are at a distance of at least d^*.
- Therefore, the union bound can be expressed

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

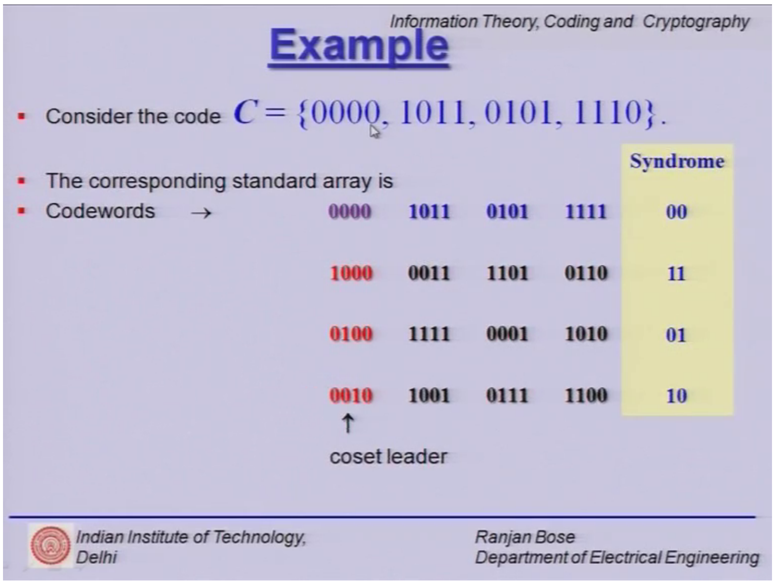
Indian Institute of Technology, Delhi

Ranjan Bose
Department of Electrical Engineering

We note that P M cannot be less than the probability of erroneously decoding a transmitted codeword as another codeword which is at a distance d star. So, what we mean is that this P M right here m goes from d star by 2 plus 1 up to d star is the errors. So, d star choose m is the total number of combinations and then we can find the corresponding upper bound for M minus 1 code words. So, this approach says that we go codeword by codeword; there is a case where you have the closest codeword to be only d star away from this received vector.

Now, all codewords that are distance at least d star because there will be seven codewords which are more than d star, d star represents the worst case scenario. So, just look at the codewords that are exactly at a distance d star or more. So, when we have at least d star then we can put a union bond as follows; P M is less than or equal to M minus 1 m goes from d star by 2 plus 1 up to d star and these are the number of ways to choose m out of d star, probability of flipping a bit raised to the power m 1 minus p d star minus m.

(Refer Slide Time: 12:28)



Information Theory, Coding and Cryptography

Example

- Consider the code $C = \{0000, 1011, 0101, 1110\}$.
- The corresponding standard array is

					Syndrome
Codewords →	0000	1011	0101	1111	00
	1000	0011	1101	0110	11
	0100	1111	0001	1010	01
	0010	1001	0111	1100	10

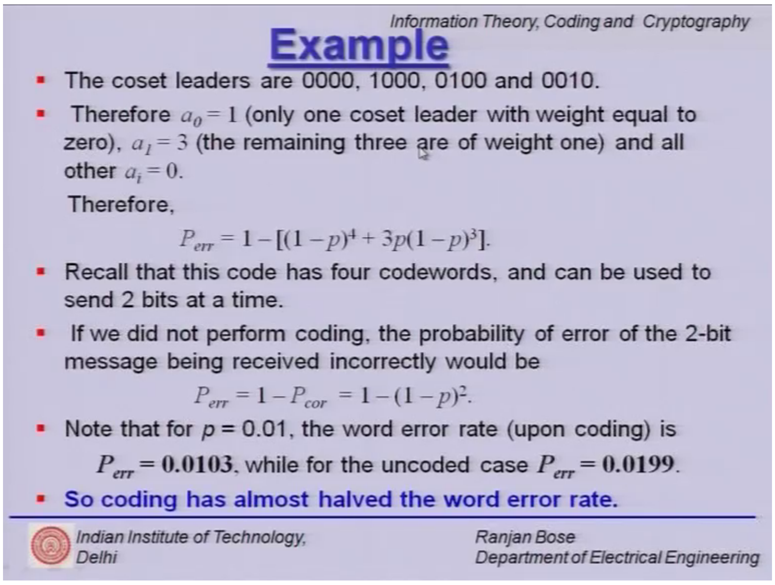
↑ coset leader

Indian Institute of Technology, Delhi

Ranjan Bose
Department of Electrical Engineering

So, let us look at a simple example. So, consider a simple example of four codewords in this code C 0000, 1011, 0101, and 1110. We build a standard array with it is own coset leaders, each row is a coset, the first row is the valid codewords and here we have a column of syndrome. So, if you receive any one of these four and you apply nu h transpose you will get a 0 because a syndrome for a valid codeword is always a 0 vector, but if you get any one of these four received vectors nu then nu h transpose will always result in 1 1 similarly if you get any one of these you will get 0 1 and for any one of this you will get 1 0. So, one syndrome corresponds to one of the coset leaders and these coset leaders are the error vectors.

(Refer Slide Time: 13:36)



Information Theory, Coding and Cryptography

Example

- The coset leaders are 0000, 1000, 0100 and 0010.
- Therefore $a_0 = 1$ (only one coset leader with weight equal to zero), $a_1 = 3$ (the remaining three are of weight one) and all other $a_i = 0$.

  Therefore,

  $$P_{err} = 1 - [(1-p)^4 + 3p(1-p)^3].$$
- Recall that this code has four codewords, and can be used to send 2 bits at a time.
- If we did not perform coding, the probability of error of the 2-bit message being received incorrectly would be

  $$P_{err} = 1 - P_{cor} = 1 - (1-p)^2.$$
- Note that for $p = 0.01$, the word error rate (upon coding) is

  $\boldsymbol{P_{err} = 0.0103}$, while for the uncoded case $\boldsymbol{P_{err} = 0.0199}$.
- **So coding has almost halved the word error rate.**

Indian Institute of Technology, Delhi

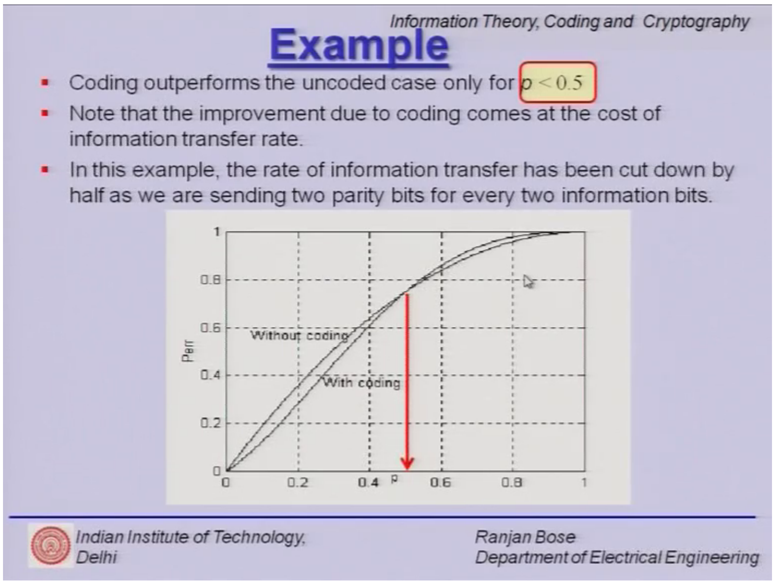
Ranjan Bose
Department of Electrical Engineering

So, you have now, these coset leaders a 0000, 1000, 0100 and 0010. So, therefore, if you plug in the values you can get the probability of error as 1 minus parenthesis open 1 minus p raised to the power 4 plus 3p 1 minus p raised to the power 3 we just plugged in the value for P error, right.

So, please note that this code has four codewords and consequently can be used to send out 2 bits at a time. So, what does it mean? Well, you have two options. Either you do coding or you do not. So, if you do not perform coding the probability of error of 2-bit message being incorrectly received is nothing, but P error is equal to 1 minus P correct is 1 minus 1 minus p whole squared. So, 2 bits being received incorrectly. So, if you have small p which is flipping of a bit as 0.01 the word error rate upon coding is given by 0.0103 is just plug in the value, while for the uncoded case it is 0.0199.

So, if you see we have almost halved the word error rate by the simple very simple example of the code that we discussed in the previous example. If you note this d star for this is 2 if you look at this code then the minimum weight is 2 and hence d star is 2. So, it is almost like a useless code, but if you see it has been able to give you a big improvement in the probability of error, residual error probability.

(Refer Slide Time: 15:45)



Information Theory, Coding and Cryptography

Example

- Coding outperforms the uncoded case only for $p < 0.5$
- Note that the improvement due to coding comes at the cost of information transfer rate.
- In this example, the rate of information transfer has been cut down by half as we are sending two parity bits for every two information bits.

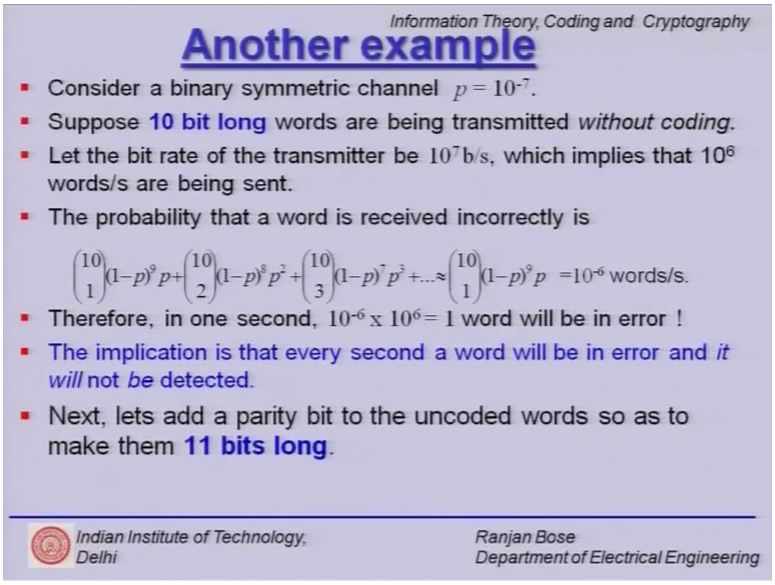
Indian Institute of Technology, Delhi

Ranjan Bose
Department of Electrical Engineering

So, let us note this thing. If you look at this graph on the x axis is small p on the y axis is probability of error and there are two curves here; one is without coding and one is with coding. Below 0.5 below p is equal to 0.5; we see that with coding is not that effective only when you go beyond 0.5 does the coding. So, below 0.5 coding has more effect whereas, if the p is much larger than the coding is counterproductive.

So, you should have the channel of certain level for coding to be effective. You just cannot take any channel and apply coding and expect it to be better. So, 0.5 is that tipping point before 0.5 below 0.5 yes, error control coding works beyond that it is detrimental.

(Refer Slide Time: 17:01)



Information Theory, Coding and Cryptography

Another example

- Consider a binary symmetric channel $p = 10^{-7}$.
- Suppose **10 bit long** words are being transmitted *without coding.*
- Let the bit rate of the transmitter be 10^{7} b/s, which implies that 10^{6} words/s are being sent.
- The probability that a word is received incorrectly is

$$\binom{10}{1}(1-p)^{9}p+\binom{10}{2}(1-p)^{8}p^{2}+\binom{10}{3}(1-p)^{7}p^{3}+\ldots\approx\binom{10}{1}(1-p)^{9}p \;=10^{-6}\text{ words/s.}$$

- Therefore, in one second, $10^{-6} \text{ x } 10^{6} = 1$ word will be in error !
- The implication is that every second a word will be in error and *it will* not *be* detected.
- Next, lets add a parity bit to the uncoded words so as to make them **11 bits long**.

Indian Institute of Technology, Delhi

Ranjan Bose
Department of Electrical Engineering

Let us look at another quick example. So, let us say we have a binary symmetric channel with small p is equal to 10 raised to the power minus 7. What does it mean, that this channel is making a mistake when you sent 10 chose 7 bits on an average one bit gets flipped. So, it as said it is a pretty low probability of error.

Now, suppose we sent 10 bit long words at a time without coding. So, without introducing any error control coding, right and you know any standard modem it will it will have it is own data rates. So, suppose this data rate is 10 raised to the power 7 bits per second which means 10 raised to the power 6 words are being sent. Why because each word is 10 bit long. So, this the simple example.

So, what is the probability that a word is received incorrectly? Well, you can find out what is the probability that word is received incorrectly either there is one error. So, 10 choose 1, 1 minus p raised to the power 9; that means, 9 of them are not in error p is the probability of error, p raised to the power 1; 1 is an error or so, it is a mutually exclusive event that two errors happen. How can two errors happen 10 choose 2 either the number of ways I can have 2 errors out of 10 then 1 minus p is the probability of correct symbols raised power 8 because 8 of them are correct and p is wrong.

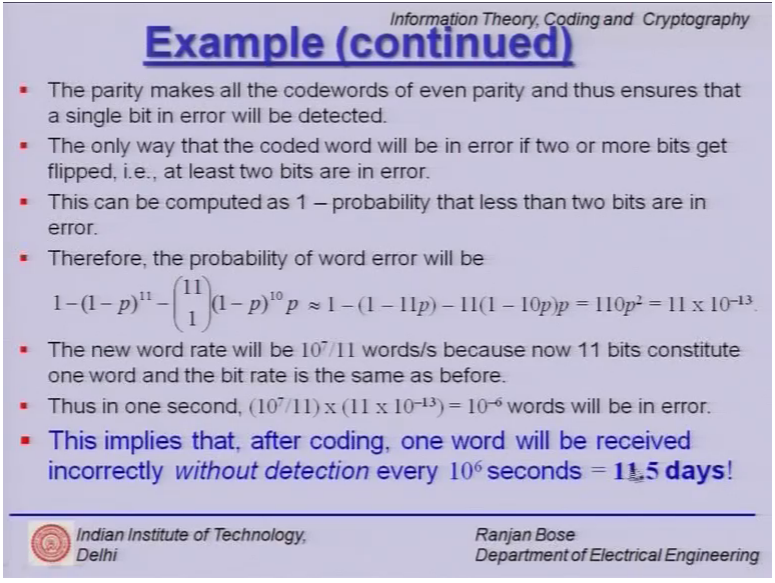
So, p raised to the power 2; 2 of them are incorrect or I have 3 possibilities or so and so forth. As you can see that p is already very small. So, p square is even smaller and p cube is even smaller whereas, 1 minus p is almost close to 1 and can be neglected.

So, we see from this that only the first term is significant and all subsequent terms are orders of magnitude smaller than the previous one. So, approximately we can retain only the first term 10 choose 1; 1 minus p raised to the power 9 times p and if you plug in the value very approximately you can see that this is 10 the first term is 1 and this is 10 raised to the power minus 7. So, you get 10 raised to the power minus 6 words per second,.

Therefore, in one second 10 raised to the power minus 6 and you are sending 10 raised to the power 6 words per second will be in error. So, every second one word is in error, and what is the problem with this is that even if a word is an error it will not be detected because if we are not doing any error control coding any bit stream of 10 bit long is a valid bit stream, but I do not even know whether an error has happened or not. So, one word will be in error over this quite good channel p is equal to 10 raised to the power minus 7 is a pretty good channel, but we will not even know will not even not even will be able to detect the error even though one word every second is an error.

So, now what we do is we add one extra bit. So, this example illustrates the power of one bit, just add one parity bit to the uncoded word to make it 11 bit long as opposed to 10 bit.

(Refer Slide Time: 20:53)



Information Theory, Coding and Cryptography

Example (continued)

- The parity makes all the codewords of even parity and thus ensures that a single bit in error will be detected.
- The only way that the coded word will be in error if two or more bits get flipped, i.e., at least two bits are in error.
- This can be computed as 1 – probability that less than two bits are in error.
- Therefore, the probability of word error will be
$$1-(1-p)^{11}-\binom{11}{1}(1-p)^{10}p \approx 1-(1-11p)-11(1-10p)p = 110p^2 = 11 \times 10^{-13}.$$
- The new word rate will be $10^7/11$ words/s because now 11 bits constitute one word and the bit rate is the same as before.
- Thus in one second, $(10^7/11) \times (11 \times 10^{-13}) = 10^{-6}$ words will be in error.
- This implies that, after coding, one word will be received incorrectly *without detection* every 10^6 seconds = **11.5 days**!

Indian Institute of Technology, Delhi

Ranjan Bose
Department of Electrical Engineering

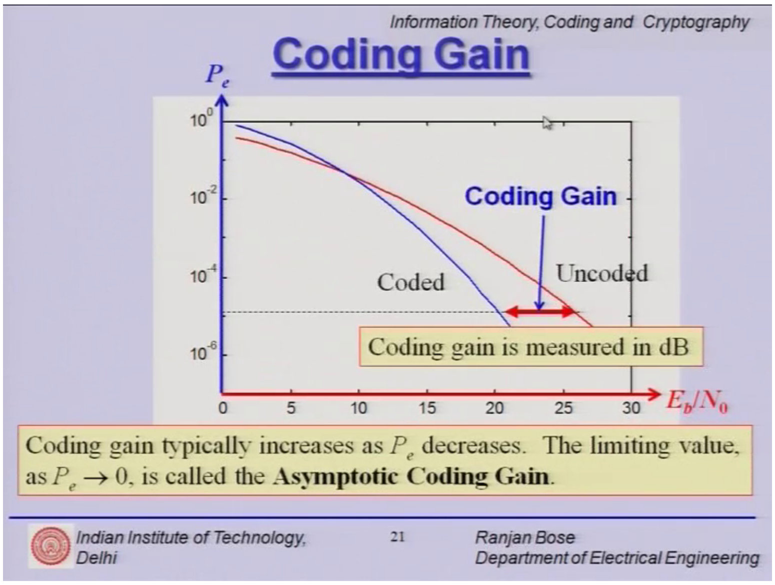
So, what does it do? What does the parity bit do? Parity bit says that one error will be detected, and even parity will become odd and vice versa. So, what will be the case when there are two or more errors well that is when the error will happen if two or more bits gets flipped then errors will happen.

So, how do we calculate the probability of error for this? Well, it is 1 minus probability that less than 2 bits are in error. So, the probability of word error can be written as follows. So, two or more bits are in error and if you write it out like this one minus all of them are in error or two. So, you have got this expression and you get 11 into 10 raised to the power 13, if you work this out.

So, the new word rate because this one will be able to detect all errors with one bit being flipped. So, the new word error rate will be 10 raised to the power 7 by 11 words per second because now 11 bits constitute one word right and the bit rate has not changed. So, thus in one second this is the effective word rate into the number of words in error per second will give you 10 power 6 words will be in error per second.

So, how many seconds will it take for one word to go undetected with error? It will take 10 raised to the power 6 seconds and which is 11 point 5 days earlier we were making a mistake every second and not even knowing about it. Now, we will have that problem once in every 11.5 days. So, it is really improved the error detection capability. So, this is the power of one bit. No strong coding, no complicated algebra just adding one parity bit has improved the performance tremendously.

(Refer Slide Time: 23:10)



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 $P_e \to 0$, is called the **Asymptotic Coding Gain**.

Indian Institute of Technology, Delhi

21

Ranjan Bose
Department of Electrical Engineering

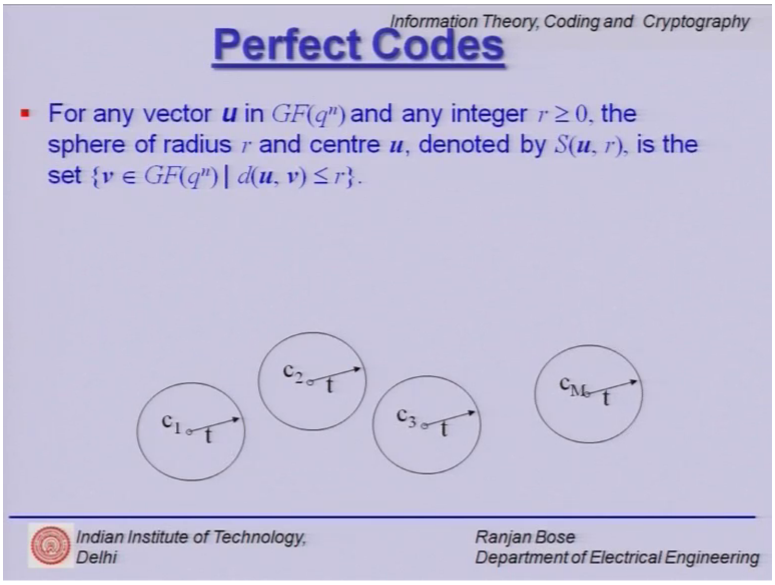
So, let us look at this very interesting concept called coding gain because we are gaining. So, much is there a way to quantify it how do we put it across. So, let us have our two most important parameters one is SNR which is E b over N naught and the other axis is my probability of error P e. We would like to define how much gain do we have based on error control coding. So, if you look at the graph here we have on the x axis the SNR E b over N naught, the y axis P e and we have the blue curve corresponding to the coding. So, coded bit stream whereas, the red one is the raw or uncoded bit stream.

And, you will see that for any given probability of error say close to 10 raised to the power minus 5 you can always draw a horizontal line and what you see is that uncoded versus coded you get again in terms of E b over N naught. This horizontal spacing is called the coding gain and clearly the units of E b over N naught is in dB.

So, the coding gain is expressed in dB if I am have a much stronger error control coding scheme I can have this blue curve shift even to the left and I can have a larger coding gain. But, this coding gain is a function of P e. So, only at 10 power minus 5 in this example I have this much coding gain if I have 10 power minus 4 the coding gain goes down and a 10 raised to the power minus 6 the coding gain is even higher.

So, coding gain is measured in dB and coding gain typically increases as P e decreases as we have seen just now P e the probability of error and a limiting value as P e tends to 0 it is very very low, it is called the asymptotic coding gain. So, that is the theoretical maximum coding gain that one can achieve. So, for coded systems coding gain is typically used to compare which one is better.

(Refer Slide Time: 25:47)



Information Theory, Coding and Cryptography

Perfect Codes

- For any vector $\boldsymbol{u}$ in $GF(q^n)$ and any integer $r \geq 0$, the sphere of radius r and centre $\boldsymbol{u}$, denoted by $S(\boldsymbol{u}, r)$, is the set $\{\boldsymbol{v} \in GF(q^n) \mid d(\boldsymbol{u}, \boldsymbol{v}) \leq r\}$.

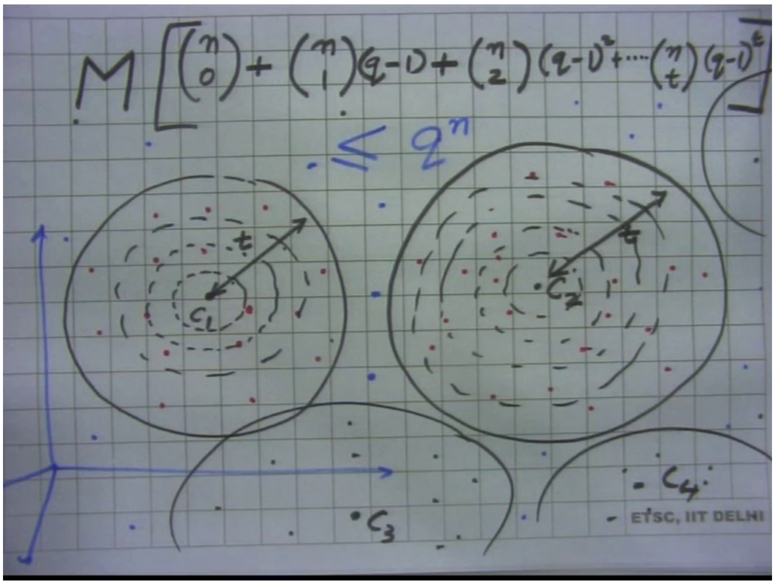
Indian Institute of Technology, Delhi

Ranjan Bose
Department of Electrical Engineering

Now, we introduce this notion of perfect codes. So, this is like reaching one of the boundaries one of the ends of the spectrum ah. Suppose, for any vector u over GF q n and any integer r greater than or equal to 0 the sphere of radius r and the center u denoted by S u comma r is the set new element of GF q n given that the distance between u and nu is less than r.

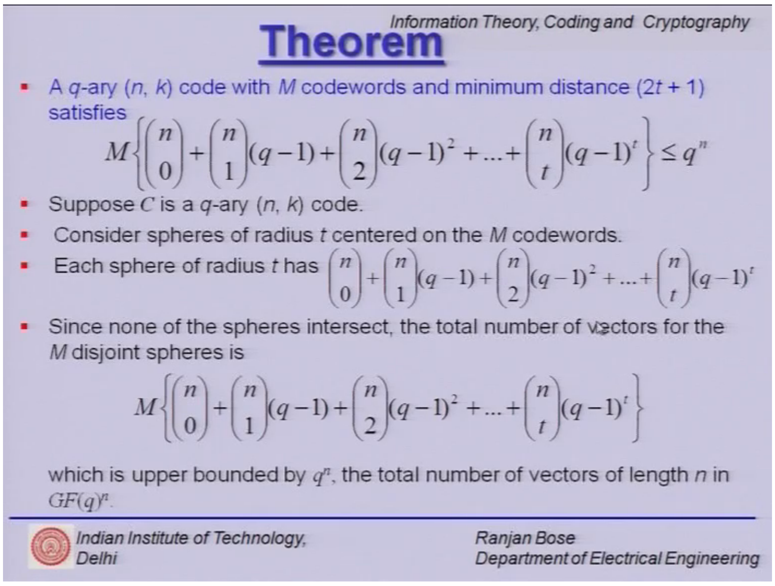
So, let us understand what is this thing in Greek, let us put it back in English. So, we have a code space and every point in this code space is a vector of length n please note this is not necessarily binary we are talking about GF q n all right. Here, we have talked about the notion of radius of spheres what are these spheres. So, within this code space centered at any codeword c 1, we can draw a sphere a hyper sphere which has several points inside it, but again each point is a vector of length n each of the points inside the sphere will have a certain distance in terms of hamming distance from that codeword c 1, and we are talking about oh, ok

(Refer Slide Time: 27:25)


$M\left[\binom{n}{0}+\binom{n}{1}(q-1)+\binom{n}{2}(q-1)^2+\cdots\binom{n}{t}(q-1)^t\right]$
$\leq q^n$
t
c_1
t
c_2
c_3
c_4
ETSC, IIT DELHI

So, far we have accounted for all the points in M spheres, but what is this total number of points all the red dots and the blue dots and the black dots. Well, all of them should necessarily be less than or equal to q raised to the power n which is the total number of vectors in this space and that is exactly what we have on our slide here.

(Refer Slide Time: 28:01)



Information Theory, Coding and Cryptography

Theorem

- A q-ary (n, k) code with M codewords and minimum distance $(2t + 1)$ satisfies

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

- Suppose C is a q-ary (n, k) code.
- Consider spheres of radius t centered on the M codewords.
- Each sphere of radius t has $\binom{n}{0}+\binom{n}{1}(q-1)+\binom{n}{2}(q-1)^2+\ldots+\binom{n}{t}(q-1)^t$
- Since none of the spheres intersect, the total number of vectors for the M disjoint spheres is

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

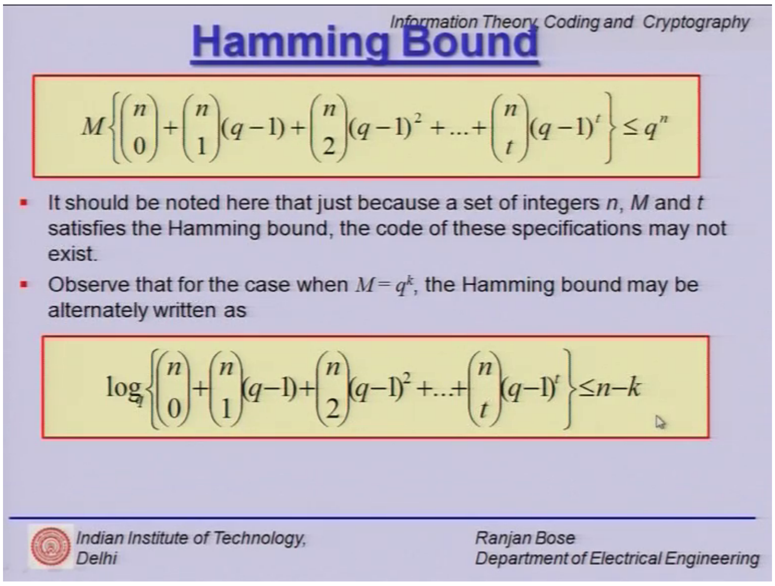
which is upper bounded by q^n, the total number of vectors of length n in $GF(q)^n$.

Indian Institute of Technology, Delhi

Ranjan Bose
Department of Electrical Engineering

So, if you go back to this slide we have on the left hand side M multiplied by n choose 0 plus n choose 1 q minus 1 and so on and so forth less or equal to the total number of vectors q raised to the power n in this space,. So, this is exactly what is being discussed here, right.

(Refer Slide Time: 28:27)



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

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

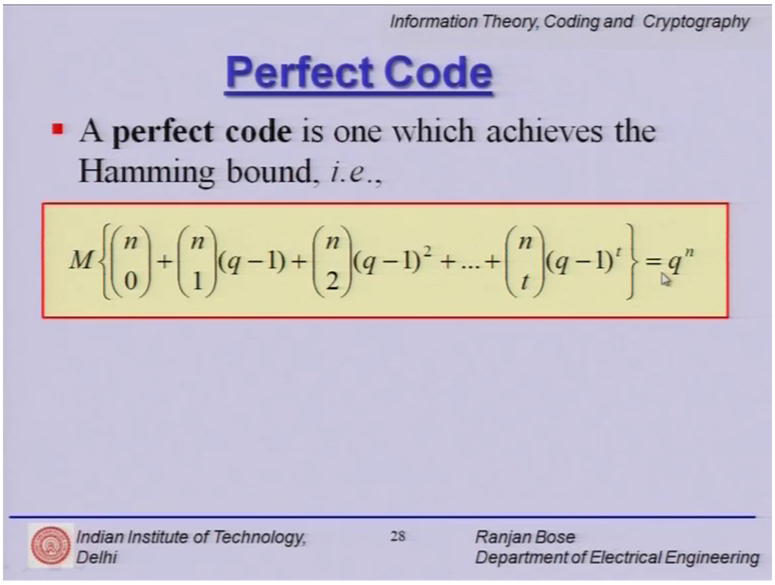
Indian Institute of Technology, Delhi

Ranjan Bose
Department of Electrical Engineering

Now, this bound this upper bound is denoted by a hamming bound and this is also termed as a sphere packing bound simply by the way in which this was derived very geometrically,. So, the total number of points contained in all the spheres decoding spheres of m codewords is less than equal to q raised to the power n the total number of points in that space.

Now, this is in general for q-ary codes, but for binary code things become slightly simpler and you can always put q as 2 and all of these parentheses q minus 1 goes away and you get this kind of a an expression. Please recall that M is nothing, but 2 raised to the power k. So, this M can be substituted by 2 raised to the power k in binary case, but if this is the in general and in if you take a log on both sides then you can write this is M is q raised to the power k this is q raised to the power M. So, n minus k will be on this right hand side and you have log to the base q and you have this expression. So, it can alternately be written as follows.

(Refer Slide Time: 29:55)



Information Theory, Coding and Cryptography

Perfect Code

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

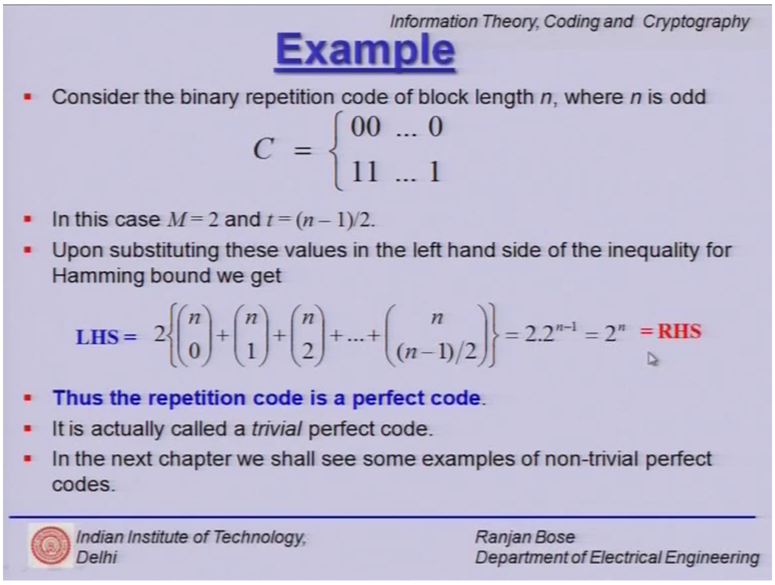
$$\boxed{M\left\{\binom{n}{0}+\binom{n}{1}(q-1)+\binom{n}{2}(q-1)^2+\ldots+\binom{n}{t}(q-1)^t\right\}=q^n}$$

Indian Institute of Technology, Delhi 28 Ranjan Bose Department of Electrical Engineering

So, now we define something called as a perfect code because we are q raised to c whether this upper bound can be met or not and what if it is met. So, we define a perfect code as one which achieve which achieves the hamming bound where equality holds. So, m which is alternately q raised to the power k times n choose 0 plus n choose one q minus 1 plus so and so forth up to n choose t q minus 1 raised to the power t should be equal to q raised to the power n. Any code that satisfies this is the hamming bound.

Now, please note m, n, q are all even t are all integers. So, you cannot just have any arbitrary choice of m, n, q and t which will make this happen. In fact, there could be only few combinations of these integers m, n, q and t which satisfy this condition.

(Refer Slide Time: 31:05)



Information Theory, Coding and Cryptography

Example

- Consider the binary repetition code of block length *n*, where *n* is odd

$$C = \begin{cases} 00\ \ldots\ 0 \\ 11\ \ldots\ 1 \end{cases}$$

- In this case $M = 2$ and $t = (n-1)/2$.
- Upon substituting these values in the left hand side of the inequality for Hamming bound we get

$$\textbf{LHS} = 2\left\{\binom{n}{0}+\binom{n}{1}+\binom{n}{2}+\ldots+\binom{n}{(n-1)/2}\right\} = 2.2^{n-1} = 2^n \ = \textbf{RHS}$$

- **Thus the repetition code is a perfect code.**
- It is actually called a *trivial* perfect code.
- In the next chapter we shall see some examples of non-trivial perfect codes.

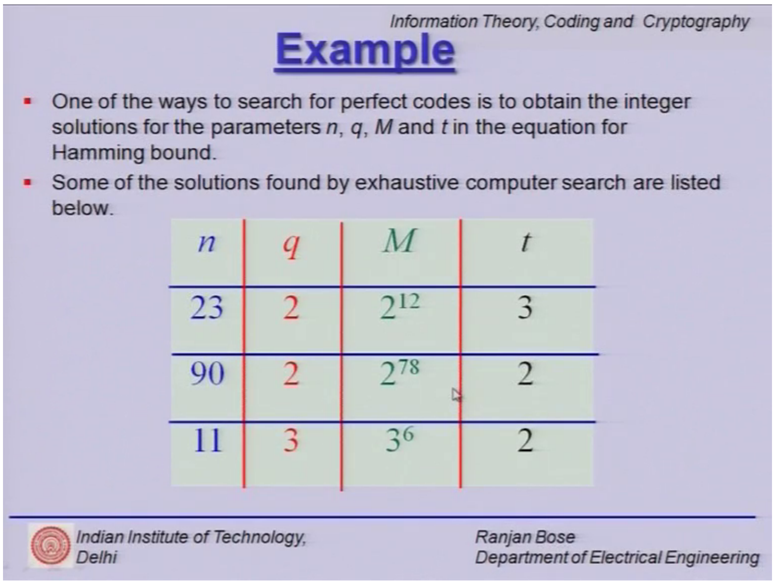
Indian Institute of Technology, Delhi

Ranjan Bose
Department of Electrical Engineering

So, let us look at a very simple example this is our favorite binary repetition code of block length n and we have taken n as odd. So, when 0 is to be sent 0 is sent as 0 0 0 0 0 repeated n times, if 1 has to be sent as 1 1 1 1 up to n times. Clearly the number of errors it can correct is n minus 1 by 2 n being odd I do not have to put any other restrictions. Why? Well, the minimum distance is n d star. Obviously, my observation is n clearly m is equal to 2, 0 and 1 are the two information words which are being sent.

Now, we put this constraint and try to check whether it satisfies the hamming bound. Well, we put left hand side and we find magically we get this repetition code to satisfy the hamming bound. Thus the repetition code are trivial code is also a perfect code, it is a trivial perfect code, all right.

(Refer Slide Time: 32:16)



Information Theory, Coding and Cryptography

Example

- One of the ways to search for perfect codes is to obtain the integer solutions for the parameters *n*, *q*, *M* and *t* in the equation for Hamming bound.
- Some of the solutions found by exhaustive computer search are listed below.

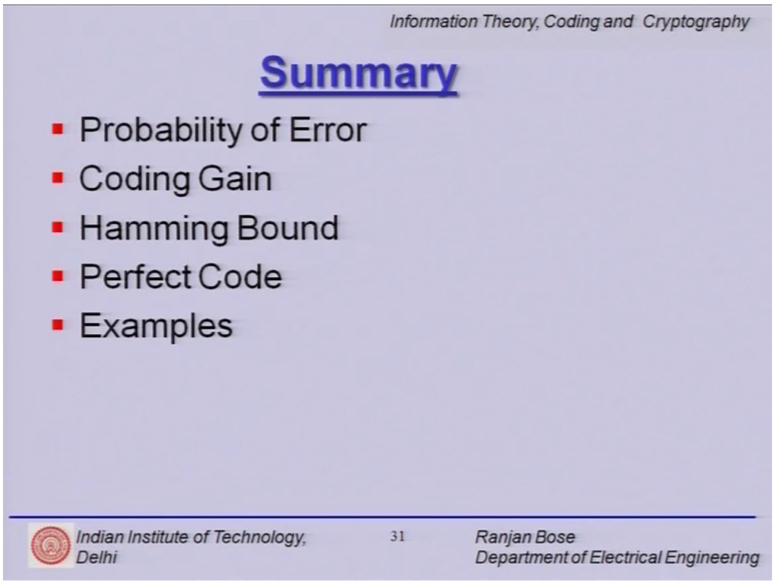
n	q	M	t
23	2	2^{12}	3
90	2	2^{78}	2
11	3	3^{6}	2

Indian Institute of Technology, Delhi

Ranjan Bose
Department of Electrical Engineering

So, one of the ways to search for this profit codes is to run a computer simulation to check for various values of n, q, M and t and see what satisfies the equation and here we have got listed out some typical values found by exhaustive computer search, but please note just because the values exist that satisfy the hamming bound does not mean that a code exists. So, not for all of these cases a code can be identified which satisfies this condition.

(Refer Slide Time: 32:54)



Information Theory, Coding and Cryptography

Summary

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

Indian Institute of Technology, Delhi

31

Ranjan Bose
Department of Electrical Engineering

So, with that we come to the end of this lecture let us summarize what we have studied so far. What we have understood is the notion of the residual probability of error. Even if you incorporate error control coding you will you can still make errors and how do we quantify that. So, we came up with some expressions and upper bound on the probability of residual error based on d star. Then we talked about coding gain and how it is measured in dB, then we established a very important bound called the hamming bound which led us to the definition of a perfect code, we also looked at some examples along the way. With that we come to the end of this module.