CT111 Introduction to Communication Systems
Lecture 13: Channel Coding

Yash M. Vasavada

Associate Professor, DA-IICT, Gandhinagar

19th March 2018

Overview of Today's Talk

Overview of Today's Talk

Overview of Today's Talk

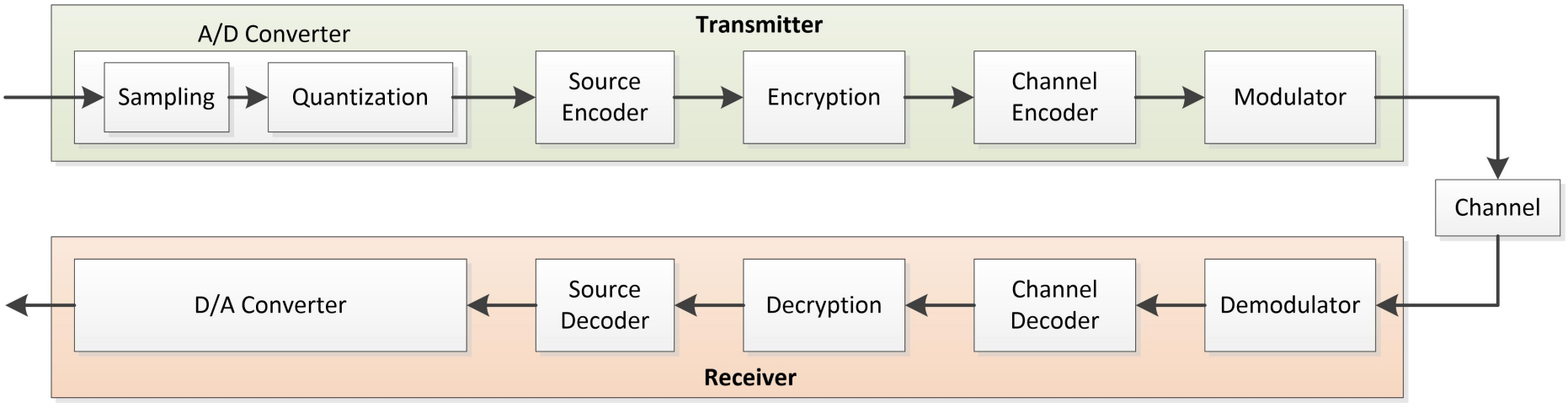
Overview of Today's Talk

Overview of Today's Talk

Overview of Today's Talk

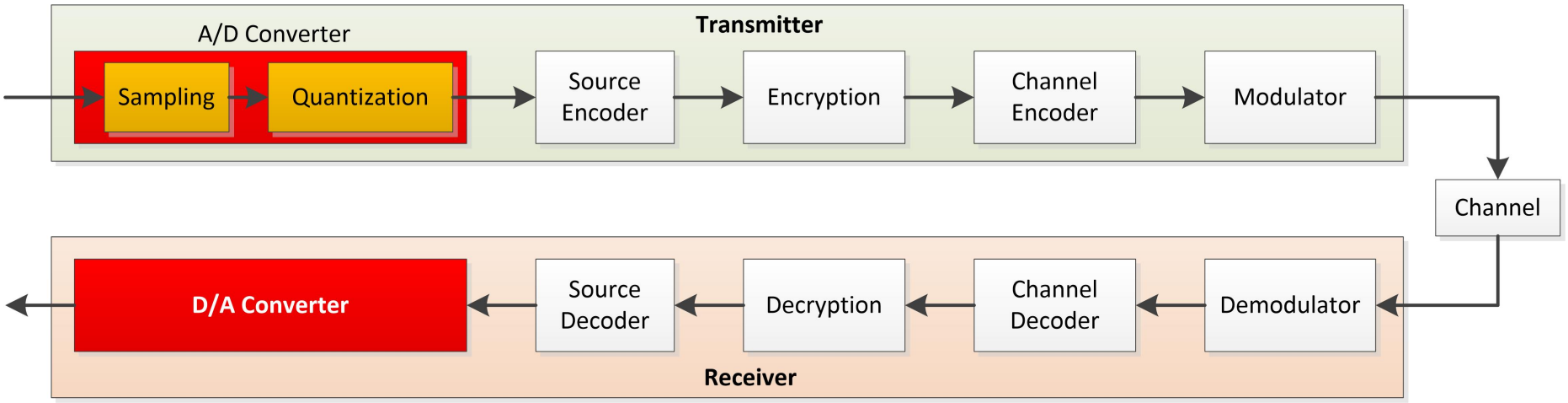
Digital Communication Transceiver
Block Diagram

- We have earlier seen this block diagram model of a digital communication transceiver



Digital Communication Transceiver
Block Diagram

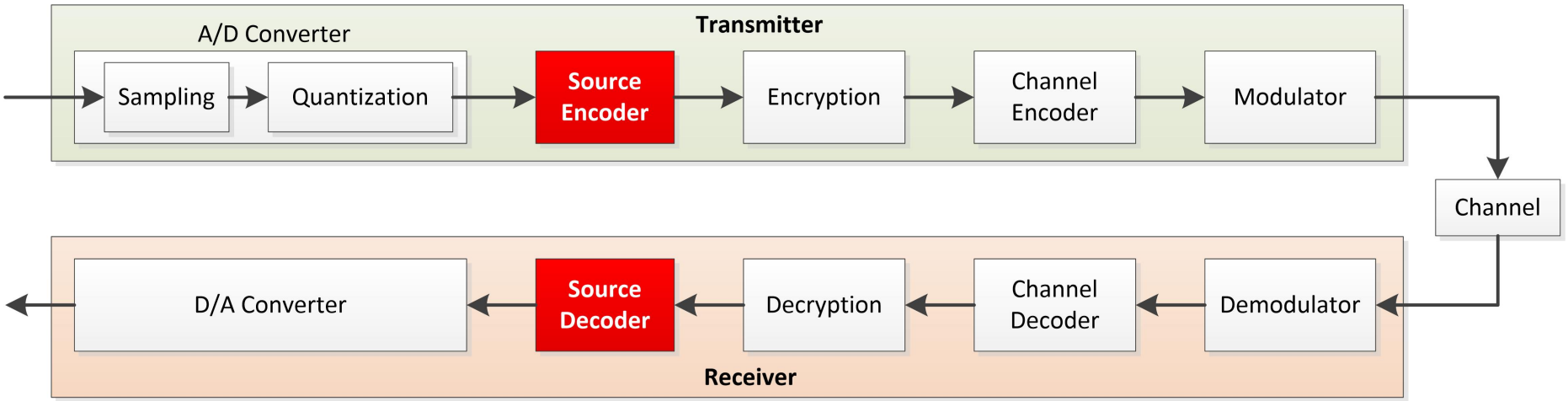
- We have studied the process of quantization of an analog information source



Digital Communication Transceiver

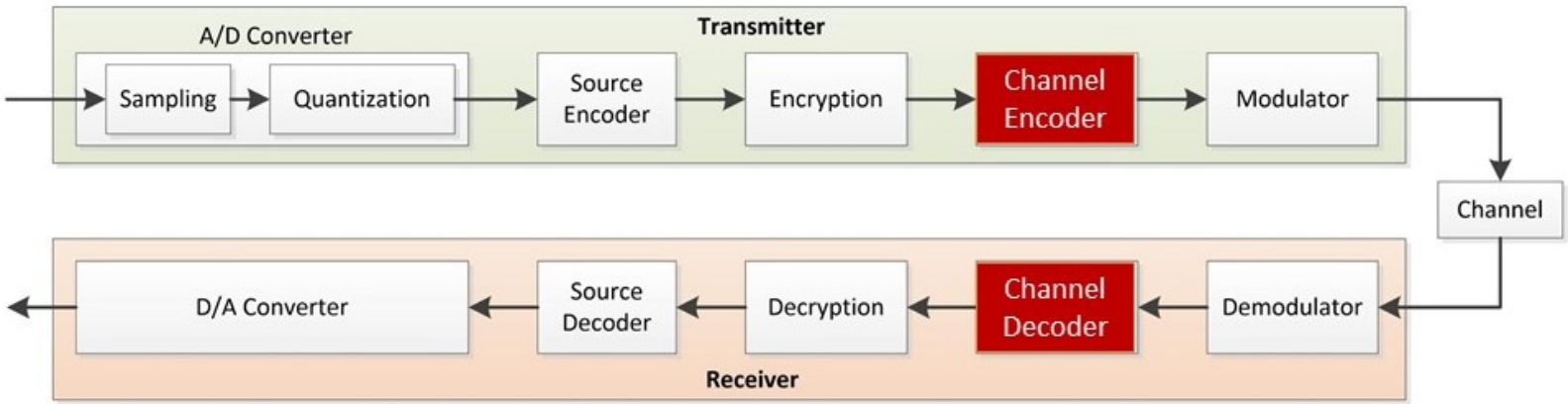
Block Diagram

- We have studied the mathematics and algorithms of source encoding



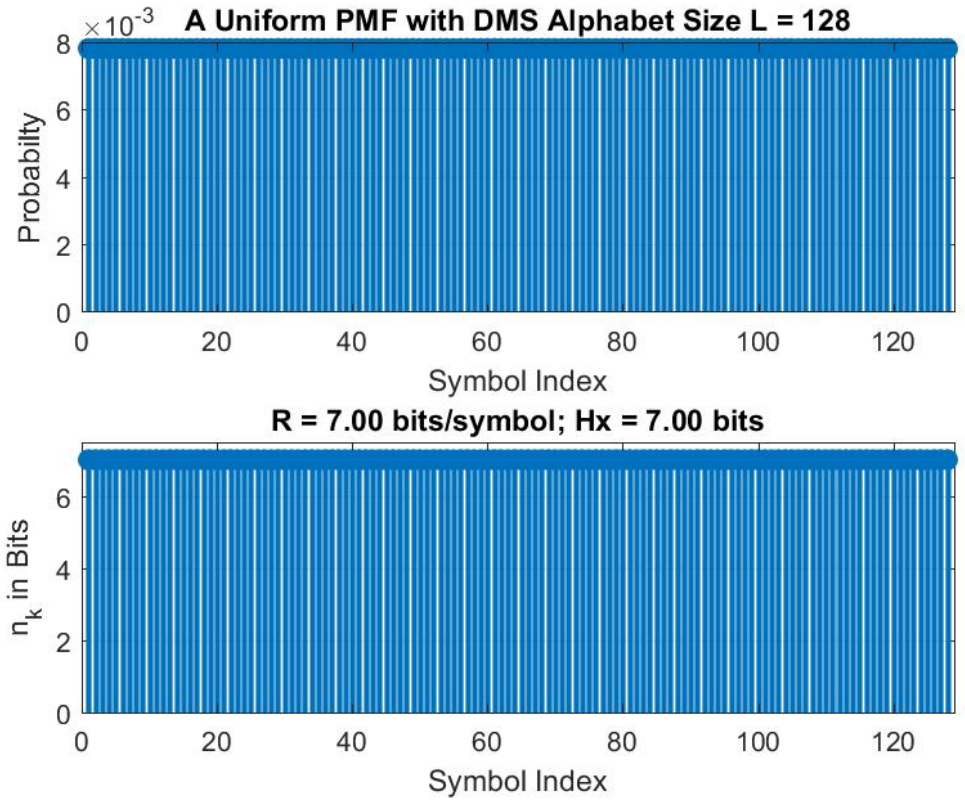
Digital Communication Transceiver
Channel Coding

- We now focus on channel encoding/decoding, also known as FEC (forward error correction) coding



A Uniform PMF

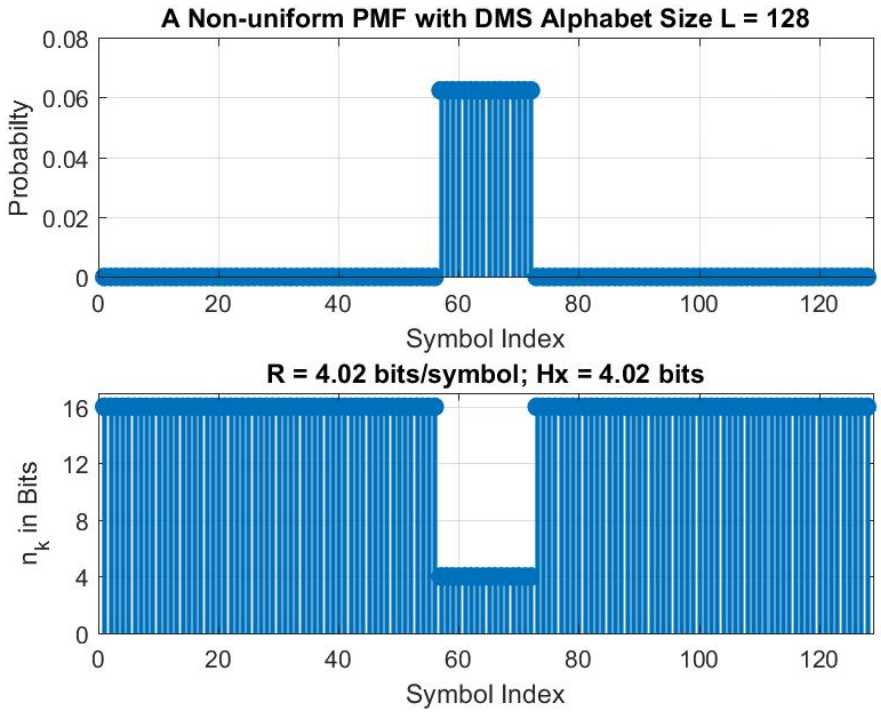
- When the DMS alphabet size is $L = 2^n$, and its PMF is uniform, the probability of each letter is $p = 1/L = 2^{-n}$, and each letter is encoded using a fixed-length code of length $-\log_2(p) = n$ bits. Shown here when $n = 7$ bits.



A Non-uniform PMF

Data Compression Possible

- Suppose $L = 128$, however, only $M = 16$ symbols actually occur realistically. Remaining 112 symbols have vanishing probability of occurrence. In this case of non-uniform PMF, one needs $n = 4$ bits instead of 7 bits (using than 4 bits will not be enough)



- An arbitrary DMS with non-uniform probabilities can be converted to a case in which most of its output sequences have vanishing probability

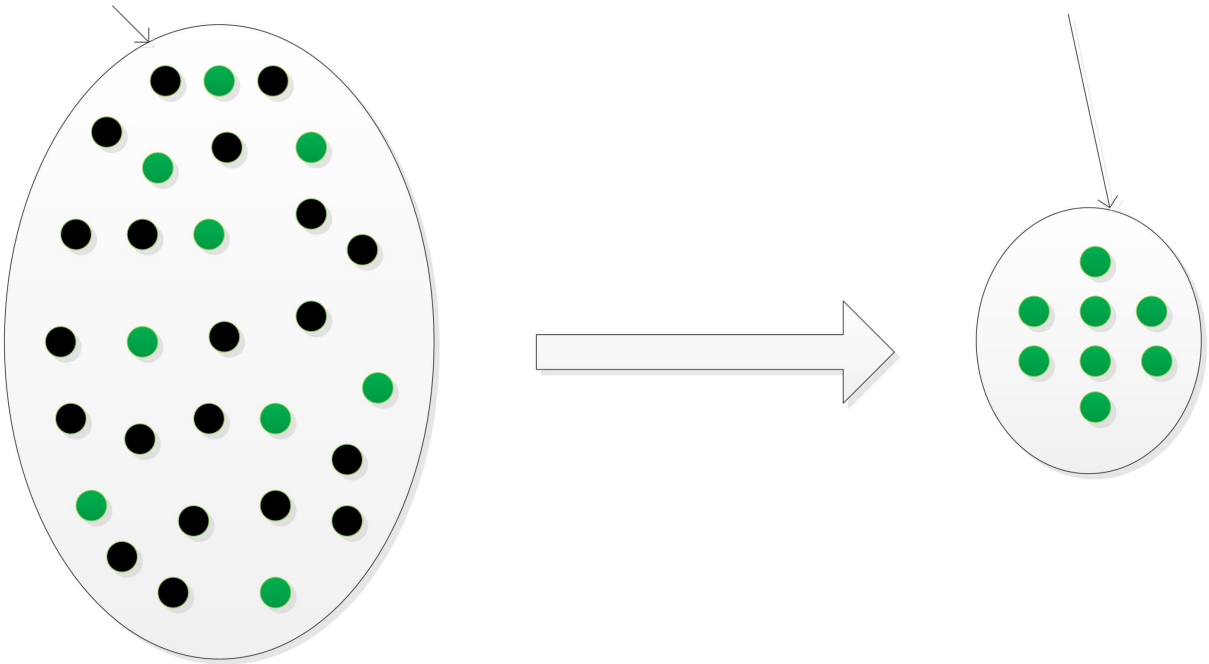
A Non-uniform PMF

Data Compression Possible

- A view of data compression

Before data compression:
Requires M bits
Total size of the set: 2^M

After data compression:
Requires M× H(X) bits
Total size of the set: $2^{M \times H(X)}$



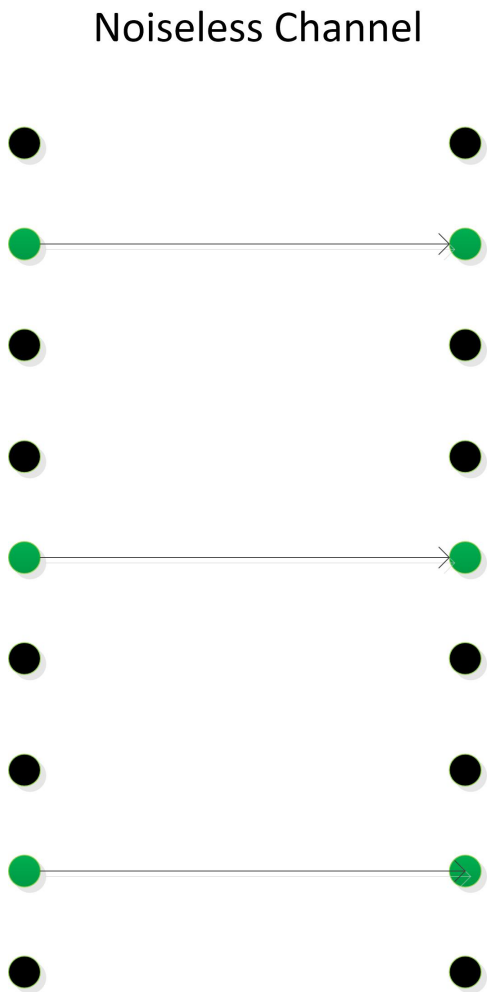
- Members of the Typical Set
- Remaining, belong to subsets with vanishing probability as N becomes large

Data Compression and Error Correction

- Data compression and error correction are dual to each other
  - Uncompressed data allows for error correction
  - Compressed data leaves no room for correcting for the errors

Data Compression and Error Correction

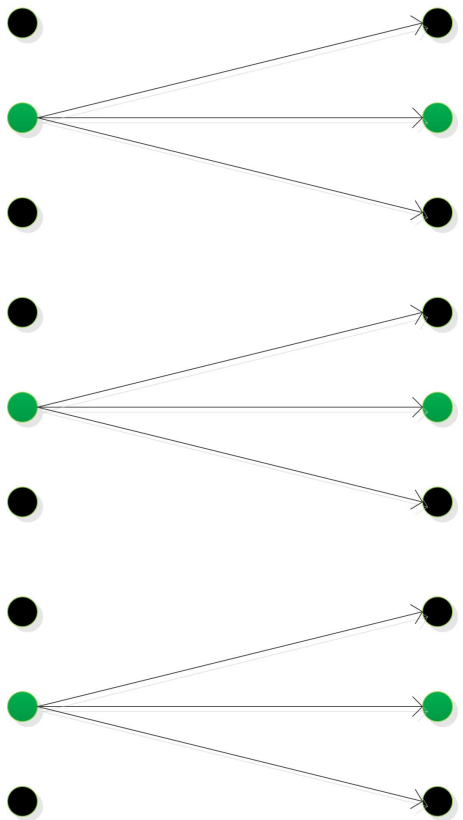
- Noiseless channel, *uncompressed* data.



Data Compression and Error Correction

- Noisy channel, *uncompressed* data.
- Effect of errors introduced by the noisy channel can be removed.

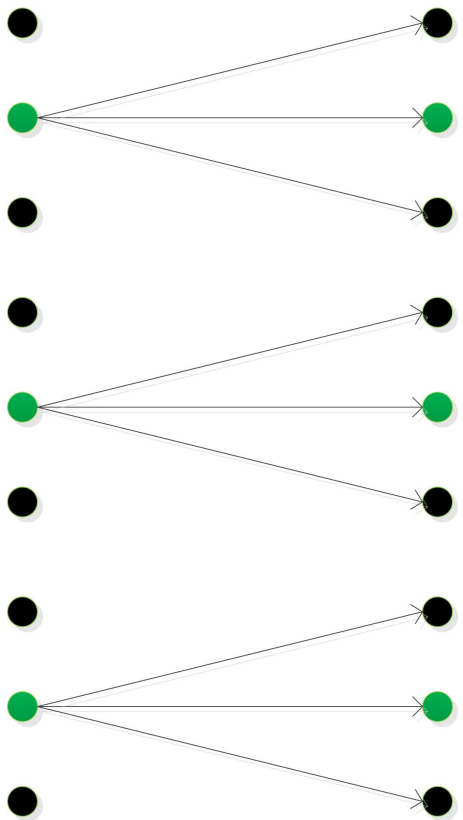
Noisy Channel



Data Compression and Error Correction

- Noisy channel, *uncompressed* data.
- Effect of errors introduced by the noisy channel can be removed.

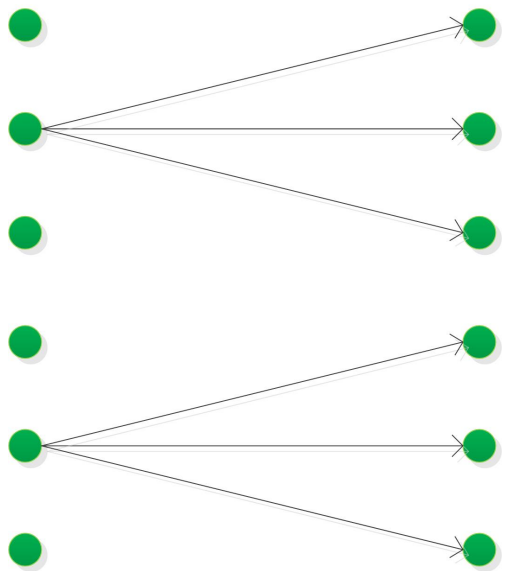
Noisy Channel



Data Compression and Error Correction

- Noisy channel, *compressed* data.
- Effect of errors introduced by the noisy channel cannot be removed.

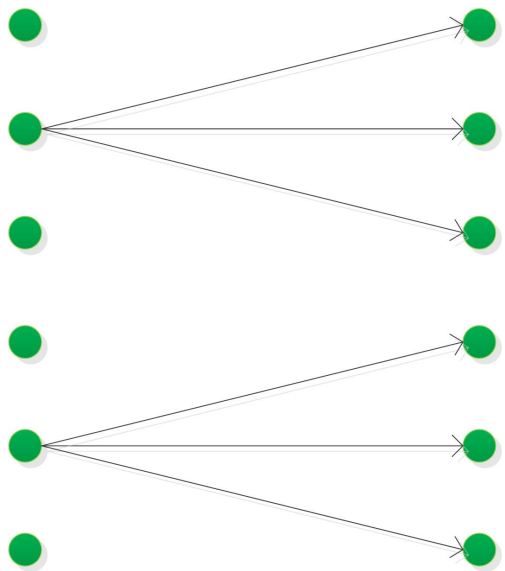
Noisy Channel



Data Compression and Error Correction

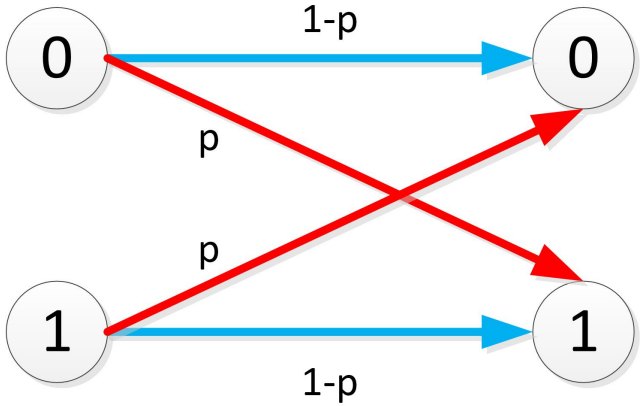
- Noisy channel, *compressed* data.
- Effect of errors introduced by the noisy channel cannot be removed.

Noisy Channel



Binary Symmetric Channel or BSC

- Consider the following simple channel for modeling the transmission of binary (zeros and ones) data



- Known as binary symmetric channel or BSC
  - Binary: this channel model does not work if the transmitted signal is nonbinary
  - Symmetric: the error affecting bit 0 has the same probability p (known as the cross-over probability of the BSC) as the error affecting bit 1; also, if an error occurs on transmitted bit 0, the receiver sees bit 1, and vice versa
  - Often parametrized by the probability p, and written as BSC(p)

Binary Symmetric Channel or BSC

- Let us say that the transmitter sends 1000 bits/sec, and p is 0. What is the maximum rate at which the information is received over this BSC($p = 0$)?
  - Obviously: 1000 bits/sec
- Let us now consider a case when p has increased and become is 0.01 (1%). What is the maximum rate at which the information is received over this BSC($p = 0.01$)?
  - One answer: 990 bits/sec (just subtract 1% of bits that are going to be received in error)
  - However, the above answer cannot be right. How can the receiver get 990 bits/sec since it does not know the positions where the errors have occurred
  - To see this, consider the worst-case when p is 0.5. In this case, although the receiver does get 1000 bits/sec over this BSC, it has no knowledge of what the transmitted bits are.
    - → A received bit of 0 could either have been due to the transmitted bit of 0 which did not get corrupted (occurs 50% of the times) or it could have been due to the transmitted bit of 1 which did get corrupted (occurs another half of the times)
    - → It would be wrong to say that the rate of information transfer is 500 bits/sec, when it is actually zero.

Binary Symmetric Channel or BSC

- Let us say that the transmitter sends 1000 bits/sec, and p is 0. What is the maximum rate at which the information is received over this BSC($p = 0$)?
  - Obviously: 1000 bits/sec
- Let us now consider a case when p has increased and become is 0.01 (1%). What is the maximum rate at which the information is received over this BSC($p = 0.01$)?
  - One answer: 990 bits/sec (just subtract 1% of bits that are going to be received in error)
  - However, the above answer cannot be right. How can the receiver get 990 bits/sec since it does not know the positions where the errors have occurred
  - To see this, consider the worst-case when p is 0.5. In this case, although the receiver does get 1000 bits/sec over this BSC, it has no knowledge of what the transmitted bits are.
    - → A received bit of 0 could either have been due to the transmitted bit of 0 which did not get corrupted (occurs 50% of the times) or it could have been due to the transmitted bit of 1 which did get corrupted (occurs another half of the times)
    - → It would be wrong to say that the rate of information transfer is 500 bits/sec, when it is actually zero.

Binary Symmetric Channel or BSC

- Let us say that the transmitter sends 1000 bits/sec, and p is 0. What is the maximum rate at which the information is received over this BSC($p = 0$)?
  - Obviously: 1000 bits/sec
- Let us now consider a case when p has increased and become is 0.01 (1%). What is the maximum rate at which the information is received over this BSC($p = 0.01$)?
  - One answer: 990 bits/sec (just subtract 1% of bits that are going to be received in error)
  - However, the above answer cannot be right. How can the receiver get 990 bits/sec since it does not know the positions where the errors have occurred
  - To see this, consider the worst-case when p is 0.5. In this case, although the receiver does get 1000 bits/sec over this BSC, it has no knowledge of what the transmitted bits are.
    - → A received bit of 0 could either have been due to the transmitted bit of 0 which did not get corrupted (occurs 50% of the times) or it could have been due to the transmitted bit of 1 which did get corrupted (occurs another half of the times)
    - → It would be wrong to say that the rate of information transfer is 500 bits/sec, when it is actually zero.

Binary Symmetric Channel or BSC

- Let us say that the transmitter sends 1000 bits/sec, and p is 0. What is the maximum rate at which the information is received over this BSC($p = 0$)?
  - Obviously: 1000 bits/sec
- Let us now consider a case when p has increased and become is 0.01 (1%). What is the maximum rate at which the information is received over this BSC($p = 0.01$)?
  - One answer: 990 bits/sec (just subtract 1% of bits that are going to be received in error)
  - However, the above answer cannot be right. How can the receiver get 990 bits/sec since it does not know the positions where the errors have occurred
  - To see this, consider the worst-case when p is 0.5. In this case, although the receiver does get 1000 bits/sec over this BSC, it has no knowledge of what the transmitted bits are.
    - → A received bit of 0 could either have been due to the transmitted bit of 0 which did not get corrupted (occurs 50% of the times) or it could have been due to the transmitted bit of 1 which did get corrupted (occurs another half of the times)
    - → It would be wrong to say that the rate of information transfer is 500 bits/sec, when it is actually zero.

Binary Symmetric Channel or BSC

- Let us say that the transmitter sends 1000 bits/sec, and p is 0. What is the maximum rate at which the information is received over this BSC($p = 0$)?
  - Obviously: 1000 bits/sec
- Let us now consider a case when p has increased and become is 0.01 (1%). What is the maximum rate at which the information is received over this BSC($p = 0.01$)?
  - One answer: 990 bits/sec (just subtract 1% of bits that are going to be received in error)
  - However, the above answer cannot be right. How can the receiver get 990 bits/sec since it does not know the positions where the errors have occurred
  - To see this, consider the worst-case when p is 0.5. In this case, although the receiver does get 1000 bits/sec over this BSC, it has no knowledge of what the transmitted bits are.
    - → A received bit of 0 could either have been due to the transmitted bit of 0 which did not get corrupted (occurs 50% of the times) or it could have been due to the transmitted bit of 1 which did get corrupted (occurs another half of the times)
    - → It would be wrong to say that the rate of information transfer is 500 bits/sec, when it is actually zero.

Binary Symmetric Channel or BSC

- Let us say that the transmitter sends 1000 bits/sec, and p is 0. What is the maximum rate at which the information is received over this BSC($p = 0$)?
  - Obviously: 1000 bits/sec
- Let us now consider a case when p has increased and become is 0.01 (1%). What is the maximum rate at which the information is received over this BSC($p = 0.01$)?
  - One answer: 990 bits/sec (just subtract 1% of bits that are going to be received in error)
  - However, the above answer cannot be right. How can the receiver get 990 bits/sec since it does not know the positions where the errors have occurred
  - To see this, consider the worst-case when p is 0.5. In this case, although the receiver does get 1000 bits/sec over this BSC, it has no knowledge of what the transmitted bits are.
    - → A received bit of 0 could either have been due to the transmitted bit of 0 which did not get corrupted (occurs 50% of the times) or it could have been due to the transmitted bit of 1 which did get corrupted (occurs another half of the times)
    - → It would be wrong to say that the rate of information transfer is 500 bits/sec, when it is actually zero.

Model of Channel Induced Bit Errors
for Binary Symmetric Channel

- To ensure a reliable information transfer over the BSC, the transmitter needs to allocate a *fraction* of transmitted bits so that the receiver is not confused about which message was transmitted
- Toward this, let us consider the BSC channel as a source of a binary string of length 1000 bits/sec in the example of the previous slide
  - Bit 1 of this BSC string specifies the location where the error has occurred
  - Bit 0 specifies the locations that are error free
- Let the transmitted binary sequence be represented as **x**
- the BSC noise string be represented as **n**,
- the received binary sequence **y** can be written in the following two ways:
  - exclusive-OR of **x** and **n**: $\mathbf{y} = \mathbf{x} \oplus \mathbf{n}$
  - sum (modulo-2) of **x** and **n**: $\mathbf{y} = (\mathbf{x} + \mathbf{n})\ mod - 2$

Model of Channel Induced Bit Errors

for Binary Symmetric Channel

- Denoting the transmitted binary sequence as **x**, and the BSC noise string as **n**, the received binary sequence **y** can be written in the following two ways:
  - exclusive-OR of **x** and **n**: $\mathbf{y} = \mathbf{x} \oplus \mathbf{n}$
  - sum (modulo-2) of **x** and **n**: $\mathbf{y} = (\mathbf{x} + \mathbf{n})\ mod - 2$
- An example:
  - $x = 0,\ 1,\ 0,\ 0,\ 1,\ 0,\ 0,\ 1$
  - $n = 0,\ 0,\ 1,\ 0,\ 1,\ 0,\ 0,\ 0$
  - $y = 0,\ 1,\ 1,\ 0,\ 0,\ 0,\ 0,\ 1$

  Notice that where **n** is 1, the corresponding bit of **x** gets flipped in **y**

Model of Channel Induced Bit Errors
for Binary Symmetric Channel

- Suppose the transmitted bit string **x** has a length of N bits
- The BSC(p) introduces errors that occur with a probability of p
  - $\rightarrow$ The BSC can be thought of as a DMS that generates length N binary string **n** in which probability of 1 equals p
- Question: how many different binary strings can the transmitted bit sequence **x** turn into because of the effect of the BSC(p)?
- Answer:
  - $\rightarrow$ For small values of N, it is not possible to specify this.
  - $\rightarrow$ However, as $N \rightarrow \infty$, the BSC(p) induced noise pattern **n** gets *trapped* into a typical set of size $2^{N \times H_b(p)}$
    - $H_b(p) \stackrel{\text{def}}{=} -(p \log_2 p + (1-p) \log_2 (1-p))$

Model of Channel Induced Bit Errors
for Binary Symmetric Channel

- Suppose the transmitted bit string **x** has a length of N bits
- The BSC(p) introduces errors that occur with a probability of p
  - $\rightarrow$ The BSC can be thought of as a DMS that generates length N binary string **n** in which probability of 1 equals p
- Question: how many different binary strings can the transmitted bit sequence **x** turn into because of the effect of the BSC(p)?
- Answer:
  - $\rightarrow$ For small values of N, it is not possible to specify this.
  - $\rightarrow$ However, as $N \rightarrow \infty$, the BSC(p) induced noise pattern **n** gets *trapped* into a typical set of size $2^{N \times H_b(p)}$
    - $H_b(p) \stackrel{\text{def}}{=} -(p \log_2 p + (1-p) \log_2 (1-p))$

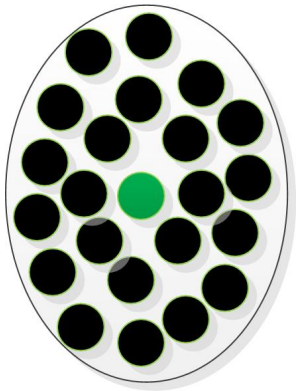
Model of Channel Induced Bit Errors
for Binary Symmetric Channel

- Suppose the transmitted bit string **x** has a length of N bits
- The BSC(p) introduces errors that occur with a probability of p
  - $\rightarrow$ The BSC can be thought of as a DMS that generates length N binary string **n** in which probability of 1 equals p
- Question: how many different binary strings can the transmitted bit sequence **x** turn into because of the effect of the BSC(p)?
- Answer:
  - $\rightarrow$ For small values of N, it is not possible to specify this.
  - $\rightarrow$ However, as $N \rightarrow \infty$, the BSC(p) induced noise pattern **n** gets *trapped* into a typical set of size $2^{N \times H_b(p)}$
    - $\triangleright$ $H_b(p) \stackrel{\text{def}}{=} -(p \log_2 p + (1-p) \log_2 (1-p))$

Model of Channel Induced Bit Errors

for Binary Symmetric Channel

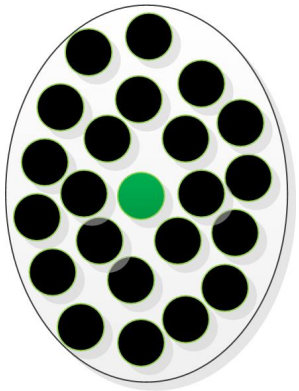
- A single transmitted binary string of $N \times H_b(p)$ bits can turn into $2^{N \times H_b(p)}$ other strings around the transmitted string



Model of Channel Induced Bit Errors

for Binary Symmetric Channel

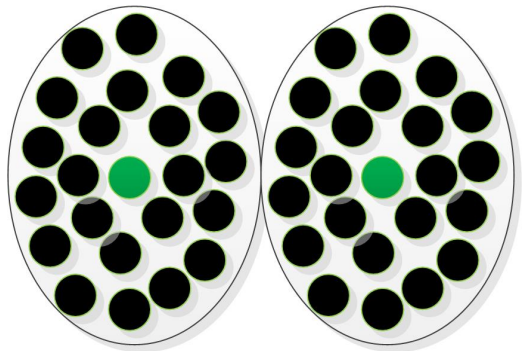
- A single transmitted binary string of $N \times H_b(p)$ bits can turn into $2^{N \times H_b(p)}$ other strings around the transmitted string



Model of Channel Induced Bit Errors

for Binary Symmetric Channel

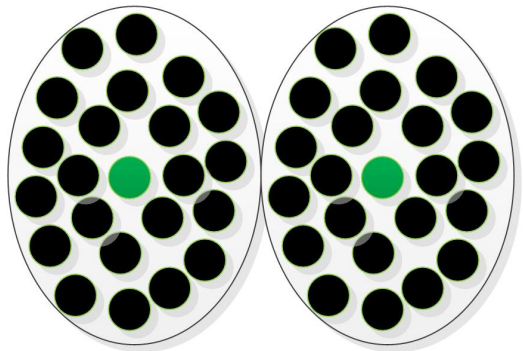
- Two different transmitted binary strings of $N \times H_b(p) + 1$ bits can turn into $2 \times 2^{N \times H_b(p)}$ other strings around the transmitted string.
- We can send one bit of information and it is ensured to not get corrupted.
- Total number of bits sent: $N \times H_b(p) + 1$; number of bits that carry information: 1; redundant bits required to isolate the transmitted message from the effect of BSC(p): $N \times H_b(p)$



Model of Channel Induced Bit Errors

for Binary Symmetric Channel

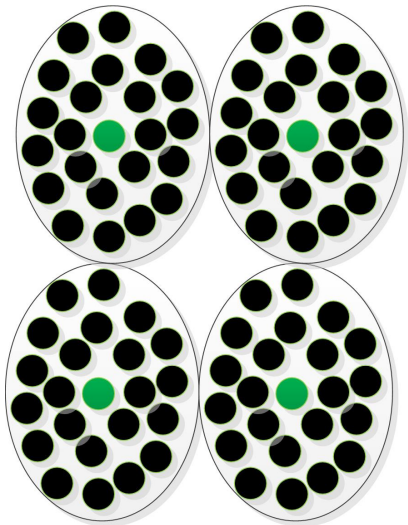
- Two different transmitted binary strings of $N \times H_b(p) + 1$ bits can turn into $2 \times 2^{N \times H_b(p)}$ other strings around the transmitted string.
- We can send one bit of information and it is ensured to not get corrupted.
- Total number of bits sent: $N \times H_b(p) + 1$; number of bits that carry information: 1; redundant bits required to isolate the transmitted message from the effect of BSC(p): $N \times H_b(p)$



Model of Channel Induced Bit Errors

for Binary Symmetric Channel

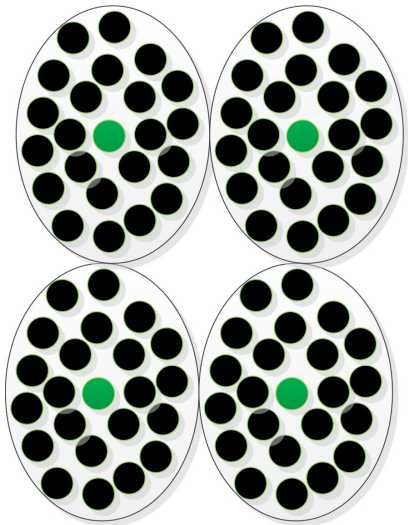
- Four different transmitted binary strings of $N \times H_b(p) + 2$ bits can turn into $4 \times 2^{N \times H_b(p)}$ other strings around the transmitted string.
- We can send two bits of information and these two bits are ensured to not get corrupted.
- Total number of bits sent: $N \times H_b(p) + 2$; number of bits that carry information: 2; redundant bits required to isolate the transmitted message from the effect of BSC(p): $N \times H_b(p)$



Model of Channel Induced Bit Errors

for Binary Symmetric Channel

- Four different transmitted binary strings of $N \times H_b(p) + 2$ bits can turn into $4 \times 2^{N \times H_b(p)}$ other strings around the transmitted string.
- We can send two bits of information and these two bits are ensured to not get corrupted.
- Total number of bits sent: $N \times H_b(p) + 2$; number of bits that carry information: 2; redundant bits required to isolate the transmitted message from the effect of BSC(p): $N \times H_b(p)$



Model of Channel Induced Bit Errors

for Binary Symmetric Channel

- Generalizing, we transmitting N bits over BSC(p), we can send information over $N \times (1 - H_b(p))$ bits and these information bits are ensured to not get corrupted.
- Total number of bits sent: N; number of bits that carry information: $N \times (1 - H_b(p))$; redundant bits required to isolate the transmitted message from the effect of BSC(p): $N \times H_b(p)$

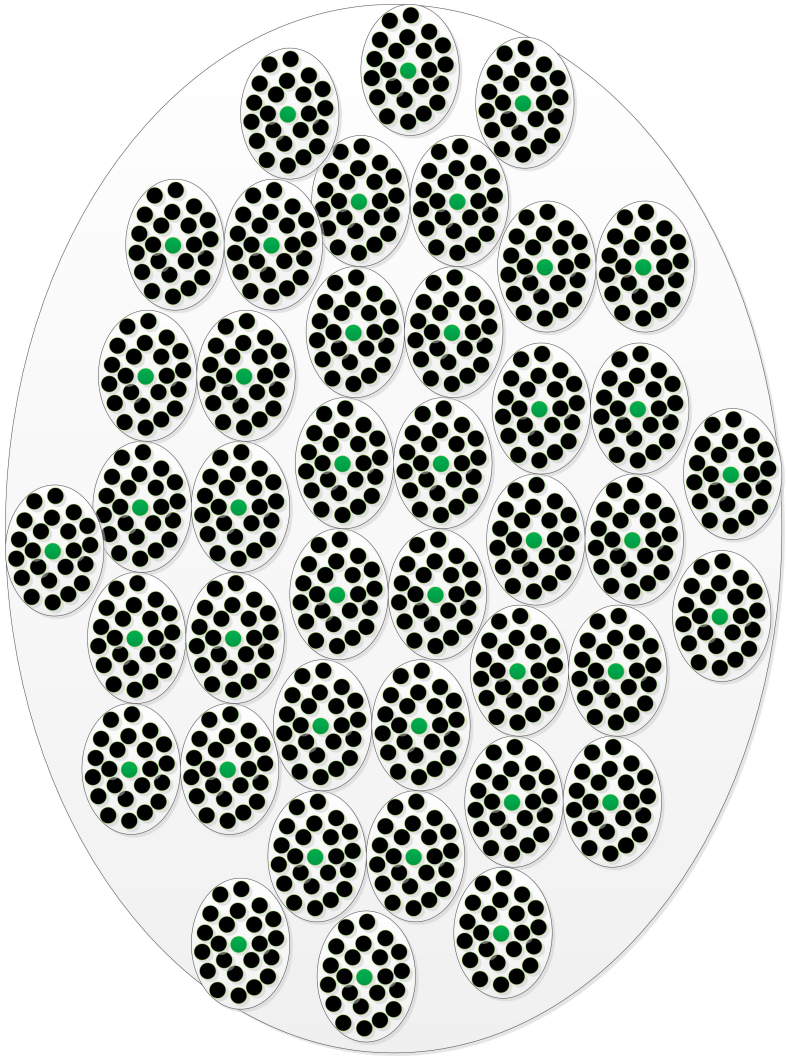
Model of Channel Induced Bit Errors

for Binary Symmetric Channel

- Generalizing, we transmitting N bits over BSC(p), we can send information over $N \times (1 - H_b(p))$ bits and these information bits are ensured to not get corrupted.
- Total number of bits sent: N; number of bits that carry information: $N \times (1 - H_b(p))$; redundant bits required to isolate the transmitted message from the effect of BSC(p): $N \times H_b(p)$

Model of Channel Induced Bit Errors

for Binary Symmetric Channel



Summary
for Binary Symmetric Channel

- When transmitting a total of N bits, we can divide the total set as $2^N = 2^{N-N\times H_b(p)} \times 2^{N\times H_b(p)}$
- Thus, we form a total of $2^{N-N\times H_b(p)}$ non-overlapping subsets of the full set of size 2^N. Size of each subset is $2^{N\times H_b(p)}$.
- Transmitted channel codewords are the centers of these subsets
- BSC(p) can turn the transmitted codeword into any one string in the corresponding subset. As $N \to \infty$, the received codeword is *ensured* to be trapped within this subset.
- Receiver calculates which subset the received string falls into, and then, it takes the transmitted codeword as the center of that subset
- Since there are $2^{N-N\times H_b(p)}$ total mutually nonoverlapping subsets, the transmitter can set a total of $N - N \times H_b(p)$ bits that carry information. It has to insert the remaining $N \times H_b(p)$ bits as redundant bits that are required to ensure the unique decodeability

Summary (Alternate Version)

for Binary Symmetric Channel

- Transmitter sends a "codeword" which is N bits long.
  - → This is the channel encoded codeword.
  - → Reason it is called a "codeword" is because it is different from the actual information or message bits that come out of the source encoder
- There are total of 2^N possible codewords that are N bit long
- Suppose the transmitter knows that its codeword has to be transmitted on a BSC(p); i.e., the transmitter knows the value of p.
  - → In reality, the transmitter may not know the value of p, but the engineering design is done for the worst possible value of p that is expected. This worst-case value of p may be possible to figure out, or at least guess
- In this case, instead of using all 2^N codewords, the transmitter pre-selects $2^{N\times(1-H_b(p))}$ codewords that it will transmit
  - → This forms the (channel) codeword book (or a dictionary of all the codewords)
  - → This codeword book is kept at the transmitter (channel encoder) as well as at the receiver (channel decoder)
  - → Just like a dictionary, this has a unique (one-to-one) map. Each one of $2^{N\times(1-H_b(p))}$ message block is mapped to a unique N-bit long codeword

Summary (Alternate Version, Continued)

for Binary Symmetric Channel

- Any message that transmitter sends is an N-bit codeword
- This message is, however, one codeword out of a codeword dictionary that has a total of $2^{N \times (1-H_b(p))}$ codewords
- Therefore, the transmitter actually needs (not N, but instead) $N \times (1 - H_b(p))$ bits to specify (or select) which codeword is to be sent
- Accordingly, the transmitter takes a block of $N \times (1 - H_b(p))$ message or information bits coming out of the source encoder to select one of $2^{N \times (1-H_b(p))}$ codewords from the dictionary
  - $\rightarrow$ The selected N-bit long codeword is one of $2^{N \times (1-H_b(p))}$ green circles on slide 28
- The BSC(p) can turn this selected codeword into a total of $2^{NH_b(p)}$ possible N-bit long binary strings that make up the little oval around the selected (green colored) codeword
  - $\rightarrow$ All the black points in this oval denote all the possible ways that the BSC can turn the transmitted codeword into the received codeword
  - $\rightarrow$ Effect of the BSC is guaranteed to be confined within this little oval

Summary (Alternate Version, Continued)

for Binary Symmetric Channel

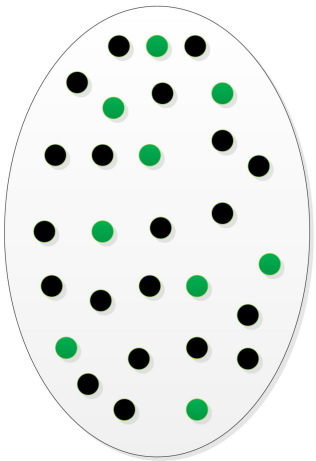
- When the receiver gets a noisy codeword over BSC, it calculates the "distance" of this received noisy codeword against all the valid, non-noisy, codewords it has in its codebook dictionary
- Receiver selects that codeword as the one that must have been transmitted for which the distance to the received noisy codeword is the smallest
- With this "algorithm", the receiver is guaranteed to select the same codeword that the transmitter has sent. It has overcome the effect of the noise.
- With this scheme, the channel encoding completely protects the transmitted information bits of length $N \times (1 - H_b(p)$.
- The cost paid is that the transmitter needed to send N bits instead of $N \times (1 - H_b(p)$ bits, the extra $NH_b(p)$ bits are required as the redundancy bits that overcome the channel induced noise errors.

Source Coding versus Channel Coding

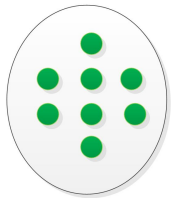
Uncompressed Source Output

After Compression

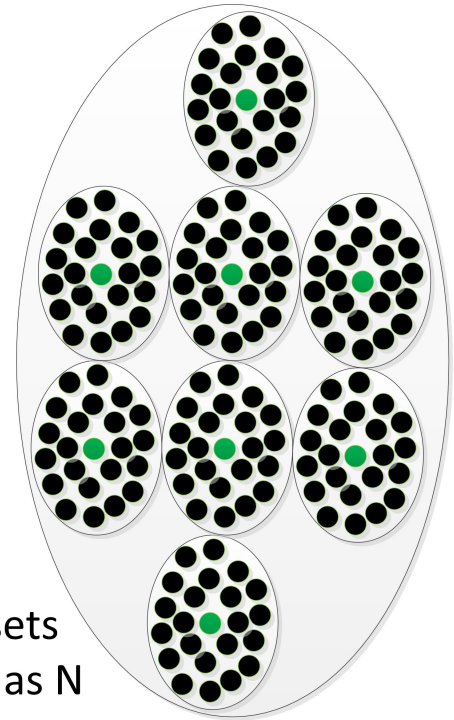
After Channel Encoding: Transmitter selects one of green cicrles; Receiver receives any one of green and black circles



Source Coding



Channel Coding

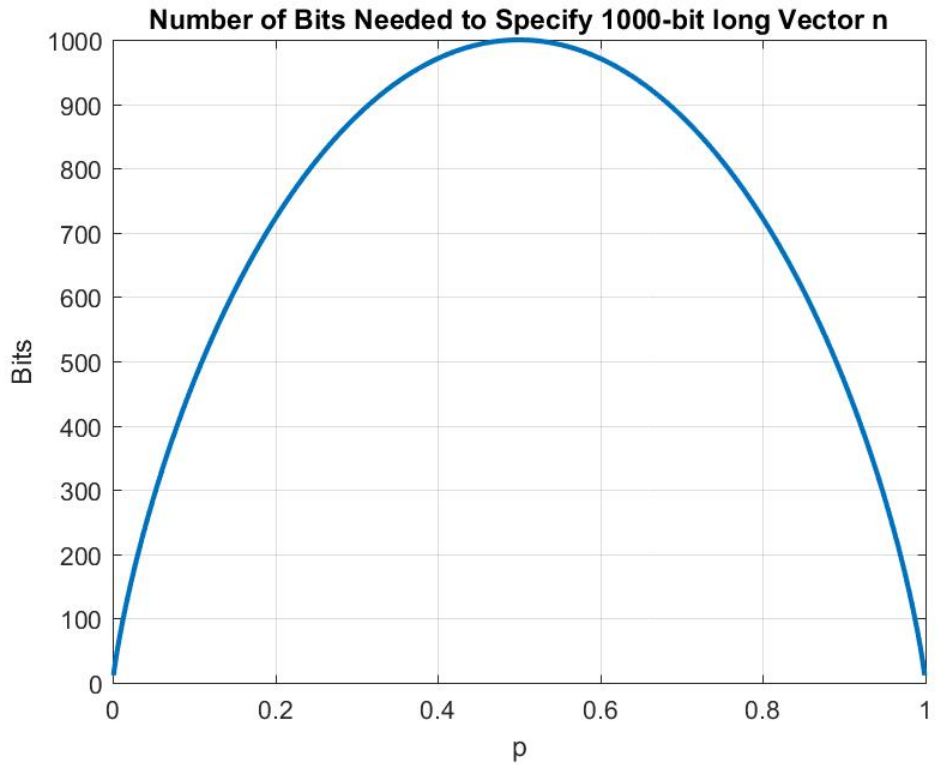


- Members of the Typical Set
- Remaining, belong to subsets with vanishing probability as N becomes large

Binary Symmetric Channel or BSC

Redundancy Bits

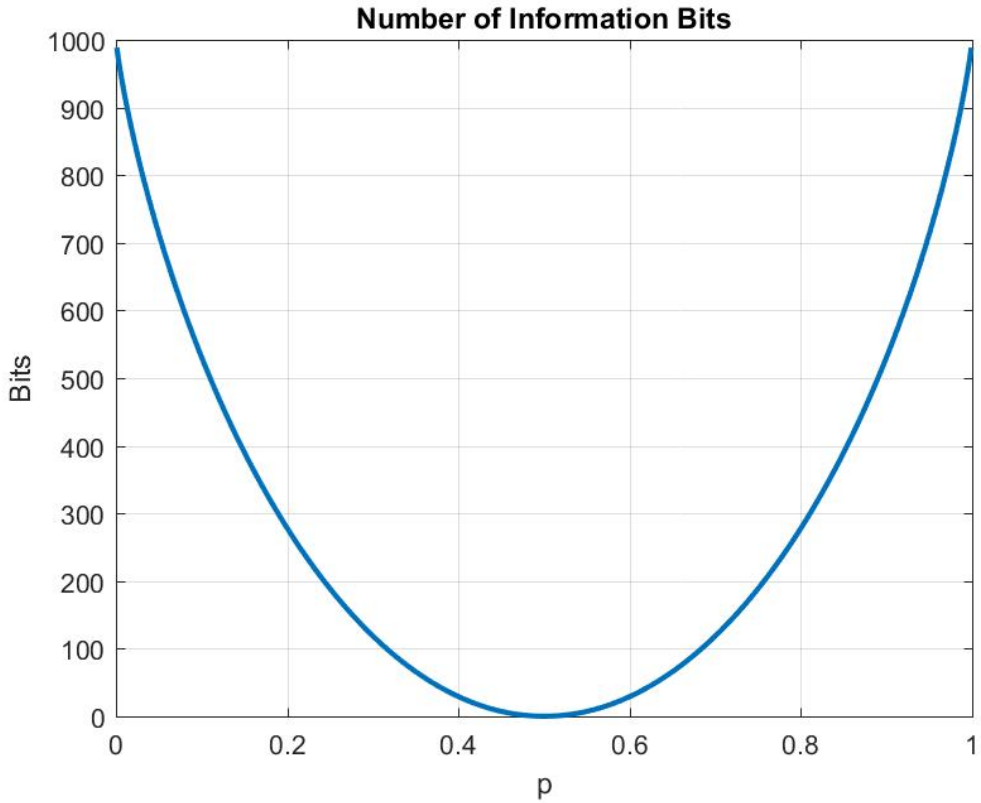
- Smallest number of bits required to isolate the effect of adding the binary string **n** of length N bits: $N \times H_b(p)$.
- Transmitter has to insert these bits, but they don't carry information (they ensure that the receiver is not confused about which message was transmitted although the individual bits can be in error)



Binary Symmetric Channel or BSC

Information Bits

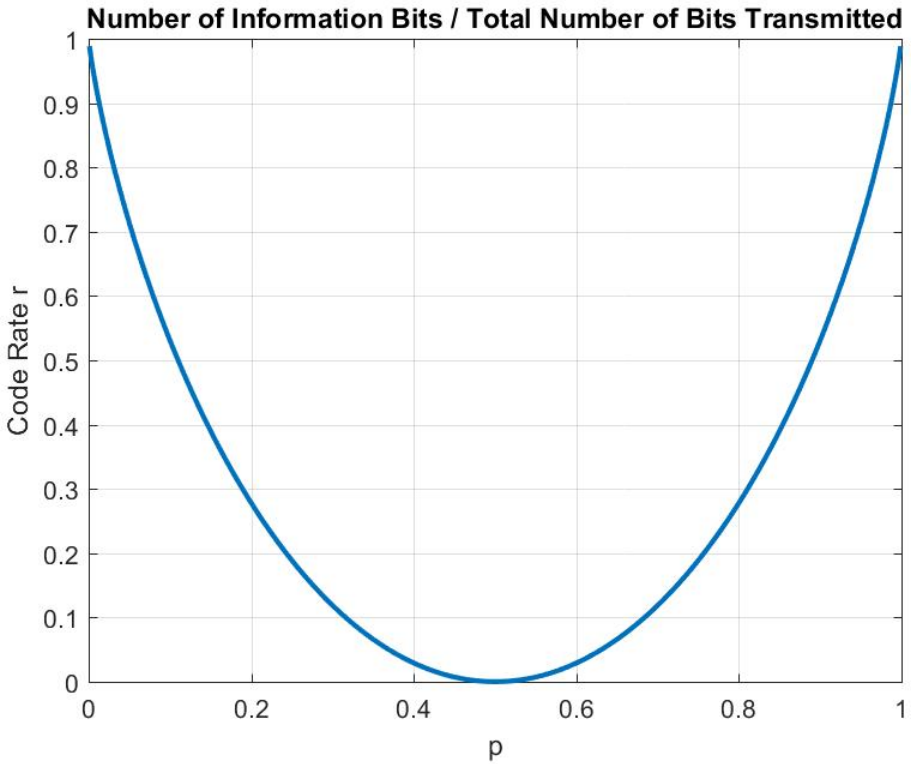
- Maximum number of bits of **x** that can carry the actual sender's information: $N - N \times H_b(p)$.
- This is the actual information transferred in bits. As expected, this becomes zero when $p = 0.5$.



Binary Symmetric Channel or BSC

Normalized Rate

- The normalized (maximum) rate of information transfer over BSC with cross-over probability of p: $r = 1 - H_b(p)$.
- Closer the value of BSC's p to 0.5, smaller the value of r, the greater the required redundancy and smaller the *capacity* of the channel to carry the actual information



Binary Symmetric Channel or BSC

Summary

- Maximum possible rate of information transfer with error-free reception:

$$C = 1 - H_b(p) bits/bit$$

- Here, $H_b(p) = -p \log_2 p - (1-p) \log_2 (1-p)$ bits is the binary entropy function.
- For the BSC(p), the transmitter is required to allocate $H_b(p)$ bits just to overcome the effect of the noise.
- It can put the information in only remaining $1 - H_b(p)$ bits

Binary Symmetric Channel or BSC

Summary

- When $p = 0.01$, $1 - H_b(p) \approx 0.92$.
  - $\rightarrow$ The transmitter can send the information in 920 bits out of 1000 bits sent every second. The maximum rate of information transfer is 920 bits/sec.
- When $p = 0.1$, $1 - H_b(p) = 0.53$.
  - $\rightarrow$ The transmitter can send the information in 530 bits out of 1000 bits sent every second. The maximum rate of information transfer is 530 bits/sec.

Example Channel Coding Techniques

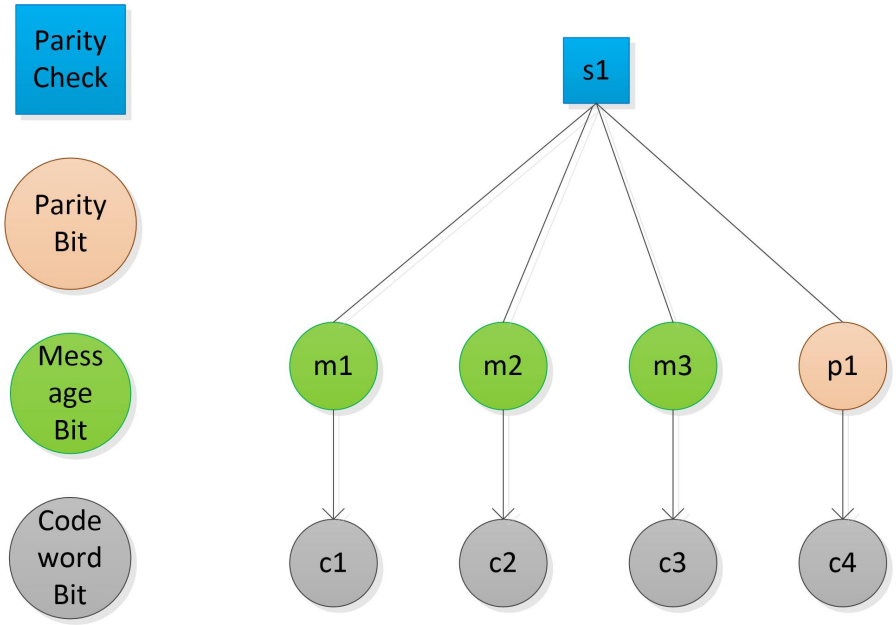
Single Parity Check (SPC) Code

- Encoding scheme:
  1. Take a block of $n-1 \geq 1$ bits as input.
  2. Add an extra n^{th} bit, called the parity check bit, which is 0 if the input block of n bits has even number of 1's, and it is 1 otherwise
  3. Transmit the resultant n bit long codeword
- Let $\{m_1, m_2, \ldots, m_{n-1}\}$ be the input block. The parity bit added is given as $p_n = \left(\sum_{k=1}^{n-1} m_k\right)_{mod-2}$
- Rate of this code is $r = \frac{n-1}{n}$.
  - $\rightarrow$ Rate r is the ratio of the number of information bits to total number of encoded bits

Example Channel Coding Techniques

Single Parity Check (SPC) Code

- Notations:
  - $\{m_k\}$: info bits,
  - $\{p_k\}$: parity bits,
  - $\{s_k\}$: check bits,
  - $\{c_k\}$: codeword bits
  - If $\mathbf{c} \stackrel{\text{def}}{=} [c_1, c_2, \ldots, c_n]^T$ is an SPC codeword, $s_1 = \sum_{i=1}^{n} c_i$ has to be zero, where this sum is modulo two.



Example Channel Coding Techniques

Single Parity Check (SPC) Code

- Decoding scheme:
  1. Take a block of n bits as input.
  2. Compute the modulot-two sum of these bits
  3. If this sum is zero, determine that zero or an even number of bit errors have occurred. If nonzero, determine that one or an odd number of bit errors have occurred
- Rate $(n-1)/n$ SPC code can detect one bit of error, but it cannot correct it

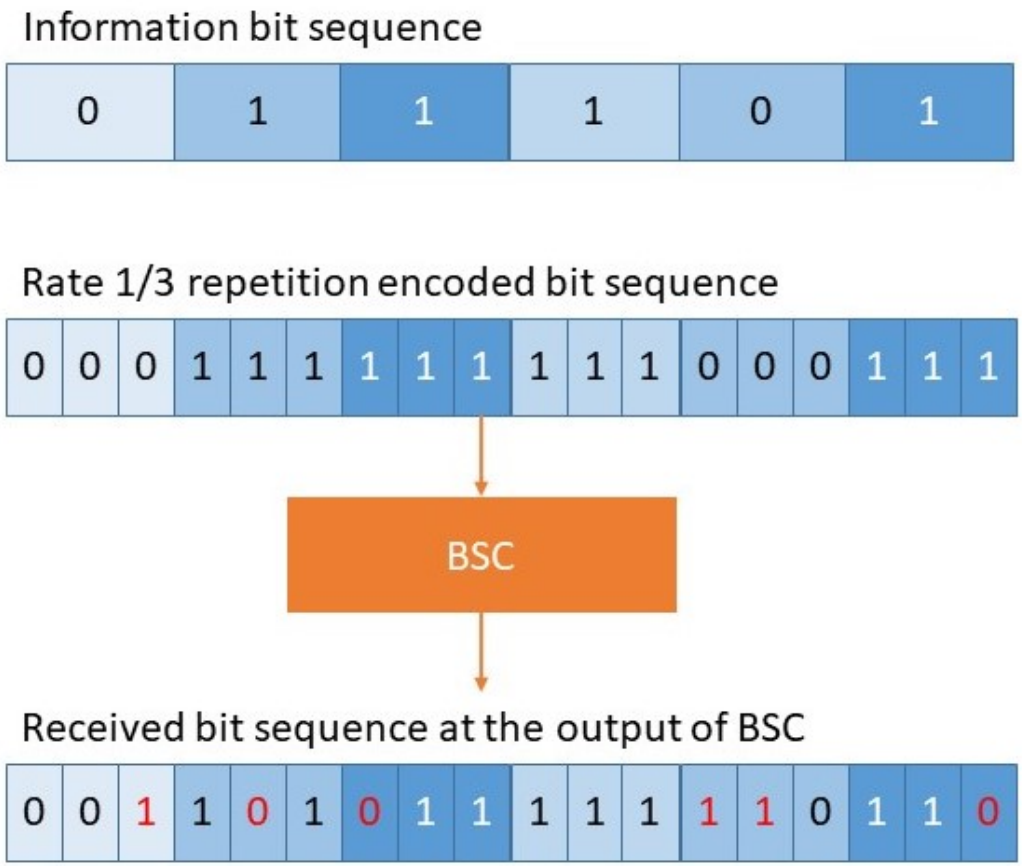
Example Channel Coding Techniques

Repetition Code

- Encoding scheme:
  1. Take one bit at a time as the input.
  2. Repeat this bit $n-1$ times
  3. Transmit the resultant n bit long codeword
- Rate of this code is $r = \frac{1}{n}$.
  - $\rightarrow$ Rate r is the ratio of the number of information bits to total number of encoded bits

Example Channel Coding Techniques

Repetition Code



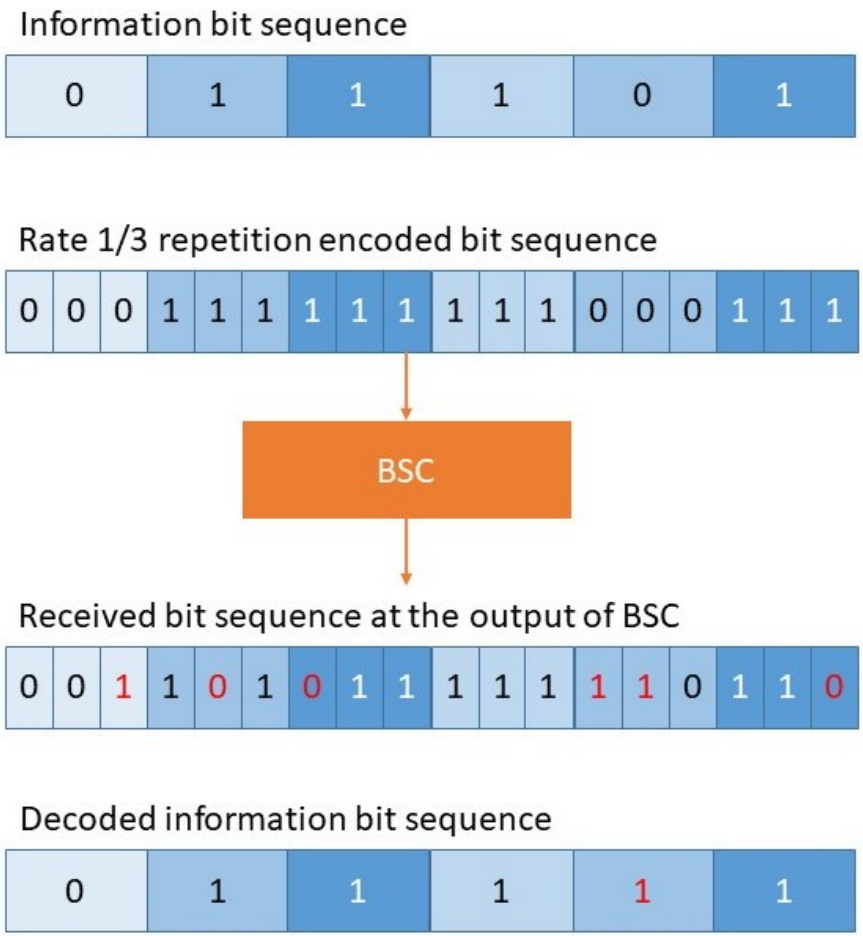
Example Channel Coding Techniques

Repetition Code

What would be a good decoding strategy for this code?

Example Channel Coding Techniques

Decoding of Repetition Code



Example Channel Coding Techniques

Decoding of Repetition Code

- The decoding algorithm is called the majority-vote decoding
  - Take $n = 3$ bit block at a time, and decode it as that bit (either 0 or 1) that occurs the majority of times
  - To avoid the confusion in decoding, it maybe preferred to make n an odd number

Example Channel Coding Techniques

Probability of Decoding Error

- Decoding error will occur in rate $r = 1/3$ repetition code if 2 or 3 bits are in error
- What is the probability of that occuring? The answer is given by the Binomial PMF

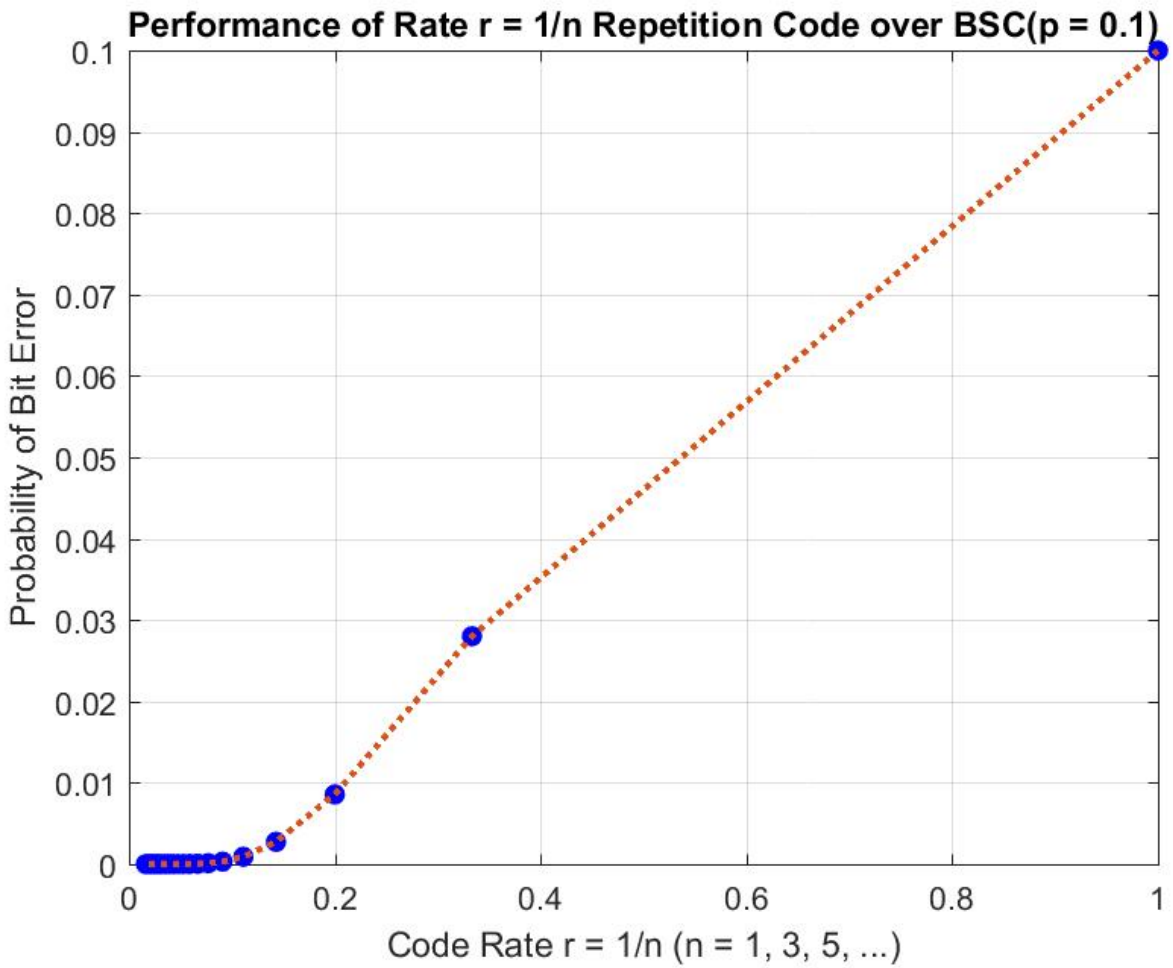
$$p_{error} = \binom{3}{2} p^2(1-p) + \binom{3}{1} p^3 = 3p^2 + p^3$$

$\rightarrow$ If $p = 0.1$, $p_{error} = 3 \times 0.1^2 \times 0.9 + 0.1^3 \approx 0.03$

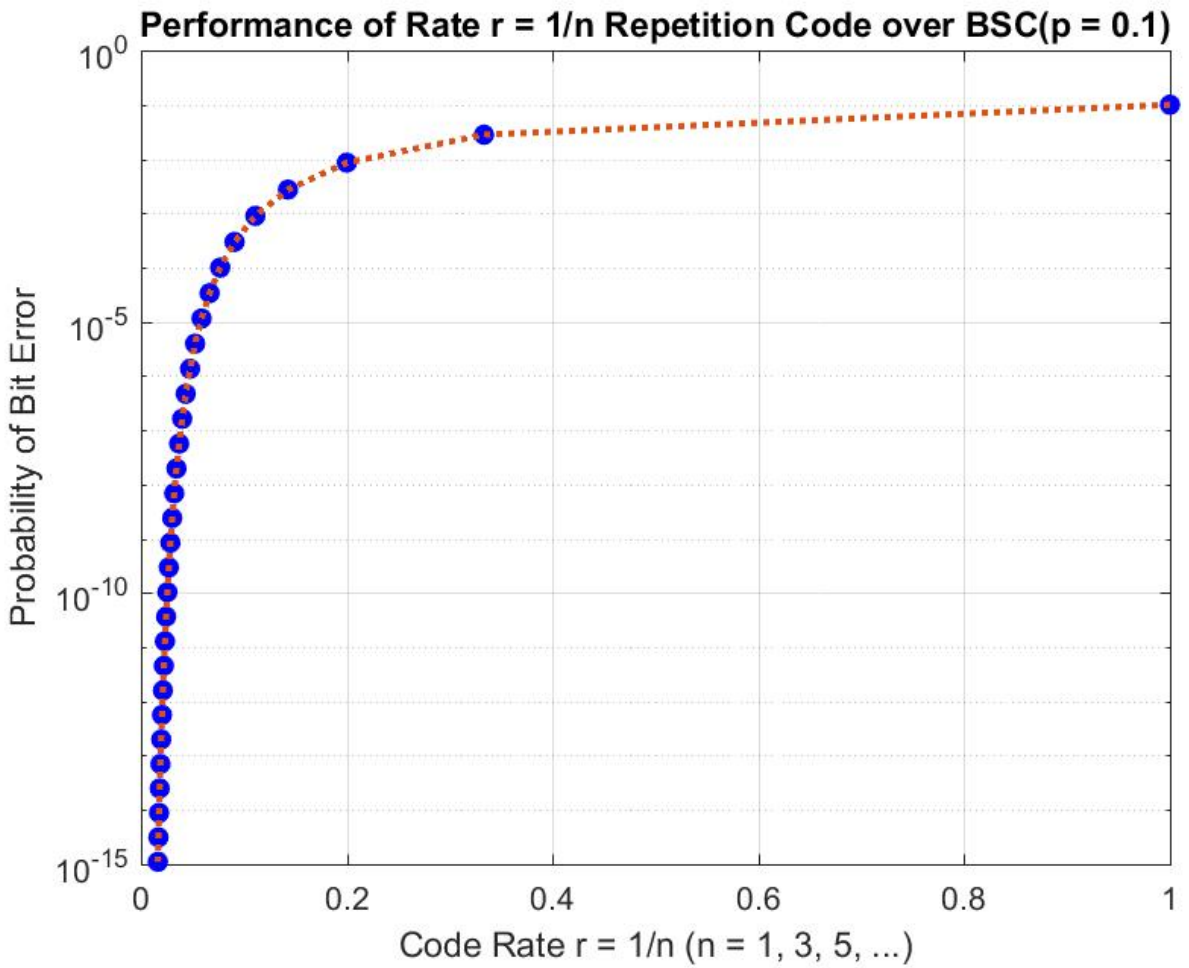
- Generalization to rate $1/n$ repetition code:

$$p_{error} = \sum_{k=(n+1)/2}^{n} \binom{n}{k} p^k (1-p)^{n-k}$$

Performance of Repetition Code
over BSC(p)



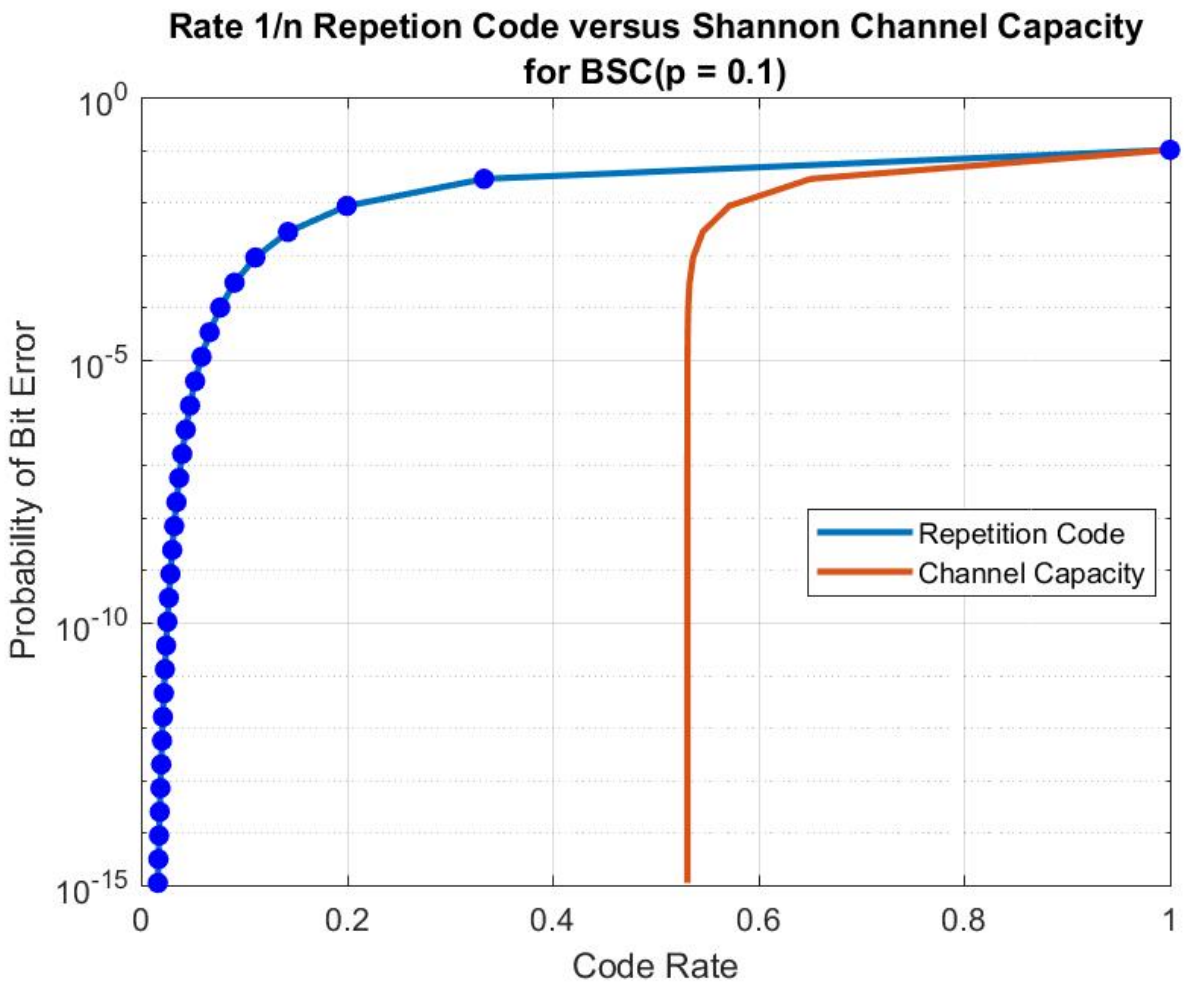
Performance of Repetition Code
over BSC(p)



Performance of Repetition Code
over BSC(p)

- If $p = 0.1$, and we would like to design a repetition code that reduces this to $p_{error} = 10^{-15}$, we need $n = 60$
  - $\rightarrow$ Repeat each bit sixty times!
- Error probability p_{error} can be arbitrarily reduced, but the price paid is huge. The rate r of information transfer is reduced significantly.
- *Comparision with the channel capacity.* we have seen earlier (on slide 33) that the channel capacity for this code is 0.53 bits/bit.

Performance of Repetition Code
over BSC(p)



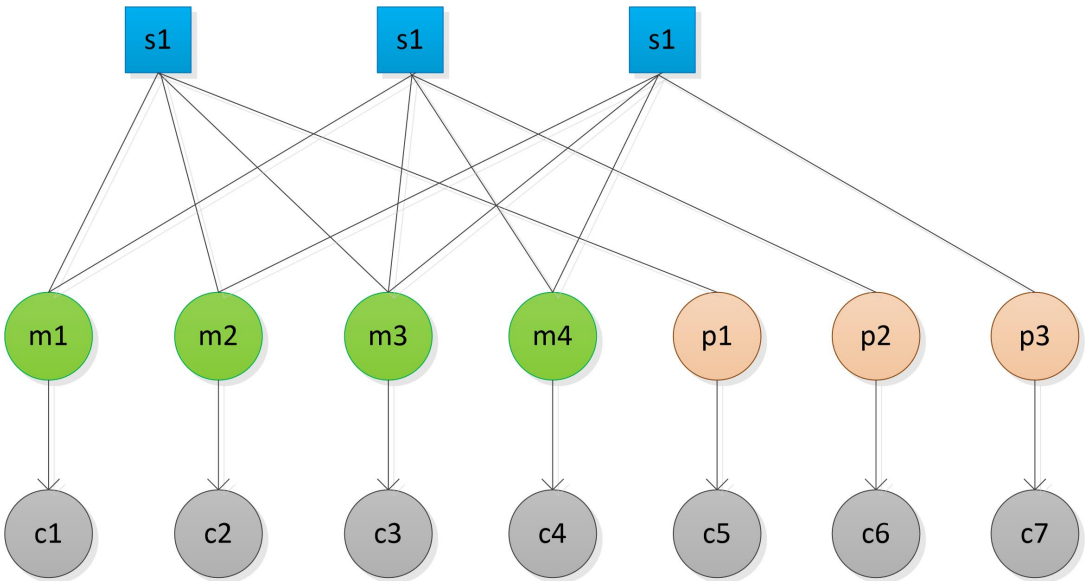
Performance of Repetition Code
over BSC(p)

- Shannon's promise: over BSC($p = 0.1$), you can transmit the information with *arbitrarily* low p_{error} with rate r as high as 0.53!
  - $\rightarrow$ Rate r does not have to be reduced to near zero
  - $\rightarrow$ In fact, p_{error} can be reduced to 10^{-15}, or 10^{-100} or 10^{-1000} at the same rate of 0.53
- Before Shannon's 1948 paper, the general belief was that attempting to reduce $p_{error} \rightarrow 0$ requires $r \rightarrow 0$.
- Shannon showed that nonzero, positive-valued, r is possible even if p_{error} is required to approach 0

Example Channel Coding Techniques

Hamming Code

- Hamming $(n = 7, k = 4)$ code is essentially three SPC codes:



$$s_1 = m_1 + m_2 + m_3 + p_1$$
$$s_1 = m_1 + m_3 + m_4 + p_2$$
$$s_1 = m_2 + m_3 + m_4 + p_3$$

Example Channel Coding Techniques

Summary of Different Codes

- Repetition Codes:
  - $k = 1; n; r = \frac{1}{n}; d_{min} = n; t_c = \left\lceil \frac{n-1}{2} \right\rceil$
- Hamming Codes:
  - $k = 2^j - 1 - j; n = 2^j - 1; r = \frac{2^j - 1 - j}{2^j - 1}; d_{min} = 3; t_c = 1$
- Golay Codes:
  - $k = 12; n = 23; r = \frac{12}{23}; d_{min} = 7; t_c = 3$
  - Only other perfect code besides the Hamming Code

Example Channel Coding Techniques

Summary of Different Codes

- Bose-Chaudhari-Hocquenghem (BCH) Codes:
  - $k; n = 2^j - 1; t_c \geq \dfrac{2^j - 1 - k}{j}$
- Reed Solomon Codes:
  - $k; n = 2^j - 1; d_{min} = n - k + 1; t_c = \left\lceil \dfrac{n-k}{2} \right\rceil$
  - Maximum Distance Separable (MDS) codes (largest d_{min} of any codes possible with the same values of n and k
- Product Codes

Example Channel Coding Techniques

Product Codes

- k (number of information bits that get encoded) is a perfect square number $(4, 9, 16, 25, \ldots)$
- $n = \left(\sqrt{k} + 1\right)^2 = k + 1 + 2\sqrt{k}$ is the number of encoded bits
- Encoding Strategy:
  - Place the information bits in $\sqrt{k} \times \sqrt{k}$ array
  - Encode each row with an SPC
  - Encode each column with an SPC

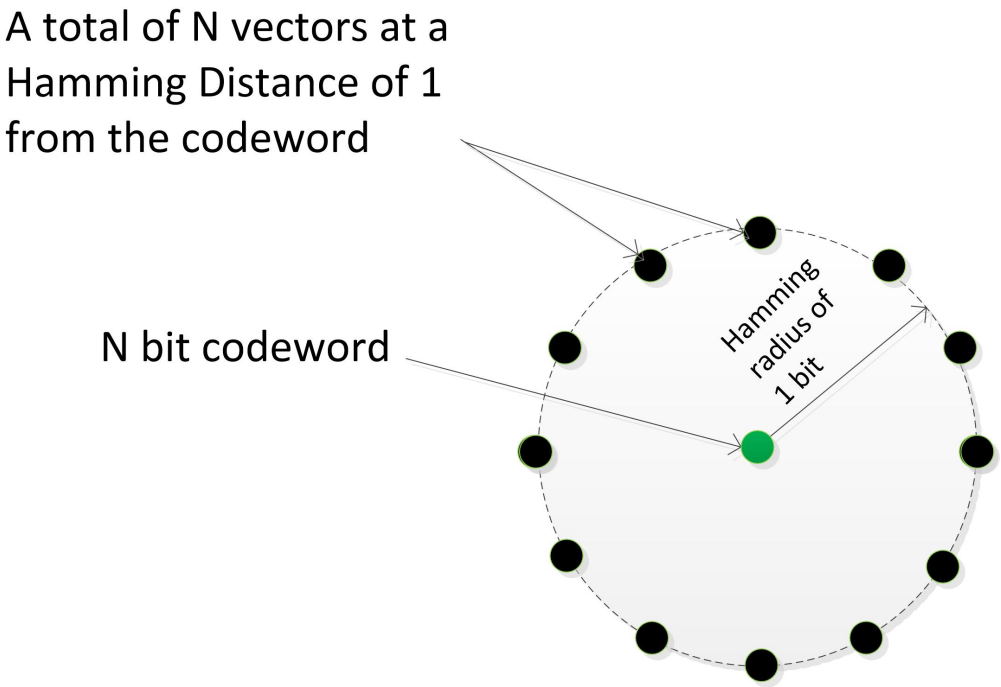
Hamming Weight and Hamming Distance
for Binary Sequences

- *Hamming Weight* is simply the number of ones in the sequence
- *Hamming Distance d*:
  - → Let $\mathbf{c}_m$ and $\mathbf{c}_n$ be two codewords, and $\mathbf{e}_{m,n} = \mathbf{c}_m \oplus \mathbf{c}_n$ be the difference vector (also binary) between these codewords
    - ▷ $\mathbf{e}_{m,n}$ has ones only in those places where the bits of $\mathbf{c}_m$ and $\mathbf{c}_n$ are differing; $\mathbf{e}_{m,n}$ is zero otherwise
  - → Hamming Distance d (or, more accurately $d_{m,n}$) between $\mathbf{c}_m$ and $\mathbf{c}_n$ is the Hamming Weight of $\mathbf{e}_{m,n}$
    - ▷ Hamming Distance is simply the number of places in which two binary sequences differ

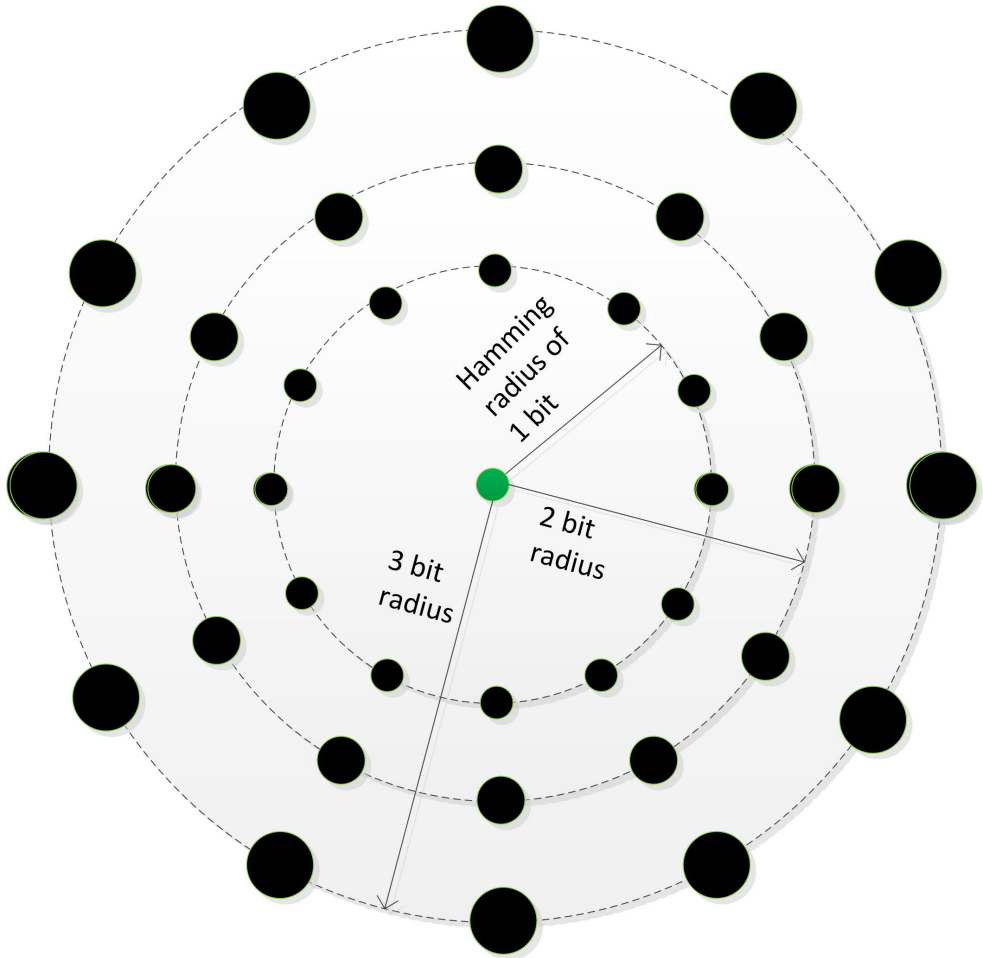
Minimum Hamming Distance d_{min}
for A Channel Code

- d_{min} is defined for a channel code with rate $r = \frac{K}{N}$
  - $\rightarrow$ This code takes K bit information sequence and generates N bit codeword
    - Thus, there are a total of 2^K codewords
- d_{min} for this channel code is the minimum Hamming distance between any two pairs of this channel code
- d_{min} relates to the error *detection* and error *correction* capabilities of the channel code. This can be visualized by drawing Hamming Circles as shown next

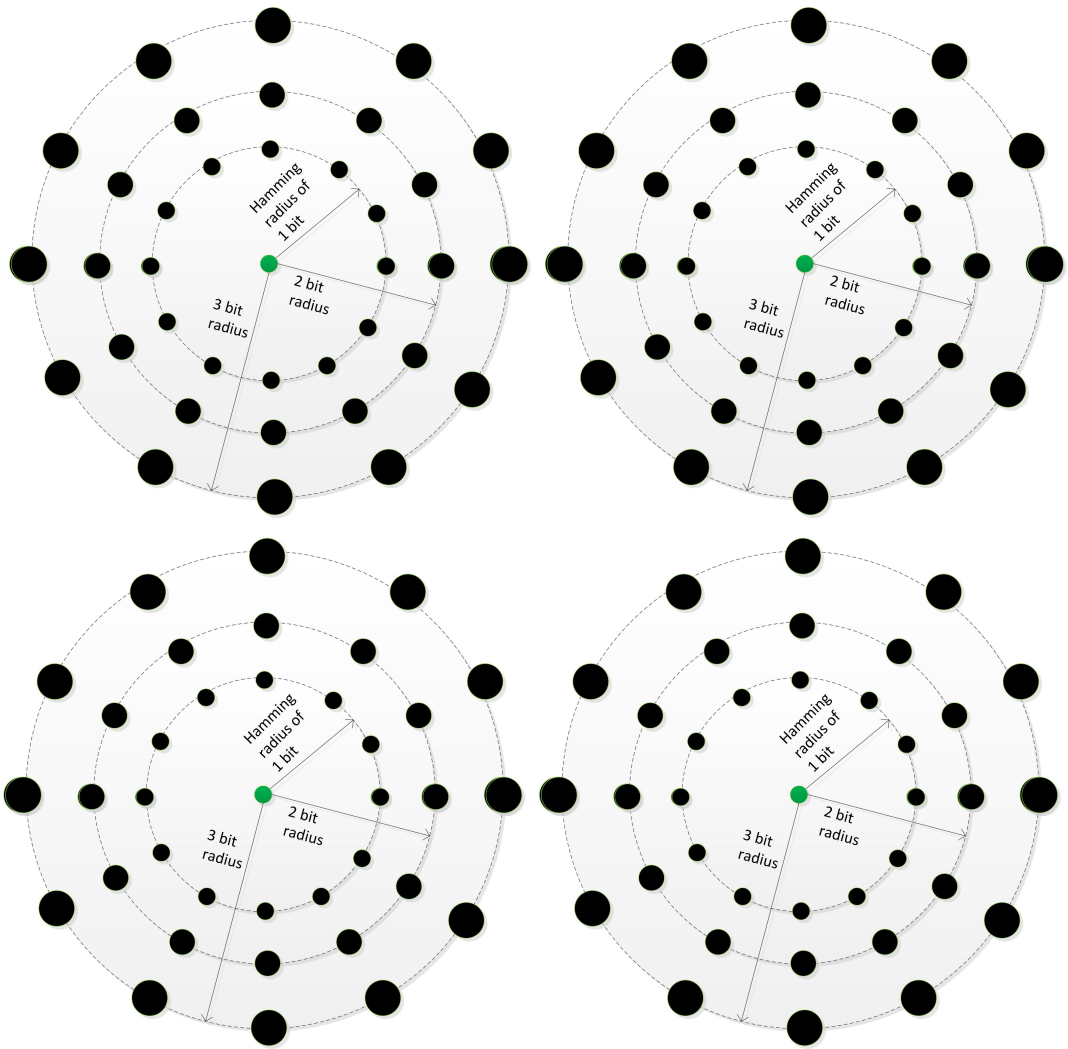
Hamming Circle of Radius 1 around a codeword



Hamming Circles around a codeword



Hamming Circles around several codewords



Mutual Information

- Consider the channel as a "black box". Let X denote the input and Y denote the output of this box.
- Mutual Information $I(X;Y)$ is defined in three, completely equivalent, ways as follows:

$$\begin{aligned} I(X;Y) &= H(X) - H(X|Y) \\ &= H(Y) - H(Y|X) \\ &= H(X) + H(Y) - H(X,Y) \end{aligned}$$

Mutual Information

- Channel Capacity $C = I(X;Y)$.
- $H(X)$ and $H(Y)$ are the Entropies of X and Y, respectively. Requires the knowledge of the individual PMFs or PDFs of $p_X(x)$ and $p_Y(y)$
- $H(X|Y)$ is called the *conditional* entropy of X given Y. Requires the knowledge of conditional PMF or PDF $p_{X|Y}(x|y)$
- Similarly, $H(Y|X)$ is called the *conditional* entropy of Y given X. Requires the knowledge of conditional PMF or PDF $p_{Y|X}(y|x)$
- Finally, $H(X,Y)$ is called the *joint* entropy of X and Y. Requires the knowledge of joint PMF or PDF $p_{X,Y}(x,y)$

Mutual Information
for BSC

- For BSC,
  - → $H(Y)$ is 1 bit.
    - ▷ This occurs when the receiver has highest degree of uncertainty about the bits that are received over the BSC. The probability of bits, as measured by the receiver, is equal and it is 0.5
  - → $H(Y|X)$ equals $H_b(p)$ bit.
    - ▷ This is because, given X, the only source of randomness in Y arises from the BSC introduced noise vector $\mathbf{n}$. Therefore, the entropy of Y, *conditioned* upon X, is the entropy of BSC, i.e., $H_b(p)$
- Thus, $I(X;Y) = 1 - H_b(p)$ bits for the BSC.

Mutual Information
for Analog Gaussian Noise Channel

- For the analog channel in which the transmitter sends an average power of P_s Watts, and the receiver introduces Gaussian noise of power P_n watts,
  - $\rightarrow$ $H(Y)$ is $\frac{1}{2}\log_2(P_s + P_n)$ bits.
    - This occurs when the receiver has highest degree of uncertainty about the analog signal that is received.
  - $\rightarrow$ $H(Y|X)$ equals $\frac{1}{2}\log_2(P_n)$ bit.
    - This is because, given X, the only source of randomness in Y arises from the analog Gaussian noise vector $\mathbf{n}$. Therefore, the entropy of Y, *conditioned* upon X, is the entropy of this Gaussian noise, and its entropy equals $\frac{1}{2}\log_2(P_n)$ bit.
- Thus, $I(X;Y) = \frac{1}{2}\log_2(P_s + P_n) - \log_2(P_n) = \frac{1}{2}\log_2\left(1 + \frac{P_s}{P_n}\right)$ bits.

Mutual Information
for Analog Gaussian Noise Channel

- According to Shannon-Nyquist sampling theorem, if the analog waveforms used for communication have a bandwidth of W Hz, the number of samples per second required is $2W$ samples/second
- Therefore, $I(X;Y) = C = W \log_2 \left(1 + \frac{P_s}{P_n}\right)$ bits/second
- This is the celebrated channel capacity theorem for analog Gaussian noise channels