CS3236 Introduction to Information theory

Lecture 1: Introduction

Course given by Pierre Senellart
Material by Stephanie Wehner, with additions by P. Senellart

Class survival guide

Class survival guide

Lecture on Monday 2-4pm

Class survival guide

Lecture on Monday 2-4pm

- Tutorial on Thursday 2-3pm, starting September 4

    Discussion, QA, help with homework exercises

    - Discussion on projects

Class survival guide

Lecture on Monday 2-4pm

- Tutorial on Thursday 2-3pm, starting September 4
  - Discussion, QA, help with homework exercises
  - Discussion on projects
- Recess week Sep 20-28

Class survival guide

Lecture on Monday 2-4pm

- Tutorial on Thursday 2-3pm, starting September 4
  - Discussion, QA, help with homework exercises
  - Discussion on projects
- Recess week Sep 20-28
- No physical lecture/tutorial on Sep 8-12, e-Learning material

Class survival guide

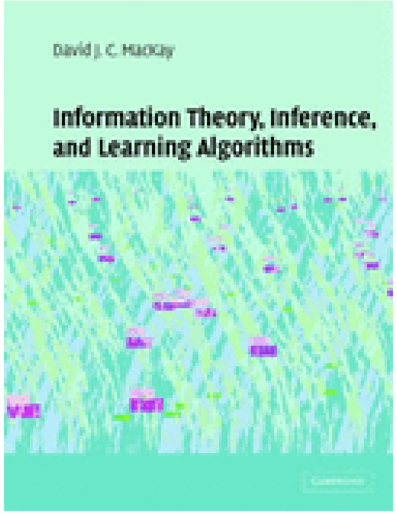
- Lecture on Monday 2-4pm
- Tutorial on Thursday 2-3pm, starting September 4
  - Discussion, QA, help with homework exercises
  - Discussion on projects
- Recess week Sep 20-28
- No physical lecture/tutorial on Sep 8-12, e-Learning material
- Available to meet for answering individual questions. Use http://www.doodle.com/psenellart to schedule a meeting. Office ICube #03-09.

Class survival guide

Lecture on Monday 2-4pm
- Tutorial on Thursday 2-3pm, starting September 4
  - Discussion, QA, help with homework exercises
  - Discussion on projects
- Recess week Sep 20-28
- No physical lecture/tutorial on Sep 8-12, e-Learning material
- Available to meet for answering individual questions. Use http://www.doodle.com/psenellart to schedule a meeting. Office ICube #03-09.
- Grading
  - 50% Exam (examination period: Nov 22 to Dec 6)
  - 50% Continuous Assessment
    - 10% Mid-Term exam (October 13)
    - 20% Homework (assigned each week, due next Monday)
    - 20% Small project (handed out September 1, due November 14)

Class survival guide

- IVLE – Slides, materials, discussion board. Make use of it
- Book (also available in electronic form for free from http://www.inference.phy.cam.ac.uk/itila/)



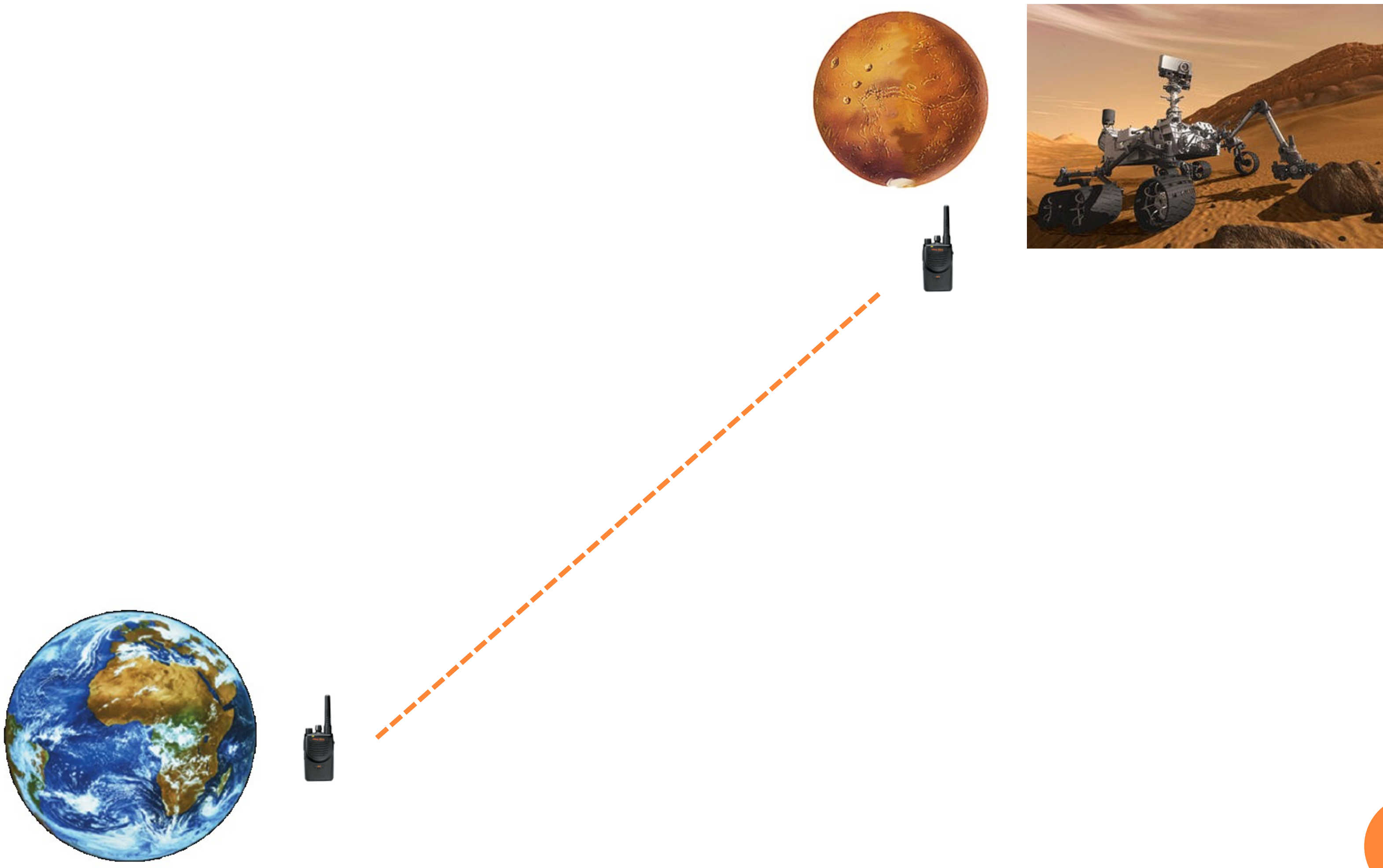
Information Theory, Inference and Learning Algorithms
by David J. C. MacKay
Cambridge University Press, 2003

A PICTURE FROM MARS

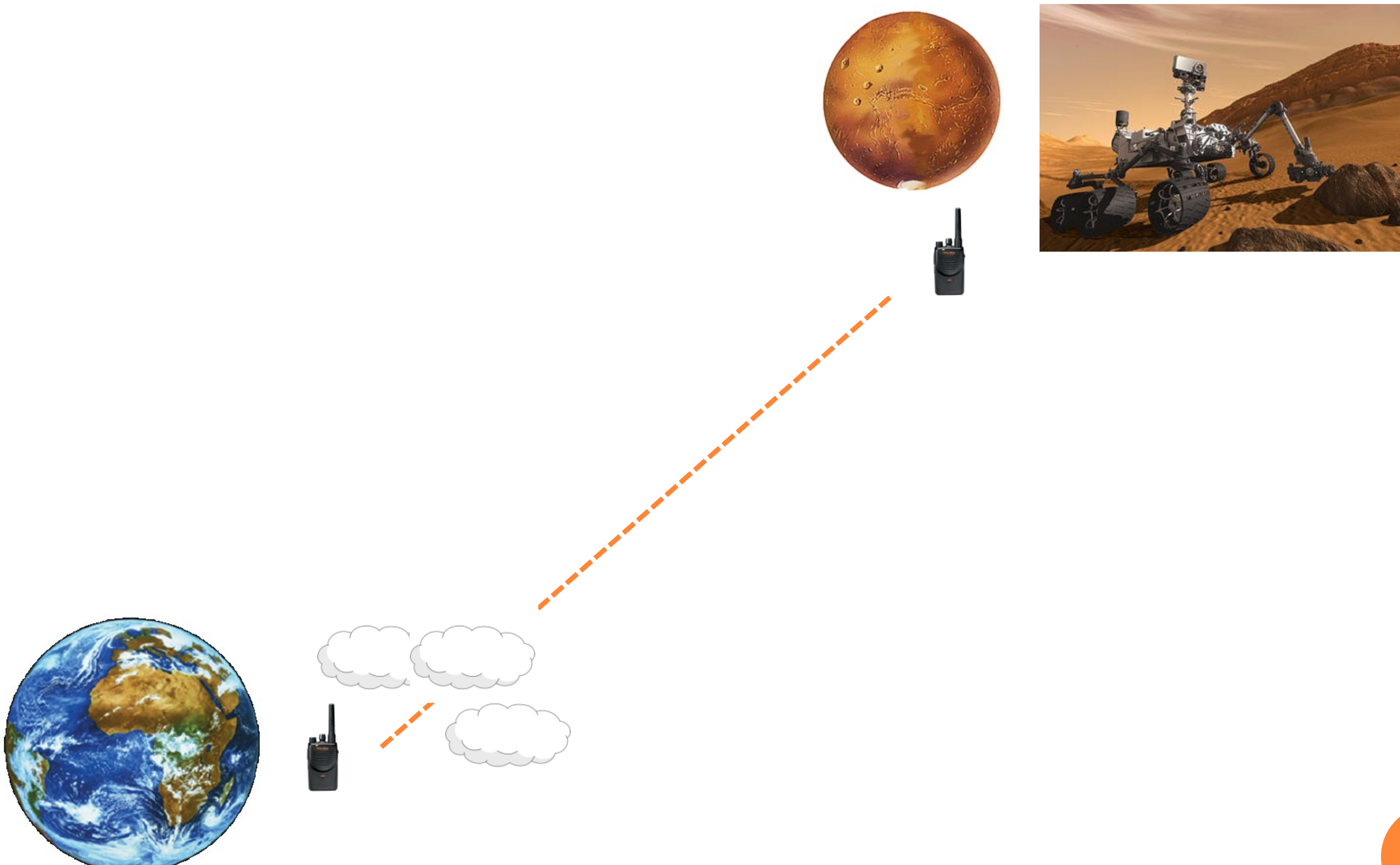
A PICTURE FROM MARS

How did we manage to get this image from Mars to Earth?

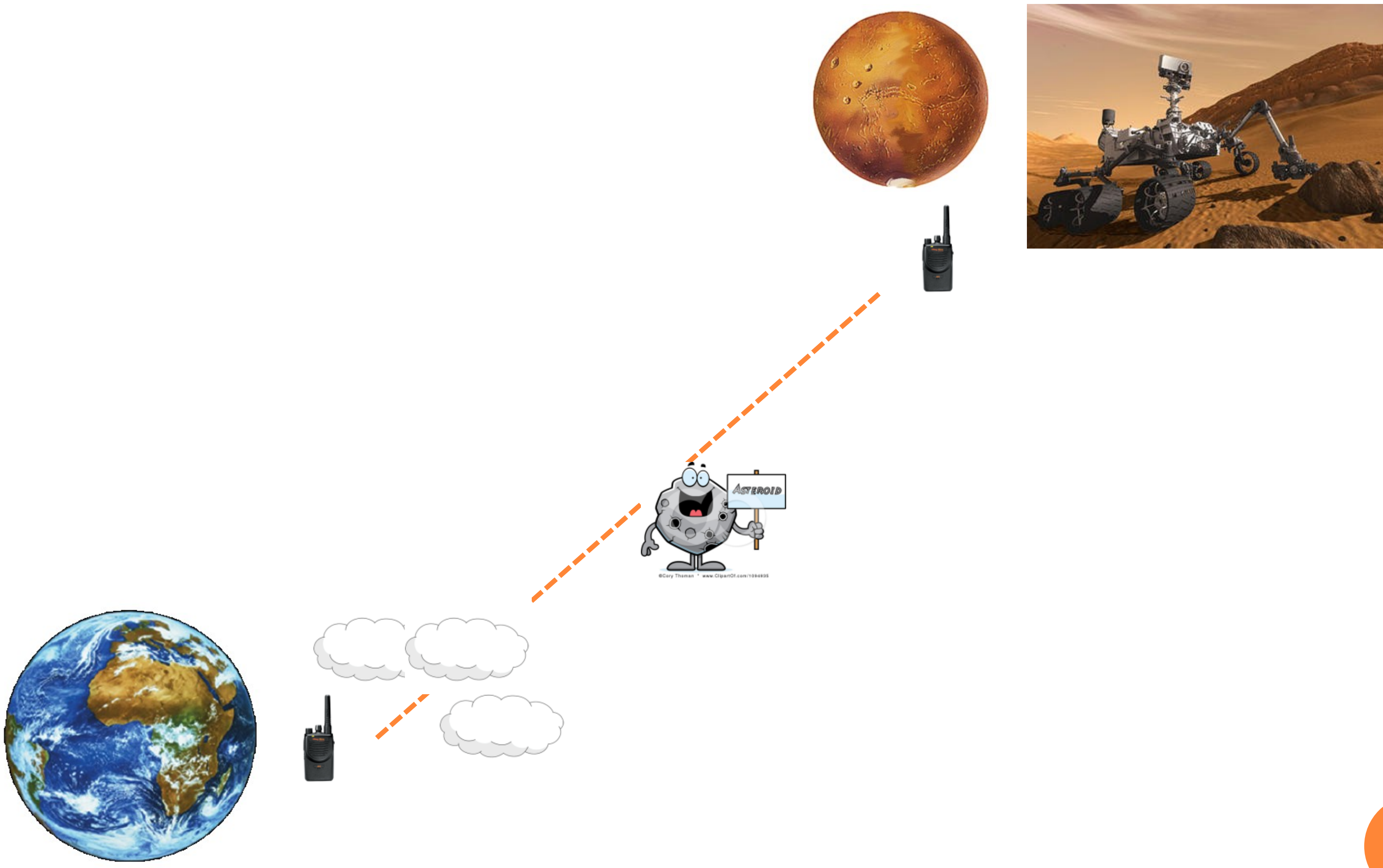
Problem of information transmission



Problem of information transmission

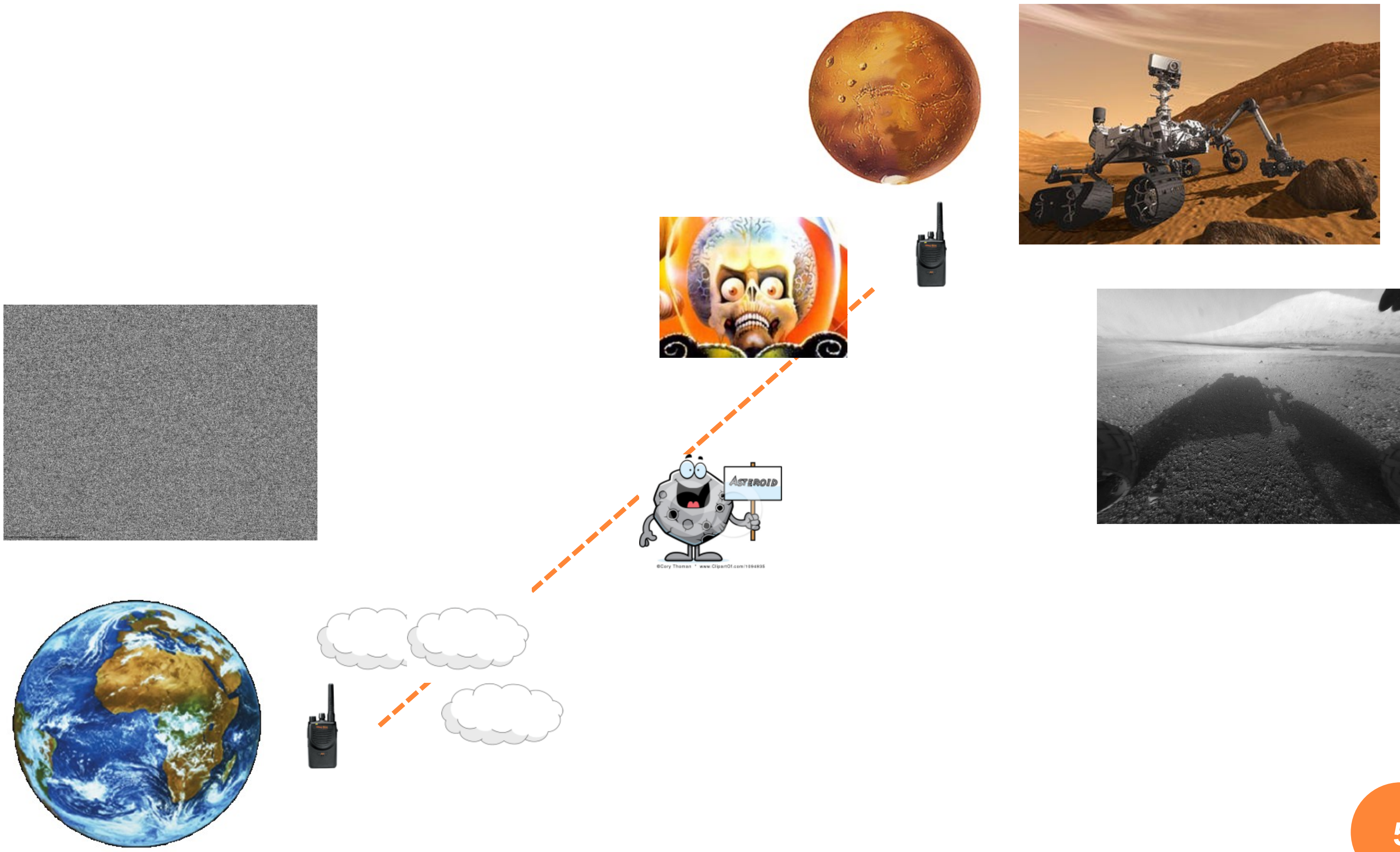


Problem of information transmission

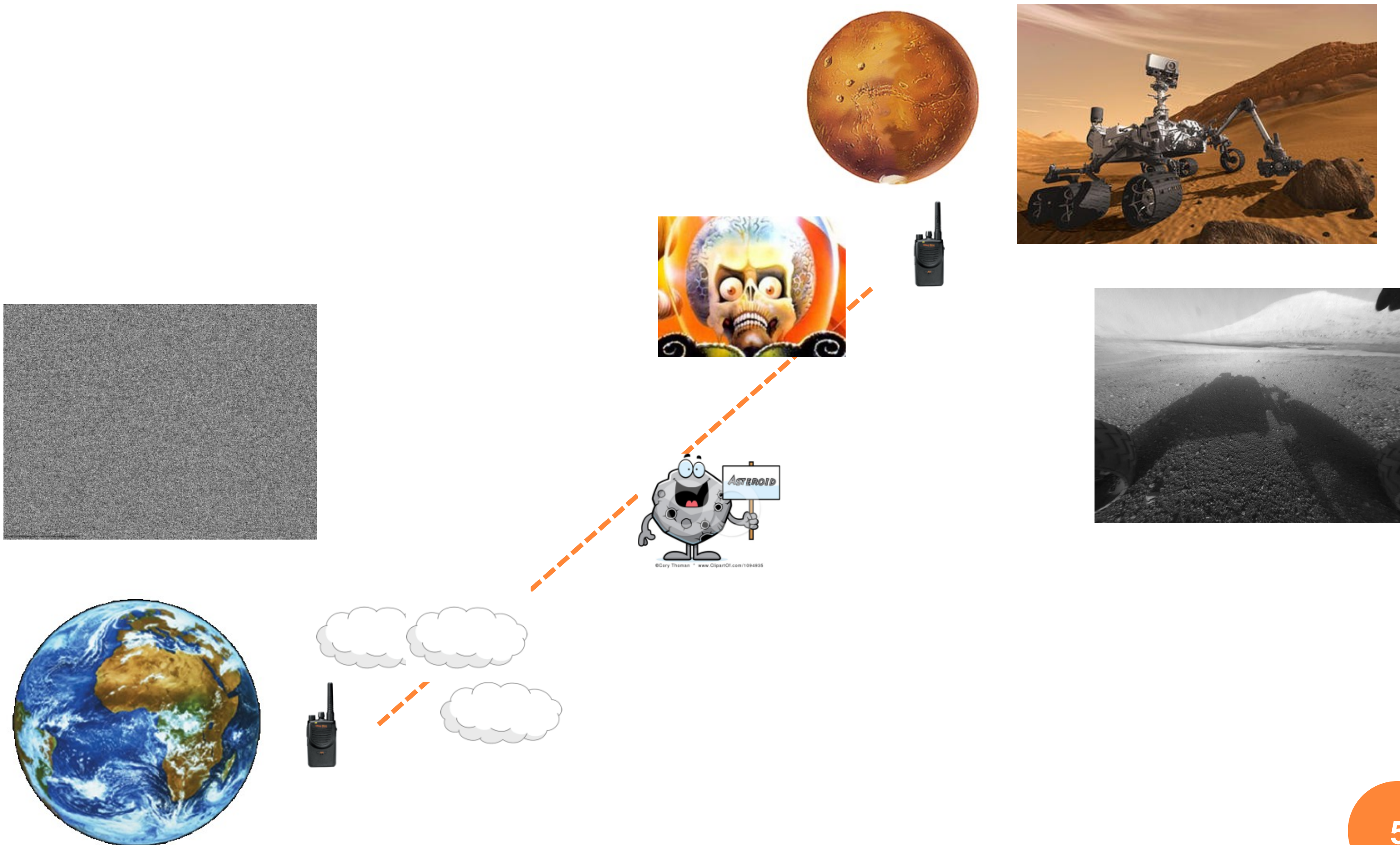


Problem of information transmission

PROBLEM OF INFORMATION TRANSMISSION



PROBLEM OF INFORMATION TRANSMISSION



How on earth can we still get the picture???

Problem of information transmission

How can we transmit voice signals correctly?

Problem of information transmission

How can we store and retrieve information from a hard disk?

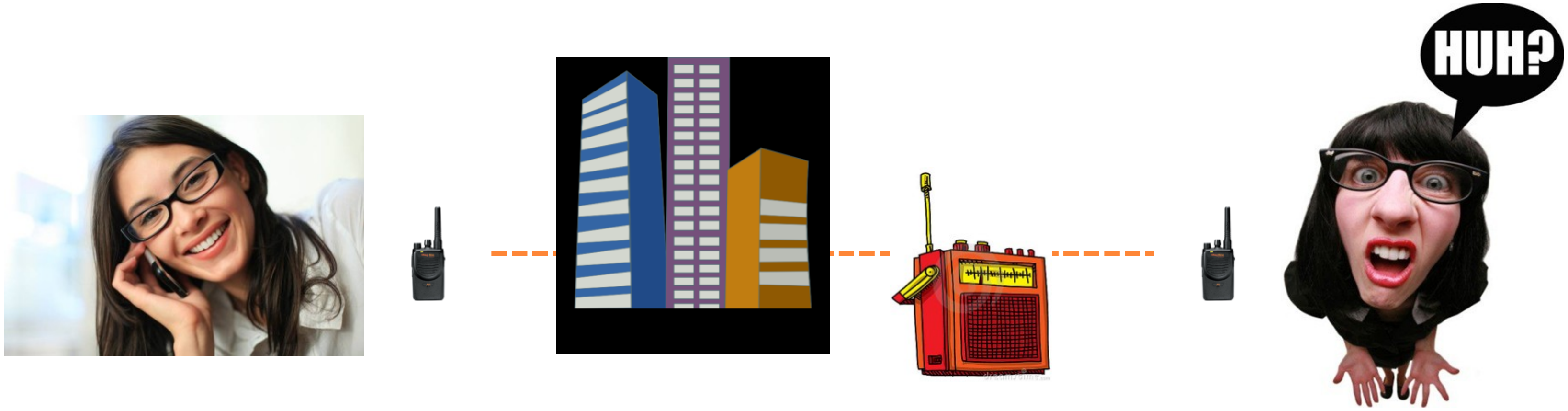
Problem of information transmission

How can biological systems store and retrieve genetic information?

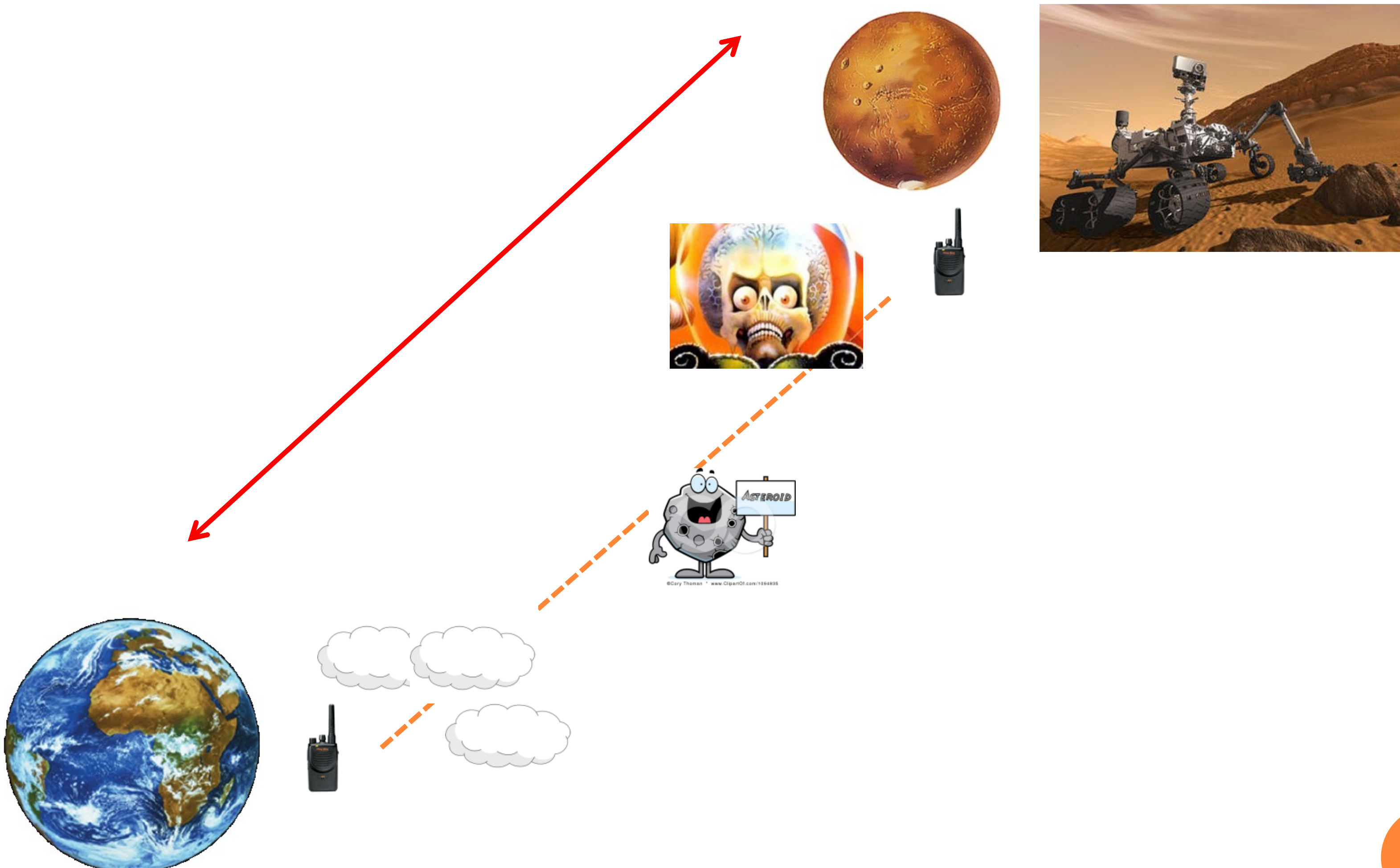
Goal of information theory

What is the best way to store or transmit information?

Method 1: if it fails, send again

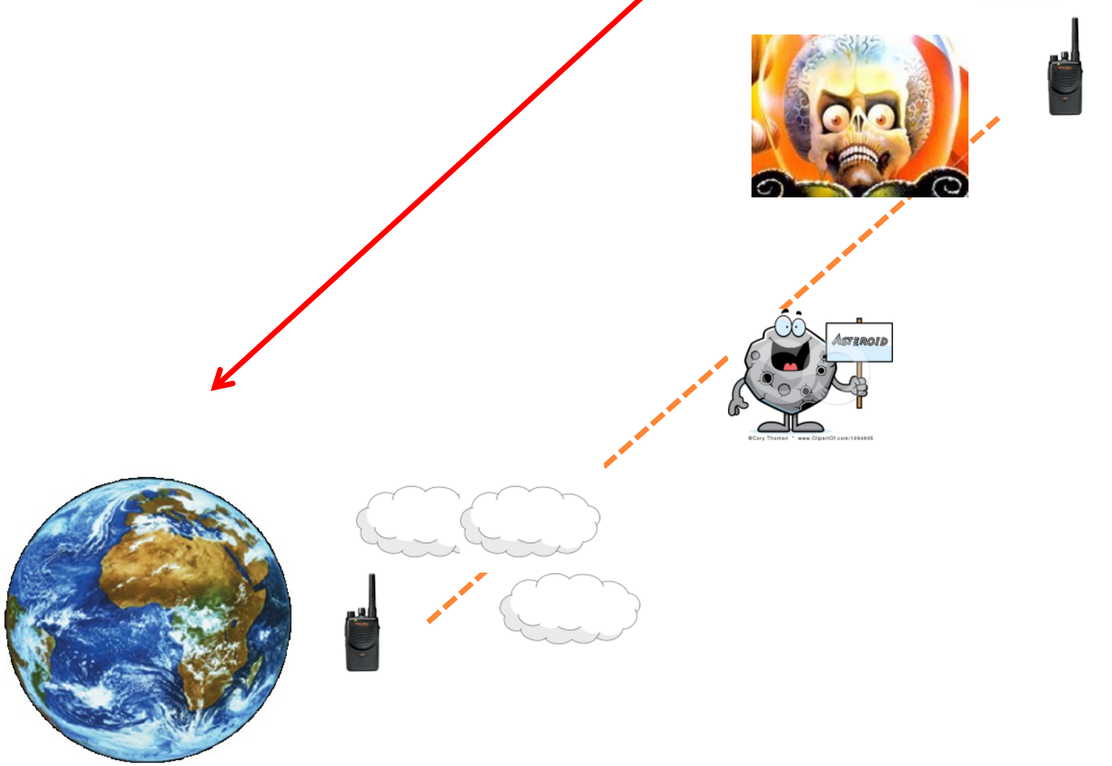


METHOD 1: IF IT FAILS, SEND AGAIN



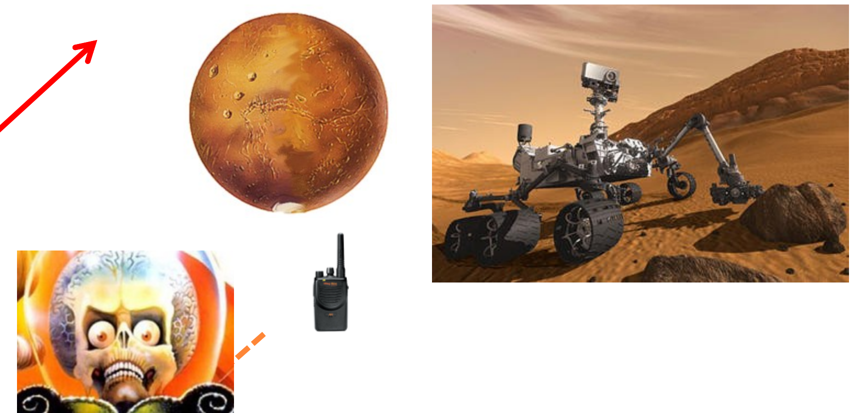
Method 1: if it fails, send again

Takes roughly 4.3 minutes to send anything!



METHOD 1: IF IT FAILS, SEND AGAIN

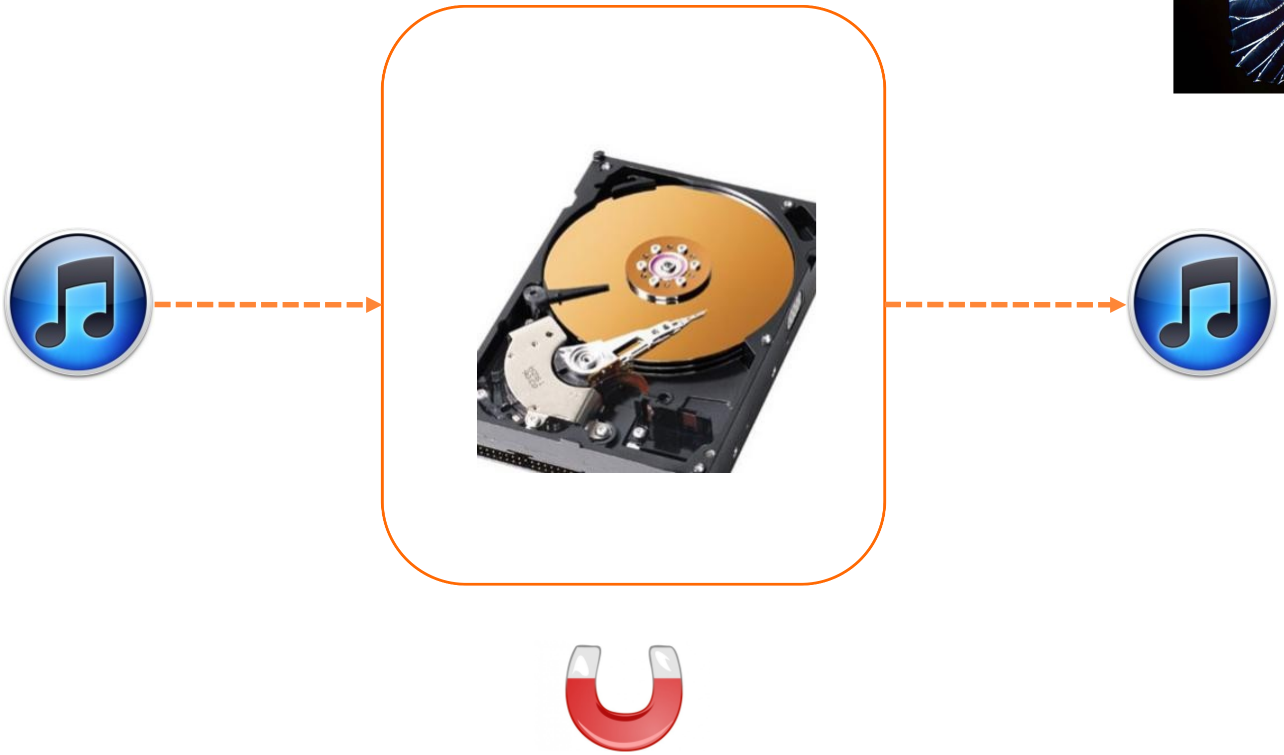
Takes roughly 4.3 minutes to send anything!



… and how do we even know it went wrong??

Need at least *error detection*!

Method 1: If it fails, send again

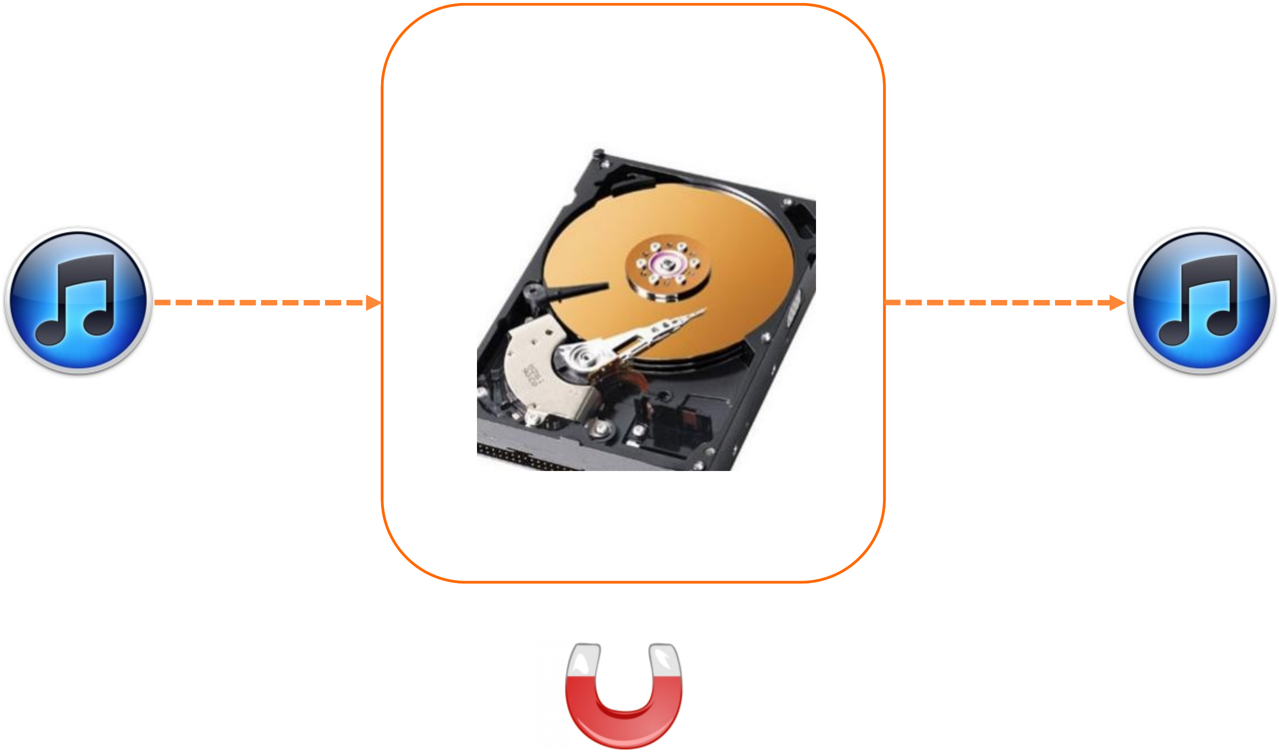


METHOD 1: IF IT FAILS, SEND AGAIN

The moment were we want to retrieve the data may be far in the future…. when original is no longer available!

Your brain already has an amazing amount of error detection and memory built in ☺

Method 2: Improve situation



Reduce the amount of errors by making a better hard disk

Very expensive!

Method 2: Improve the situation

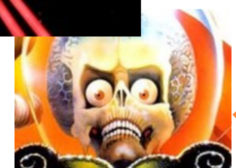
METHOD 2: IMPROVE THE SITUATION

Method 2: Improve the situation

METHOD 2: IMPROVE THE SITUATION

Method 2: Improve the situation




Not possible to eliminate all errors!

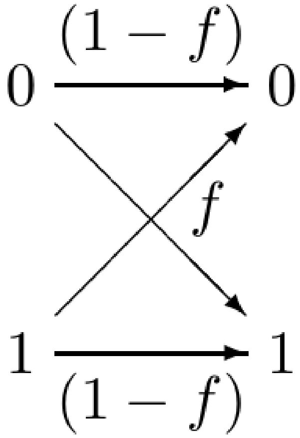
Goal of Information Theory

What is the best way to store or transmit information?

Without changing the situation!

What is noise?

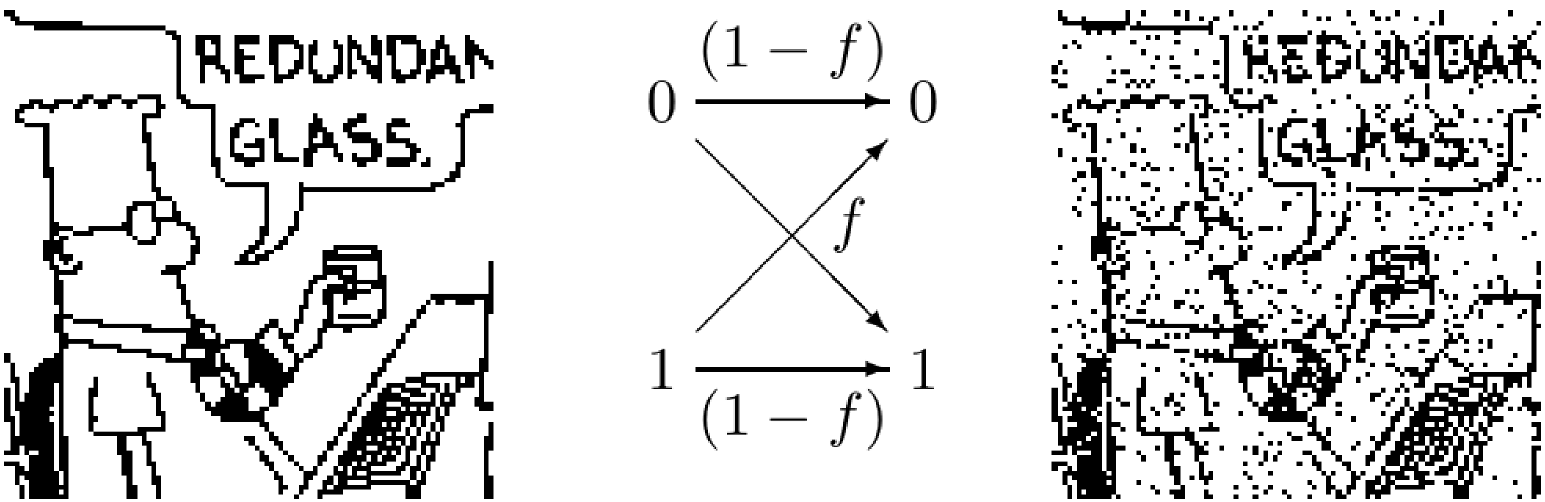
- Example: binary symmetric channel



Flips the input bit with probability *f*

What is noise?

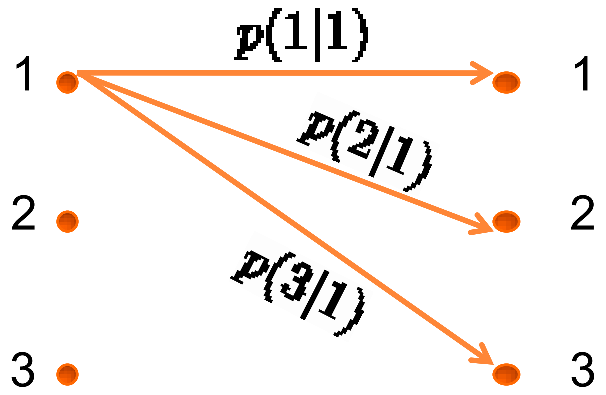
- Any noise channel is given by conditional probability distributions mapping inputs to output



Correct: $p(0|0) = p(1|1) = 1 - f$

Error: $p(1|0) = p(0|1) = f$

WHAT IS NOISE?

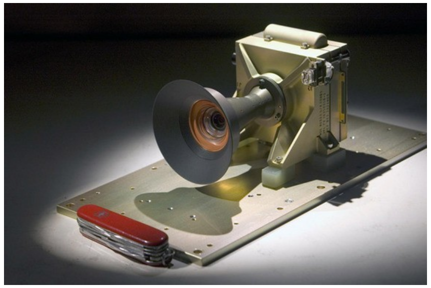


Any noisy channel is given by such conditional probabilities.

“Using the channel”: give one input and receive one output

What is a source (of Information)?

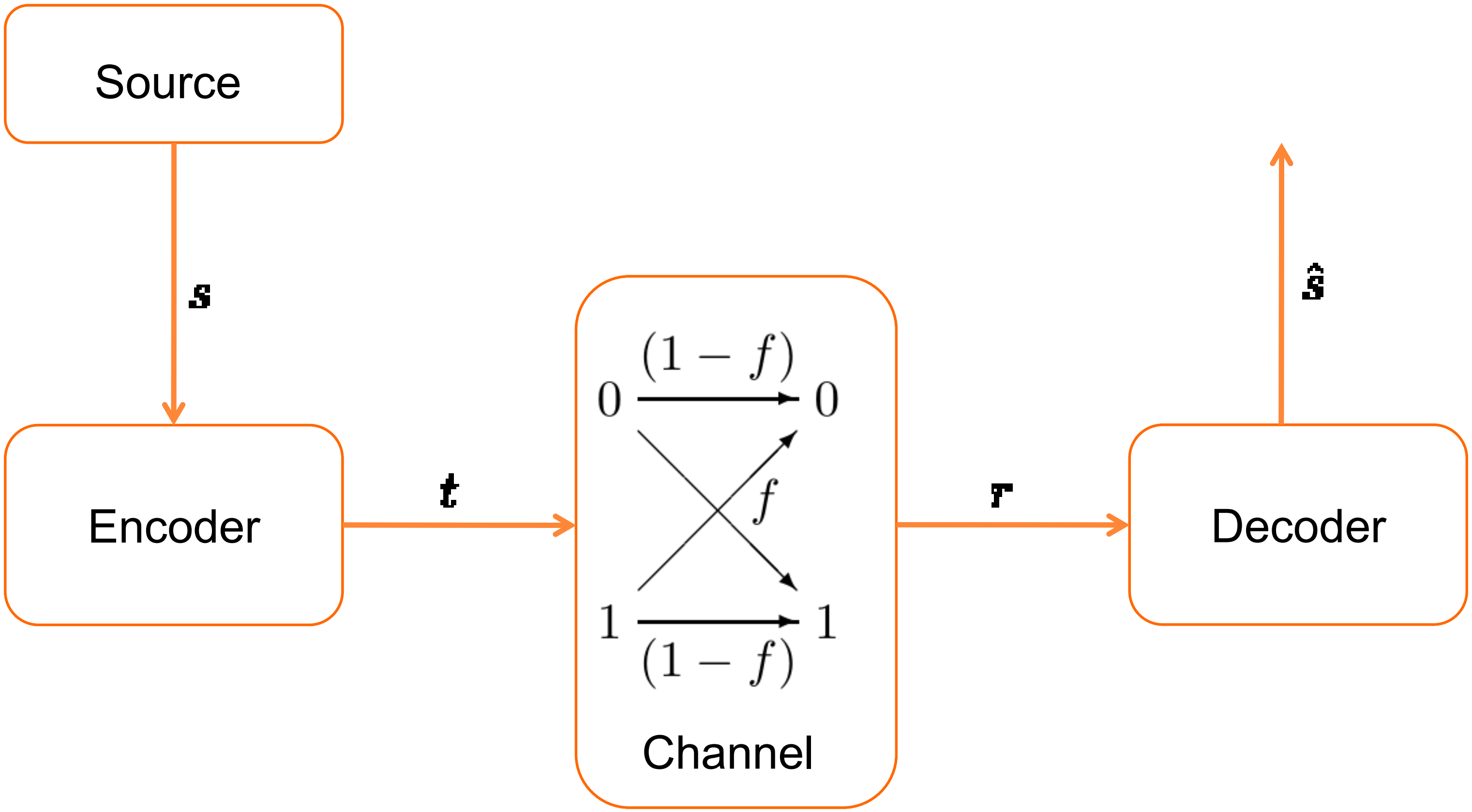
- Determines the information we want to send



- Mathematically, probability distribution

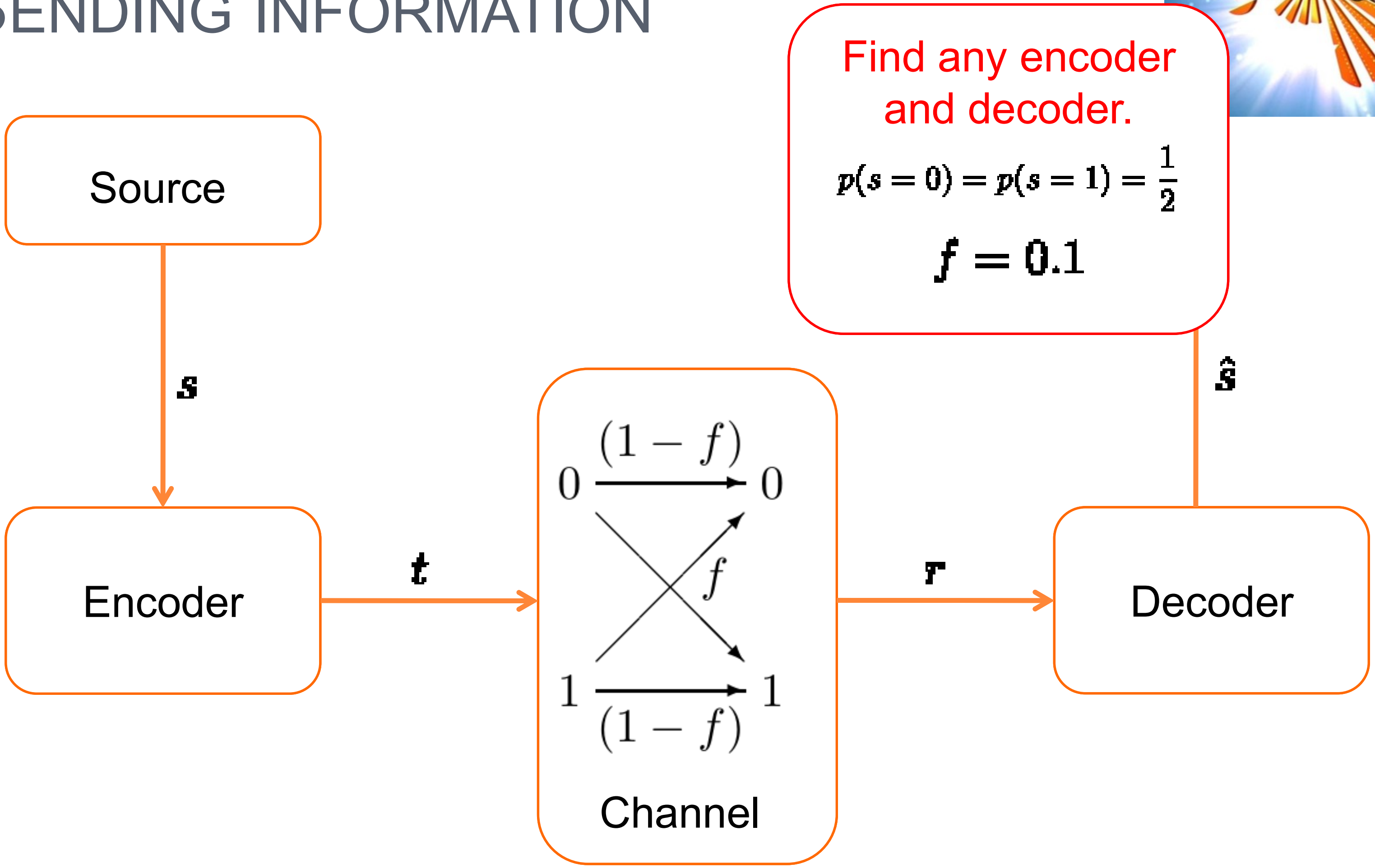
$$p(s=0)=p(s=1)=\frac{1}{2}$$

SENDING INFORMATION



Goal: Construct an encoder and decoder such that we can recover the information $\hat{\mathbf{s}} = \mathbf{s}$

SENDING INFORMATION



Goal: Construct an encoder and decoder such that we can recover the information $\hat{s} = s$

Repetition code: Encoder

- Send each input symbol 3 times (Code R_3)

$$s = 0 \qquad t = 000$$

$$s = 1 \qquad t = 111$$

- Errors can be described by adding a noise vector

s	0	0	1	0	1	1	0
t	000	000	111	000	111	111	000
n	000	001	000	000	101	000	000
r	000	001	111	000	010	111	000

$$0 \oplus 0 = 1 \oplus 1 = 0$$

$$0 \oplus 1 = 1 \oplus 0 = 1$$

REPETITION CODE: DECODER

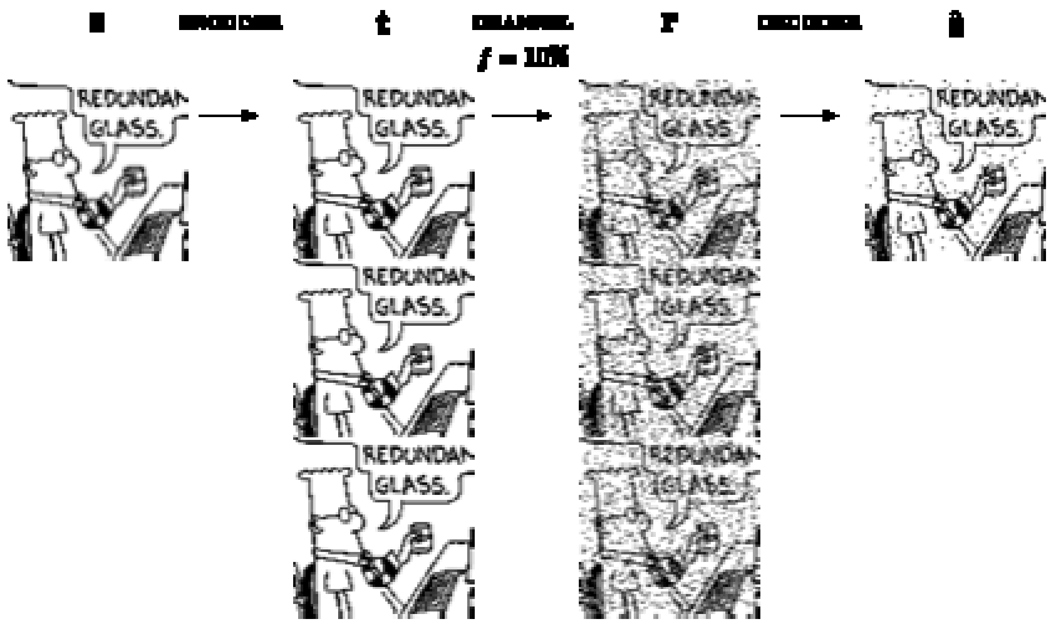
s	0	0	1	0	1	1	0
t	000	000	111	000	111	111	000
n	000	001	000	000	101	000	000
r	000	001	111	000	010	111	000
$\hat{\mathbf{s}}$	0	0	1	0	0	1	0
corrected errors		⋆					
undetected errors					⋆		

- Compute the majority of each 3 bit block

Repetition code: Decoder

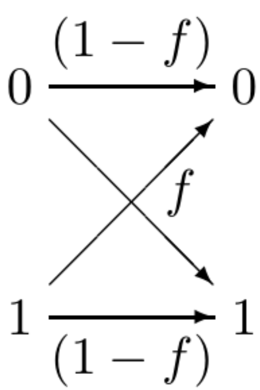
s	0	0	1	0	1	1	0
t	000	000	111	000	111	111	000
n	000	001	000	000	101	000	000
r	000	001	111	000	010	111	000
ŝ	0	0	1	0	0	1	0
corrected errors		⋆					
undetected errors					⋆		

- Compute the majority of each 3 bit block



How good is our coding scheme?

- Coding scheme: an encoder and a decoder
- Want to compute the probability that our reconstruction is *incorrect*

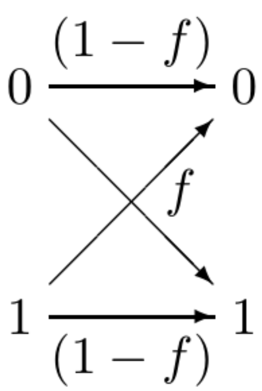


Channel

s	0	0	1	0	1	1	0
t	000	000	111	000	111	111	000
n	000	001	000	000	101	000	000
r	000	001	111	000	010	111	000
$\hat{s}$	0	0	1	0	0	1	0
corrected errors		⋆					
undetected errors					⋆		

How good is our coding scheme?

- Coding scheme: an encoder and a decoder
- Want to compute the probability that our reconstruction is *incorrect*



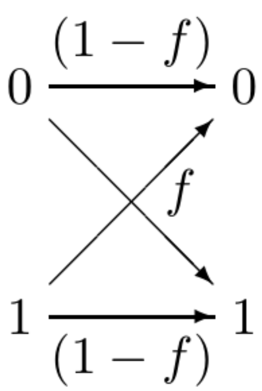
Channel

s	0	0	1	0	1	1	0
t	000	000	111	000	111	111	000
n	000	001	000	000	101	000	000
r	000	001	111	000	010	111	000
ŝ	0	0	1	0	0	1	0
corrected errors		⋆					
undetected errors					⋆		

$$p_{\text{error}} = \sum_{s} p(s)p(\hat{s} \neq s|s) = \frac{1}{2}\sum_{s} p(\hat{s} \neq s|s)$$

How good is our coding scheme?

- Coding scheme: an encoder and a decoder
- Want to compute the probability that our reconstruction is *incorrect*



Channel

s	0	0	1	0	1	1	0
t	000	000	111	000	111	111	000
n	000	001	000	000	101	000	000
r	000	001	111	000	010	111	000
ŝ	0	0	1	0	0	1	0
corrected errors		⋆					
undetected errors					⋆		

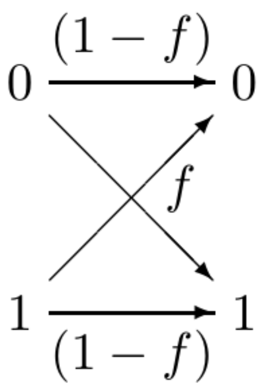
$$p_{\text{error}} = \sum_s p(s)p(\hat{s} \neq s|s) = \frac{1}{2}\sum_s p(\hat{s} \neq s|s)$$

$$p(\hat{s} \neq s|s) = 3f^2(1-f) + f^3 \approx 0.03$$

How good is our coding scheme?

- Coding scheme: an encoder and a decoder
- Want to compute the probability that our reconstruction is *incorrect*

s	0	0	1	0	1	1	0
t	000	000	111	000	111	111	000
n	000	001	000	000	101	000	000
r	000	001	111	000	010	111	000
$\hat{s}$	0	0	1	0	0	1	0
corrected errors		⋆					
undetected errors					⋆		



Channel

$$p_{\text{error}} = \sum_s p(s) p(\hat{s} \neq s | s) = \frac{1}{2} \sum_s p(\hat{s} \neq s | s)$$

$$p(\hat{s} \neq s | s) = 3f^2(1-f) + f^3 \approx 0.03$$

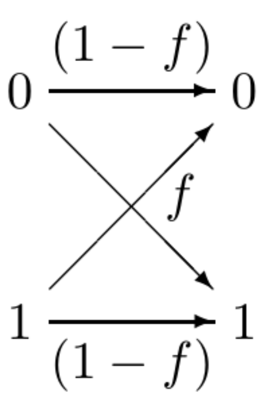
Exactly 2 flipped

How good is our coding scheme?

- Coding scheme: an encoder and a decoder

- Want to compute the probability that our reconstruction is *incorrect*

s	0	0	1	0	1	1	0
t	000	000	111	000	111	111	000
n	000	001	000	000	101	000	000
r	000	001	111	000	010	111	000
ŝ	0	0	1	0	0	1	0
corrected errors		⋆					
undetected errors					⋆		



Channel

$$p_{\text{error}} = \sum_s p(s) p(\hat{s} \neq s | s) = \frac{1}{2} \sum_s p(\hat{s} \neq s | s)$$

$$p(\hat{s} \neq s | s) = 3f^2(1-f) + f^3 \approx 0.03$$

Exactly 2 flipped

All 3 flipped

How good is our coding scheme?

- Makes an error with probability $p_{\text{error}} \approx 0.03$
- Not bad! Remember before we had an error with probability $f = 0.1$ so we improved!
- But… we had to send 3 bits for every 1 bit!
- Rate

$$R = \frac{\#\text{ of bits sent}}{\#\text{ of bits actually transmitted}} = \frac{|\mathbf{s}|}{|\mathbf{t}|} = \frac{1}{3}$$

Maybe there is a better decoder?

- For every 3 bits $\mathbf{r} = r_1 r_2 r_3$ we want to pick the most likely input s!

$$p(s=0|r_1r_2r_3) = \frac{p(r_1r_2r_3|s=0)p(s=0)}{p(r_1r_2r_3)} = \frac{p(r_1r_2r_3|s=0)}{2p(r_1r_2r_3)}$$

$$p(s=1|r_1r_2r_3) = \frac{p(r_1r_2r_3|s=1)p(s=1)}{p(r_1r_2r_3)} = \frac{p(r_1r_2r_3|s=1)}{2p(r_1r_2r_3)}$$

Maybe there is a better decoder?

- For every 3 bits $\mathbf{r} = r_1 r_2 r_3$ we want to pick the most likely input s!

$$p(s=0|r_1r_2r_3) = \frac{p(r_1r_2r_3|s=0)p(s=0)}{p(r_1r_2r_3)} = \frac{p(r_1r_2r_3|s=0)}{2p(r_1r_2r_3)}$$

Which one is bigger?

$$p(s=1|r_1r_2r_3) = \frac{p(r_1r_2r_3|s=1)p(s=1)}{p(r_1r_2r_3)} = \frac{p(r_1r_2r_3|s=1)}{2p(r_1r_2r_3)}$$

Maybe there is a better decoder?

- For every 3 bits $\mathbf{r} = r_1 r_2 r_3$ we want to pick the most likely input s!

$$p(s=0|r_1r_2r_3) = \frac{p(r_1r_2r_3|s=0)p(s=0)}{p(r_1r_2r_3)} = \frac{p(r_1r_2r_3|s=0)}{2p(r_1r_2r_3)}$$

Which one is bigger?

$$p(s=1|r_1r_2r_3) = \frac{p(r_1r_2r_3|s=1)p(s=1)}{p(r_1r_2r_3)} = \frac{p(r_1r_2r_3|s=1)}{2p(r_1r_2r_3)}$$

Compute the likelihood ratio: $\frac{p(r_1r_2r_3|s=1)}{p(r_1r_2r_3|s=0)}$

MAYBE THERE IS A BETTER DECODER?

- For every 3 bits $\mathbf{r} = r_1 r_2 r_3$ we want to pick the most likely input s!

$$p(s=0|r_1r_2r_3) = \frac{p(r_1r_2r_3|s=0)p(s=0)}{p(r_1r_2r_3)} = \frac{p(r_1r_2r_3|s=0)}{2p(r_1r_2r_3)}$$

Which one is bigger?

$$p(s=1|r_1r_2r_3) = \frac{p(r_1r_2r_3|s=1)p(s=1)}{p(r_1r_2r_3)} = \frac{p(r_1r_2r_3|s=1)}{2p(r_1r_2r_3)}$$

Compute the likelihood ratio: $$\frac{p(r_1r_2r_3|s=1)}{p(r_1r_2r_3|s=0)}$$ < 1 ? > 1 ?

MAYBE THERE IS A BETTER DECODER?

Compute the likelihood ratio: $\dfrac{p(r_1 r_2 r_3 | s = 1)}{p(r_1 r_2 r_3 | s = 0)}$

- Claim: Majority vote is the best thing we can do, because it gives us the same reconstruction as picking the most likely one!

$$\frac{p(r_1 r_2 r_3 | s = 1)}{p(r_1 r_2 r_3 | s = 0)} = \prod_{n=1}^{3} \frac{p(r_n | t_n(1))}{p(r_n | t_n(0))}$$

$$\frac{p(r_n | t_n(1))}{p(r_n | t_n(0))} = \begin{cases} \gamma & \text{if } r_n = 1 \\ \gamma^{-1} & \text{if } r_n = 0 \end{cases}$$

$$\gamma = \frac{1-f}{f} > 1$$

Maybe there is a better decoder?

- Let's compare majority decisions to the most likely one

$$\frac{p(r_1 r_2 r_3 | s=1)}{p(r_1 r_2 r_3 | s=0)} = \prod_{n=1}^{3} \frac{p(r_n | t_n(1))}{p(r_n | t_n(0))}$$

$$\gamma = \frac{1-f}{f} > 1$$

$$\frac{p(r_n | t_n(1))}{p(r_n | t_n(0))} = \begin{cases} \gamma & \text{if } r_n = 1 \\ \gamma^{-1} & \text{if } r_n = 0 \end{cases}$$

Received sequence $\mathbf{r}$	Likelihood ratio $\frac{P(\mathbf{r} \mid s=1)}{P(\mathbf{r} \mid s=0)}$	Decoded sequence $\hat{\mathbf{s}}$
000	γ^{-3}	0
001	γ^{-1}	0
010	γ^{-1}	0
100	γ^{-1}	0
101	γ^{1}	1
110	γ^{1}	1
011	γ^{1}	1
111	γ^{3}	1

How could we improve our coding scheme?

- Suggestions?

How could we improve our coding scheme?

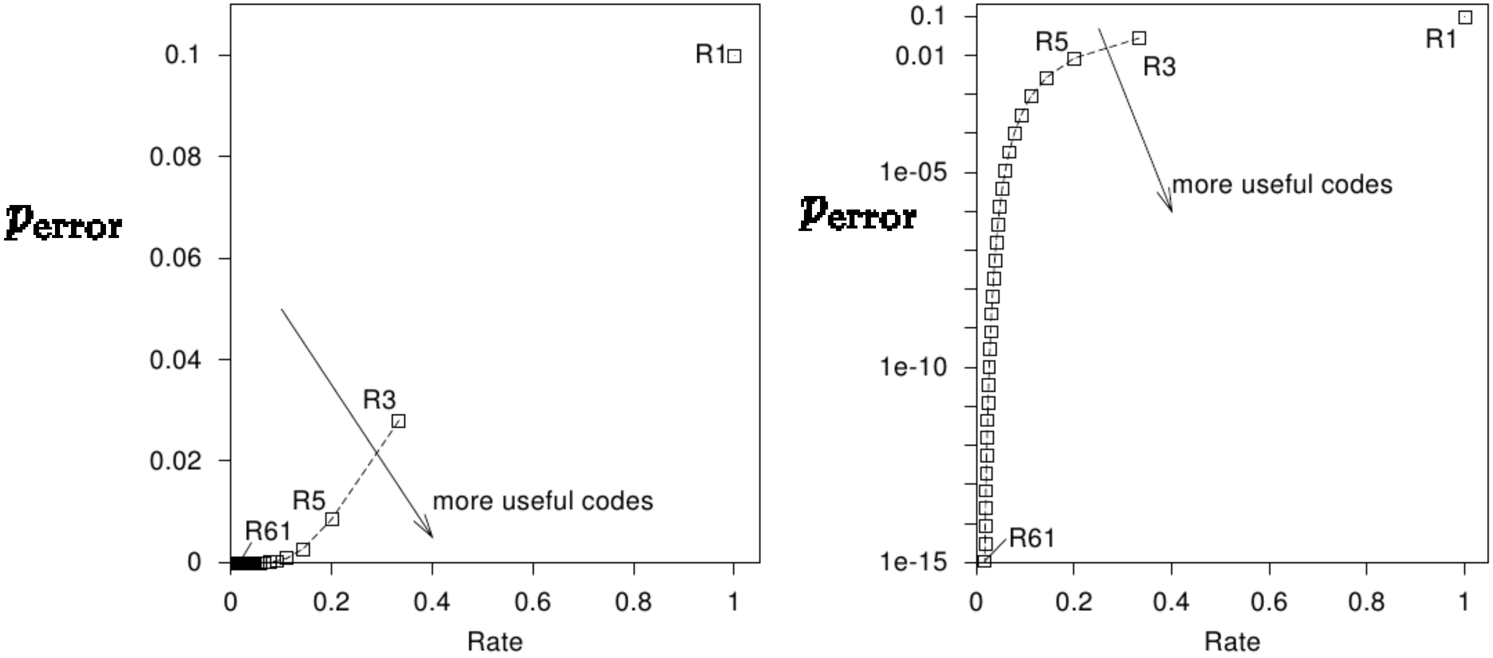
- Suggestions?

  Use more repetitions!

How could we improve our coding scheme?

- Suggestions?

  Use more repetitions!

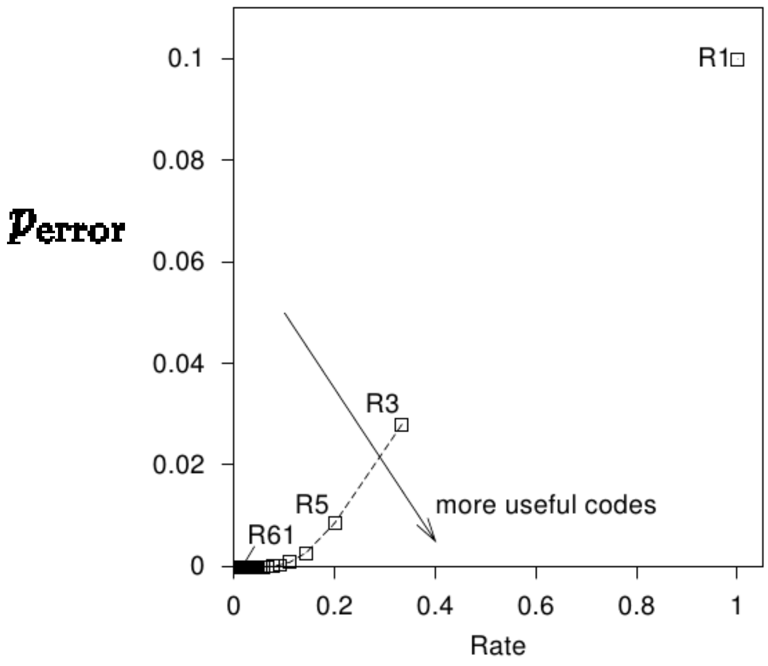


How could we improve our coding scheme?

- Suggestions?

  Use more repetitions!

To make the error arbitrarily small we need to send an arbitrarily large number of bits!



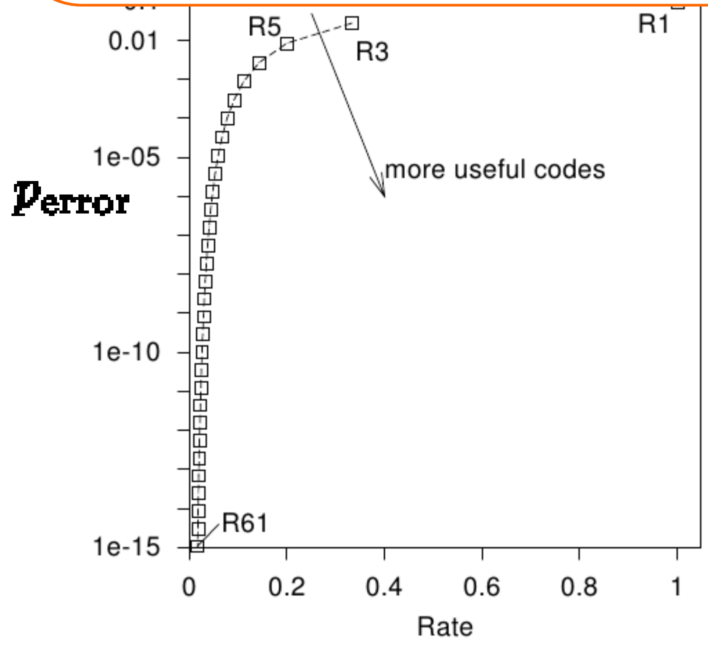


Illustration of error vs. number of bits Transmitted

- Let's assume we want to store our bits on a hard disk and want the error to be very small

Illustration of error vs. number of bits transmitted

- Let's assume we want to store our bits on a hard disk and want the error to be very small

  How many GB do we need to transmit in order to send 1 GB of actual data with error $p_{\mathrm{error}} = 10^{-15}$ using repetition coding for $f = 0.1$?

ILLUSTRATION OF ERROR VS. NUMBER OF BITS TRANSMITTED

- Let's assume we want to store our bits on a hard disk and want the error to be very small

  How many GB do we need to transmit in order to send 1 GB of actual data with error $p_{\text{error}} = 10^{-15}$ using repetition coding for $f = 0.1$?

- A

  bout 60!! (do the exercise, 1.3d!)

Illustration of error vs. number of bits transmitted

- Let's assume we want to store our bits on a hard disk and want the error to be very small

  How many GB do we need to transmit in order to send 1 GB of actual data with error $p_{\text{error}} = 10^{-15}$ using repetition coding for $f = 0.1$?

- A

  bout 60!! (do the exercise, 1.3d!)

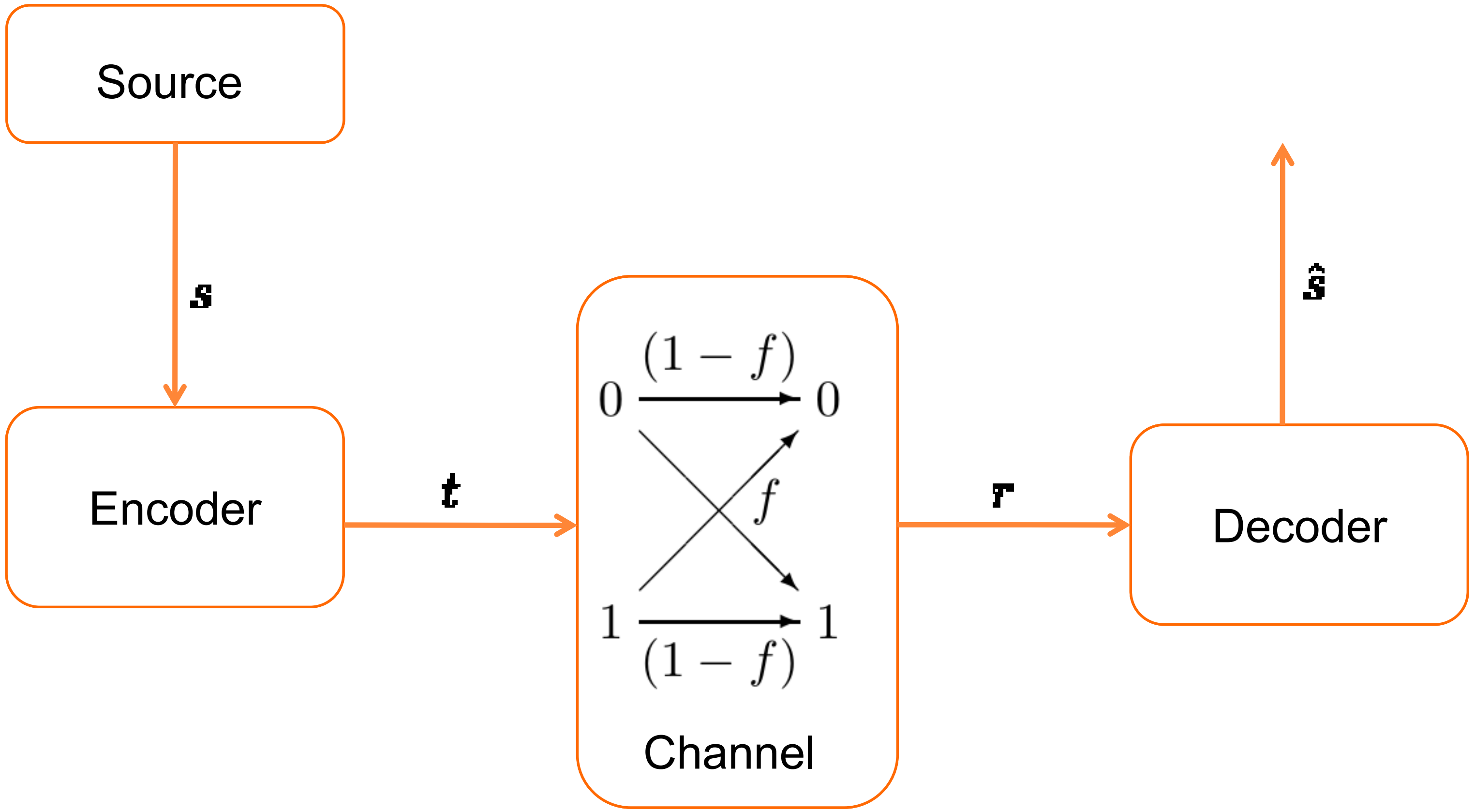
- … imagine the size of your phone if we were to store information like this ☺

ARE THERE BETTER CODES THAN REPETITION?

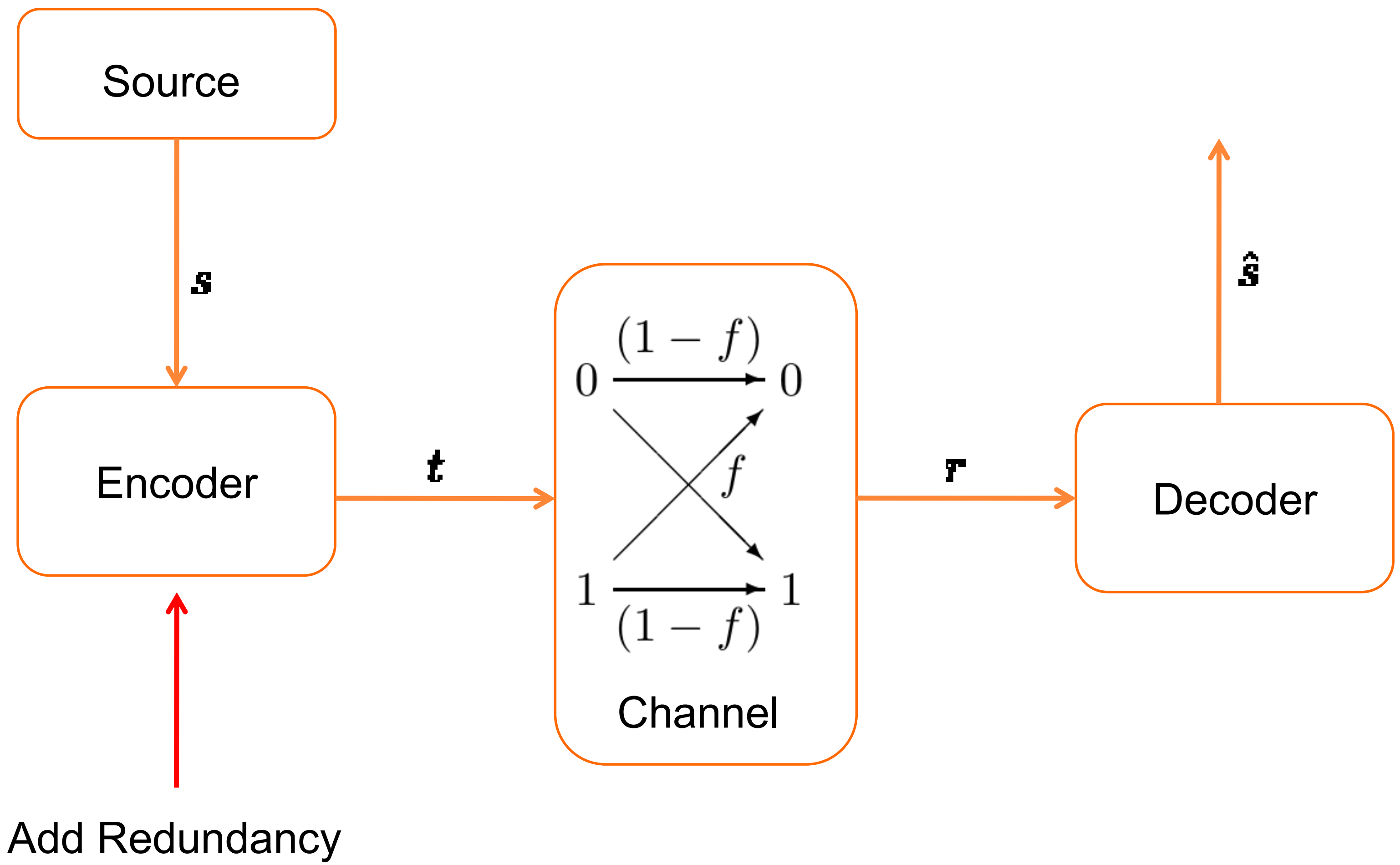
- What do you think?
- Let's remember our goal! We want to figure out:

What is the best way to store or transmit information?

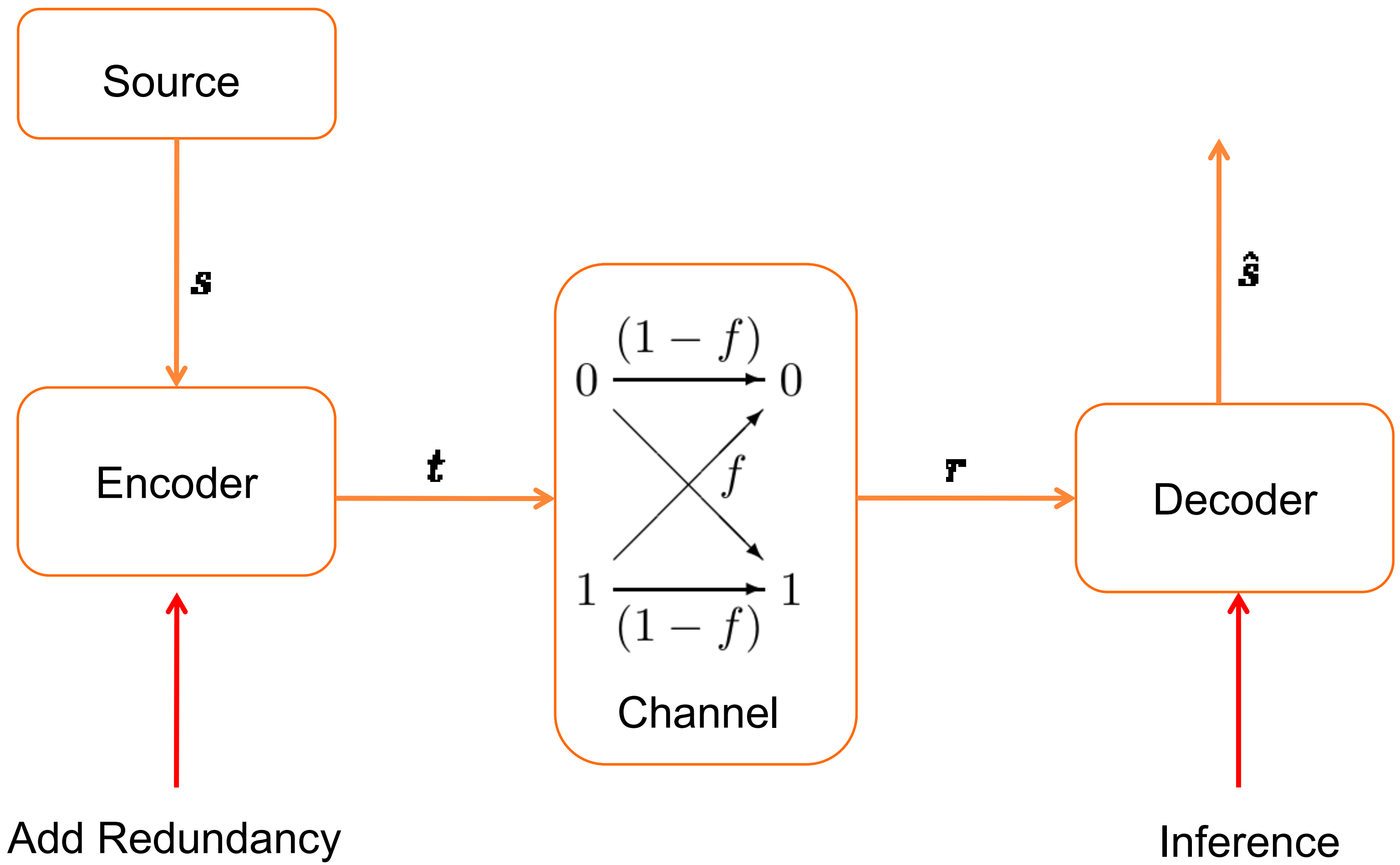
Sending information



SENDING INFORMATION



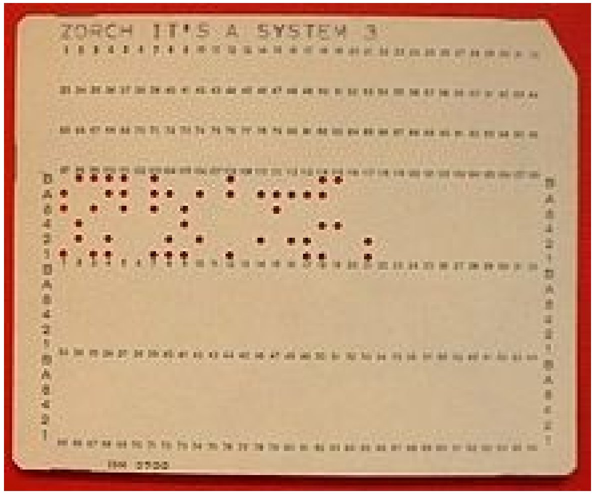
Sending information

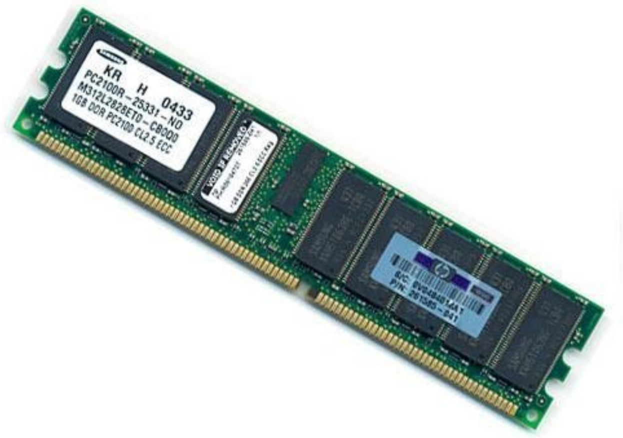


EXAMPLE: (7,4)-HAMMING CODE

- Invented by Richard Hamming

- Originally used to correct errors in punch cards but still in use today!





ENCODING IN THE (7,4)-HAMMING CODE

- Block code: each block of K=4 bits is encoded into N=7 bits
- Rate $R = \frac{4}{7}$

$$\mathbf{s} = s_1 s_2 s_3 s_4$$

$$\mathbf{t} = t_1 t_2 t_3 t_4 t_5 t_6 t_7$$

ENCODING IN THE (7,4)-HAMMING CODE

- Block code: each block of K=4 bits is encoded into N=7 bits
- Rate $R = \frac{4}{7}$

$\mathbf{s} = s_1 s_2 s_3 s_4$ 1 0 0 0

$\mathbf{t} = t_1 t_2 t_3 t_4 t_5 t_6 t_7$ 1 0 0 0 1 0 1

N-K Parity Bits

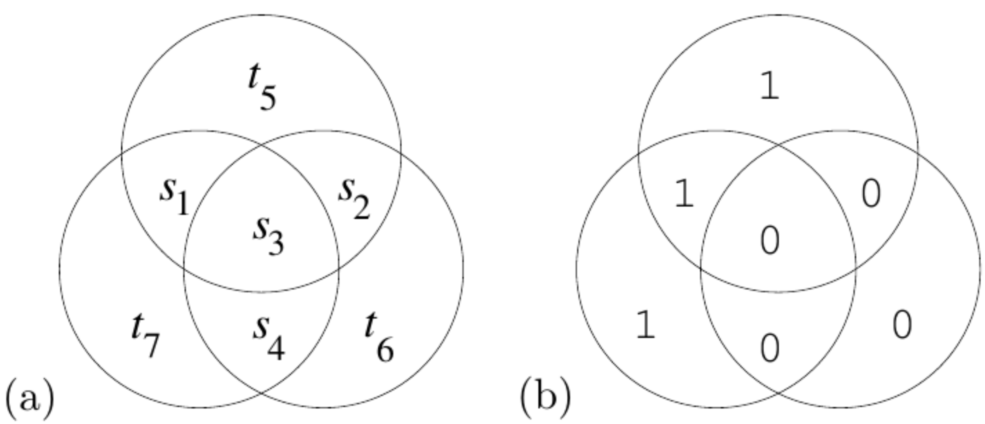
Encoding in the (7,4)-Hamming code

- Block code: each block of K=4 bits is encoded into N=7 bits
- Rate $R = \frac{4}{7}$

$\mathbf{s} = s_1 s_2 s_3 s_4$ 1 0 0 0

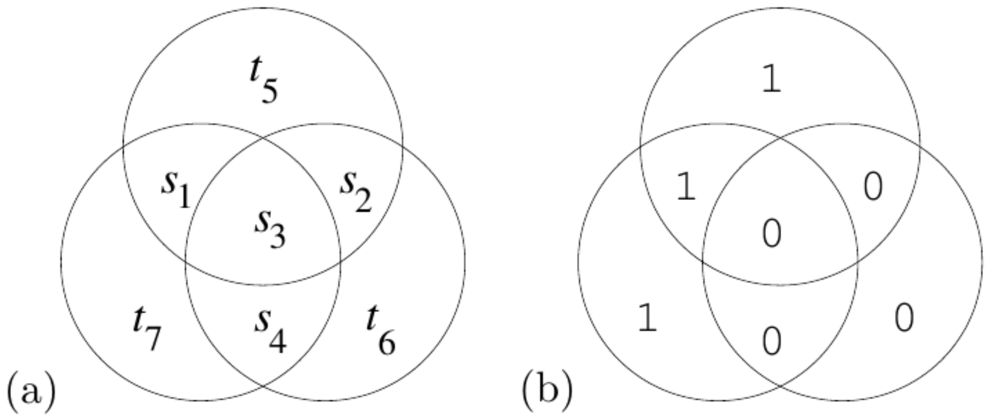
$\mathbf{t} = t_1 t_2 t_3 t_4 t_5 t_6 t_7$ 1 0 0 0 1 0 1

N-K Parity Bits



← Parity in each circle should be even (even # of 1's)

Encoding the (7,4)-Hamming code



Parity in each circle should be even (even # of 1's)

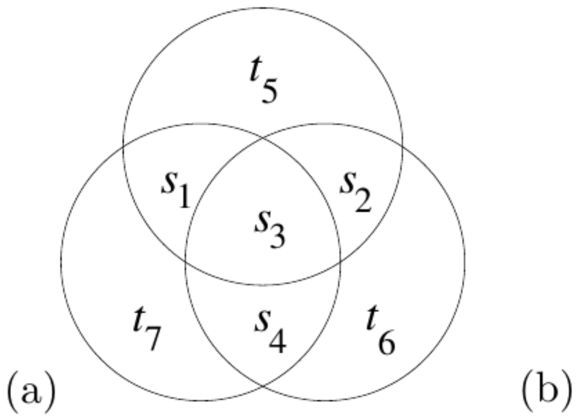
s	t	s	t	s	t	s	t
0000	0000000	0100	0100110	1000	1000101	1100	1100011
0001	0001011	0101	0101101	1001	1001110	1101	1101000
0010	0010111	0110	0110001	1010	1010010	1110	1110100
0011	0011100	0111	0111010	1011	1011001	1111	1111111

t is also called the *codeword*

As you can see every pair of code words differs in at least three bits!

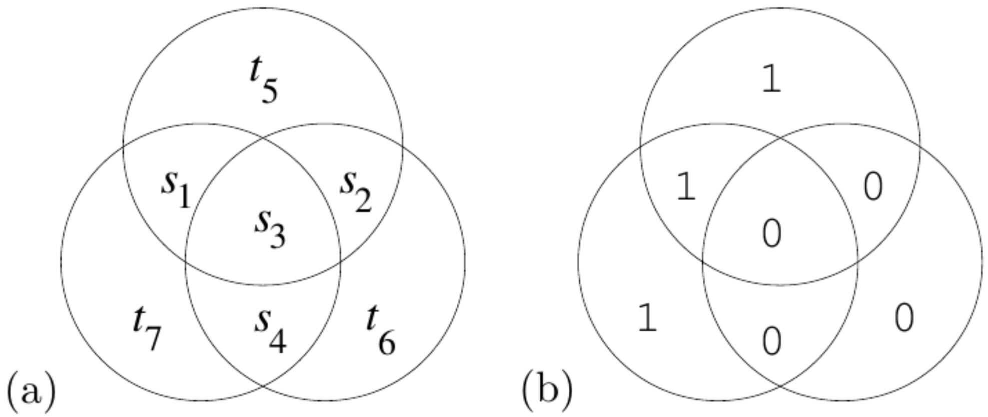
Distance d = 3 where $$d = \min_{\mathbf{t},\mathbf{t}'} \#\text{bits in which } \mathbf{t} \text{ and } \mathbf{t}' \text{ differ}$$

DECODING THE (7,4)-HAMMING CODE



Parity in each circle should be even (even # of 1's)

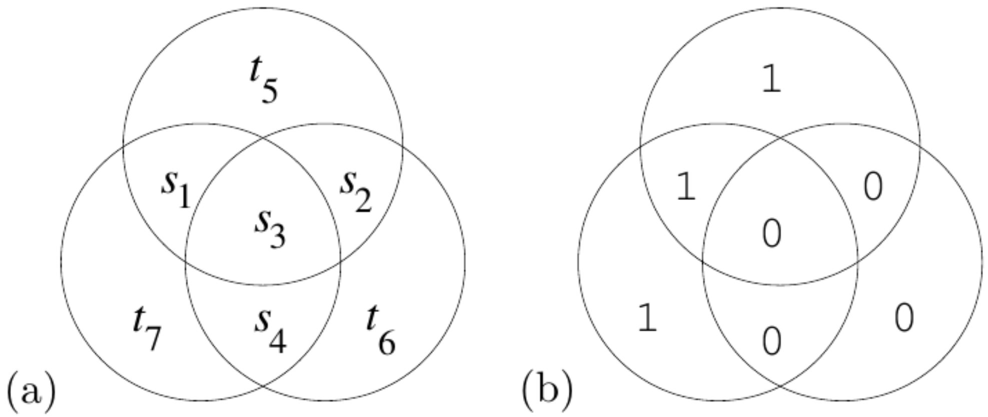
DECODING THE (7,4)-HAMMING CODE



Parity in each circle should be even (even # of 1's)

- If any of the circles have odd parity we know something went wrong!

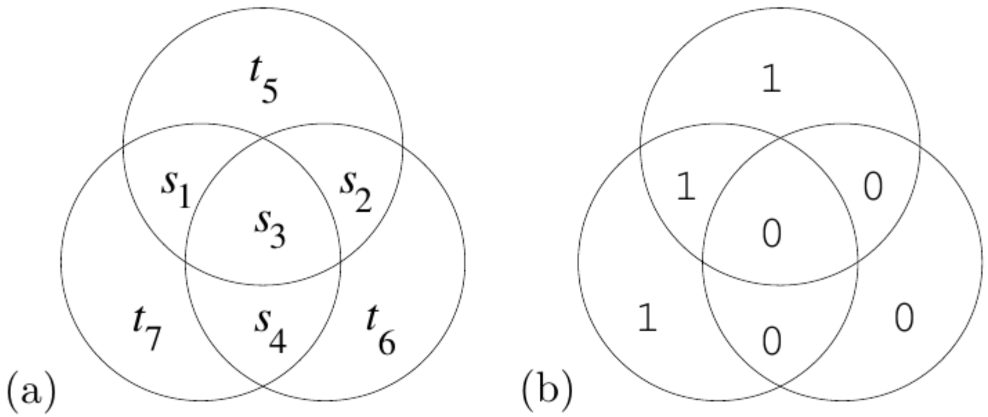
DECODING THE (7,4)-HAMMING CODE



Parity in each circle should be even (even # of 1's)

- If any of the circles have odd parity we know something went wrong!
- Remember: from the string **r** we received we want to decode to the *most likely* input **s**

DECODING THE (7,4)-HAMMING CODE

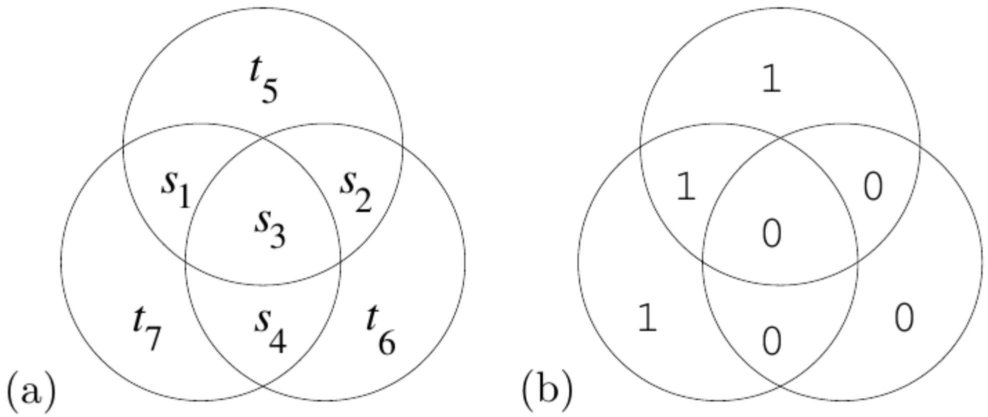


Parity in each circle should be even (even # of 1's)

- If any of the circles have odd parity we know something went wrong!
- Remember: from the string **r** we received we want to decode to the *most likely* input **s**

  A flip is *less likely* than no flip

DECODING THE (7,4)-HAMMING CODE



Parity in each circle should be even (even # of 1's)

- If any of the circles have odd parity we know something went wrong!

- Remember: from the string **r** we received we want to decode to the *most likely* input **s**

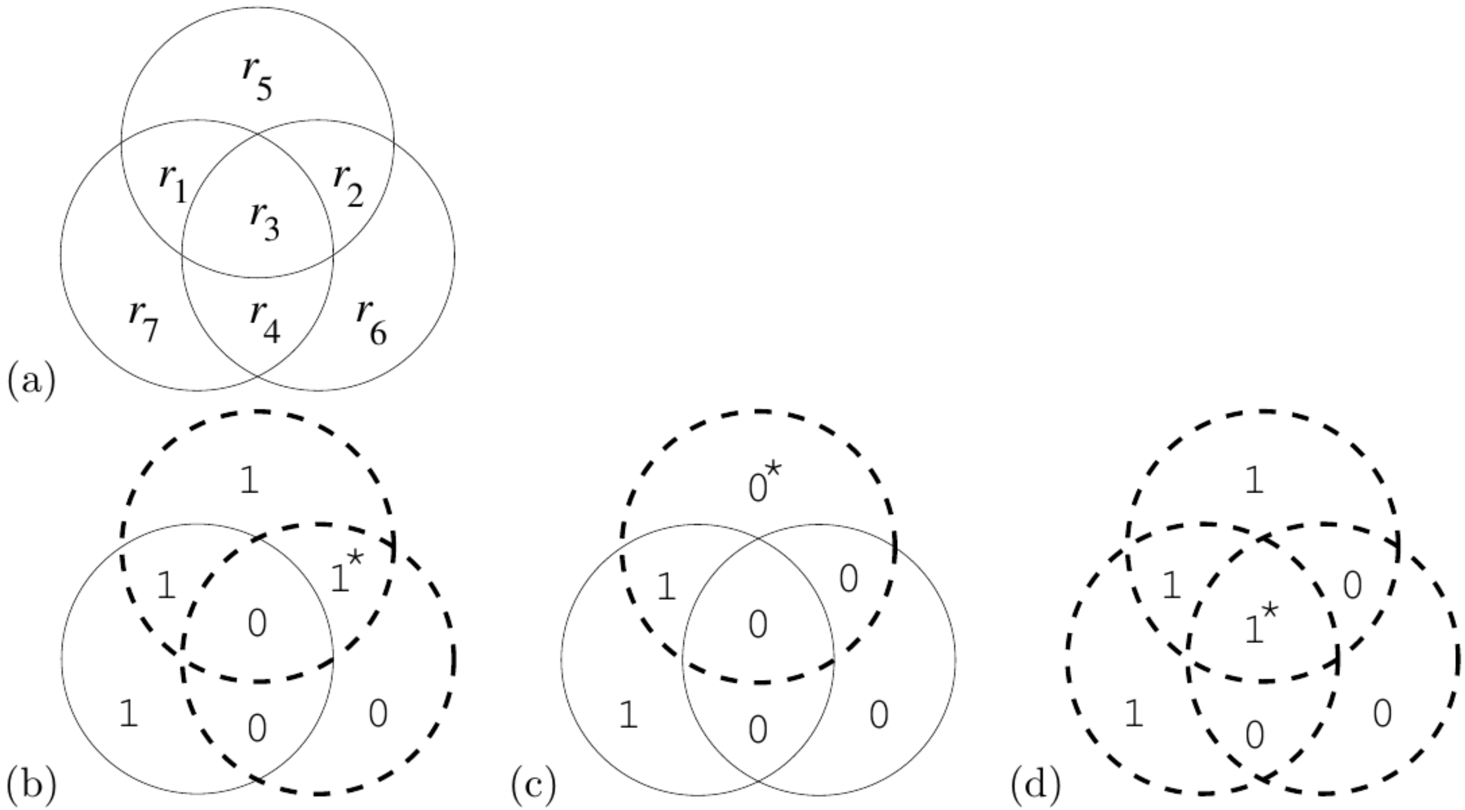
  A flip is *less likely* than no flip

- Find the smallest number of flips we can make ourselves to restore the cycles to correct parity

Decoding the (7,4)-Hamming code

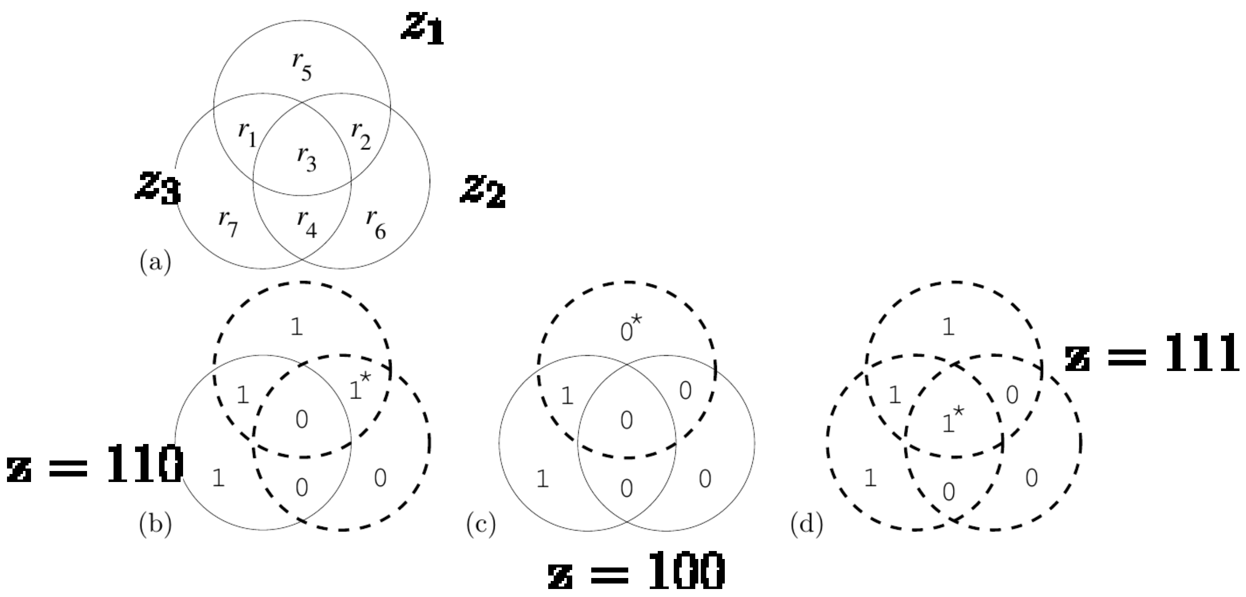
Want to find the least number of flips to restore all cycles!

Can we find a unique bit that lies
inside all unhappy cycles but **outside** all happy ones?



SYNDROME DECODING

- The syndrome $\mathbf{z} = z_1 z_2 z_3$ tells us which circle is violated



If $\mathbf{z} = 000$ it means that no error is detected!

Later we will see how this code can easily be implemented as a matrix and decoded without the visual picture.

How well are we doing now?

- Since we encode a block of 4 bits two different errors:
  - Probability of block error – any bit in the block is wrong

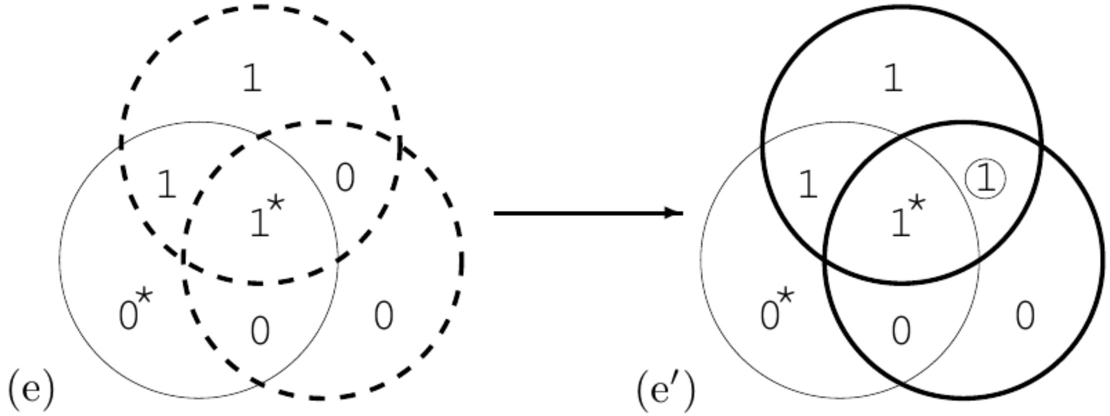
$$p_{\text{error Block}} = p(\hat{s} \neq s)$$

  - Probability of bit error – average probability that a bit is wrong

$$p_{\text{error Bit}} = \frac{1}{4}\sum_{k=1}^{K} p(\hat{s}_k \neq s_k)$$

How well are we doing now?

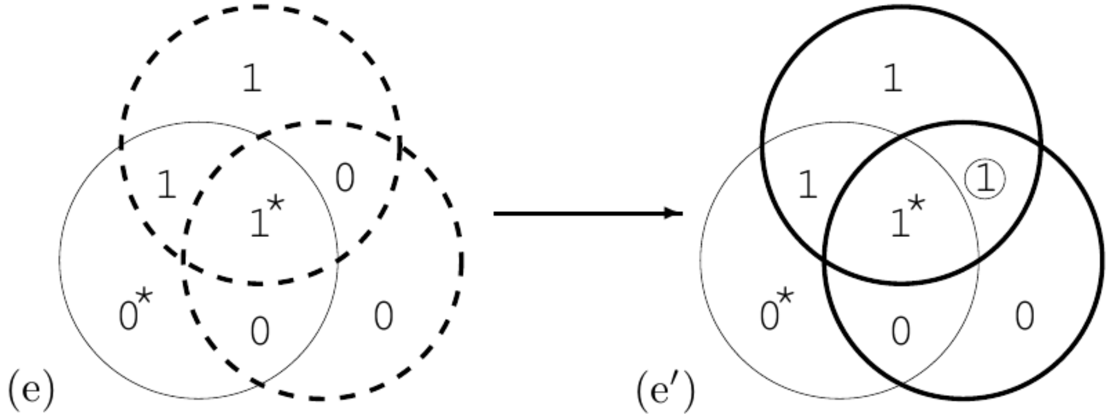
Block error occurs if more than one bit is flipped



$$p_{\text{error Block}} = p(\hat{s} \neq s)$$

How well are we doing now?

Block error occurs if more than one bit is flipped

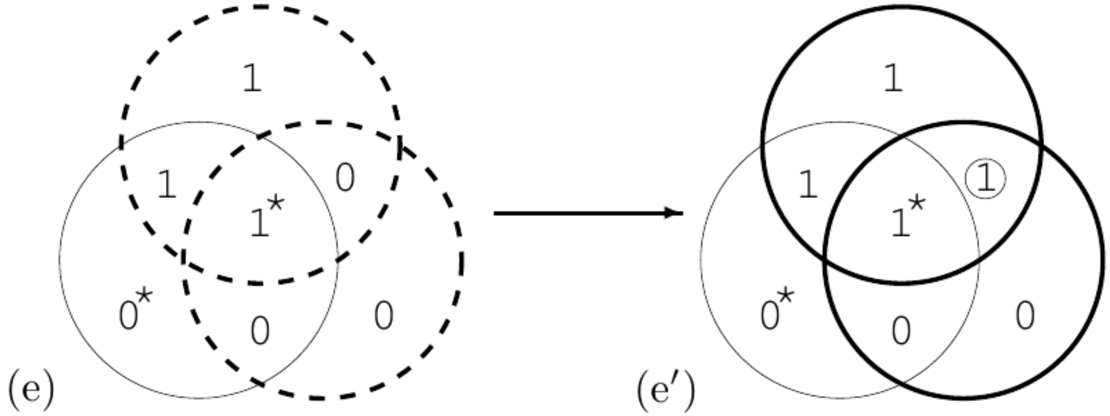


$$p_{\text{error Block}} = p(\hat{s} \neq s) = \sum_{r=2}^{7} \binom{7}{r} f^r (1-f)^{7-r} \simeq \binom{7}{2} f^2 = 21 f^2$$

Much worse than repetition code of 1 bit into 3 bits?

HOW WELL ARE WE DOING NOW?

Block error occurs if more than one bit is flipped



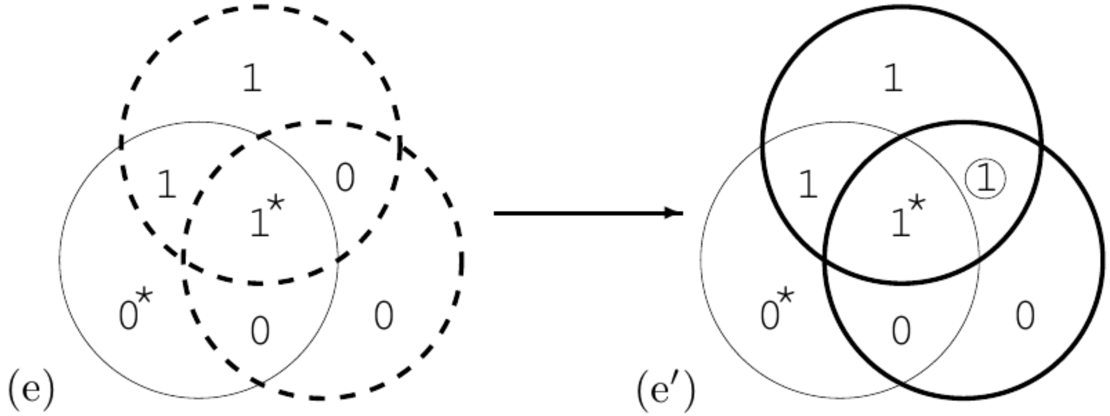
$$p_{\text{error Block}} = p(\hat{s} \neq s) = \sum_{r=2}^{7} \binom{7}{r} f^r (1-f)^{7-r} \simeq \binom{7}{2} f^2 = 21 f^2$$

Much worse than repetition code of 1 bit into 3 bits?

Bit error is more important than block error. Also big advantage:

How well are we doing now?

Block error occurs if more than one bit is flipped



$$p_{\text{error Block}} = p(\hat{s} \neq s) = \sum_{r=2}^{7} \binom{7}{r} f^r (1-f)^{7-r} \simeq \binom{7}{2} f^2 = 21 f^2$$

Much worse than repetition code of 1 bit into 3 bits?

Bit error is more important than block error. Also big advantage:

$$R = \frac{4}{7}$$

How about the bit error?

- Turns out about 7% of all bits will be wrong (probability of bit error)

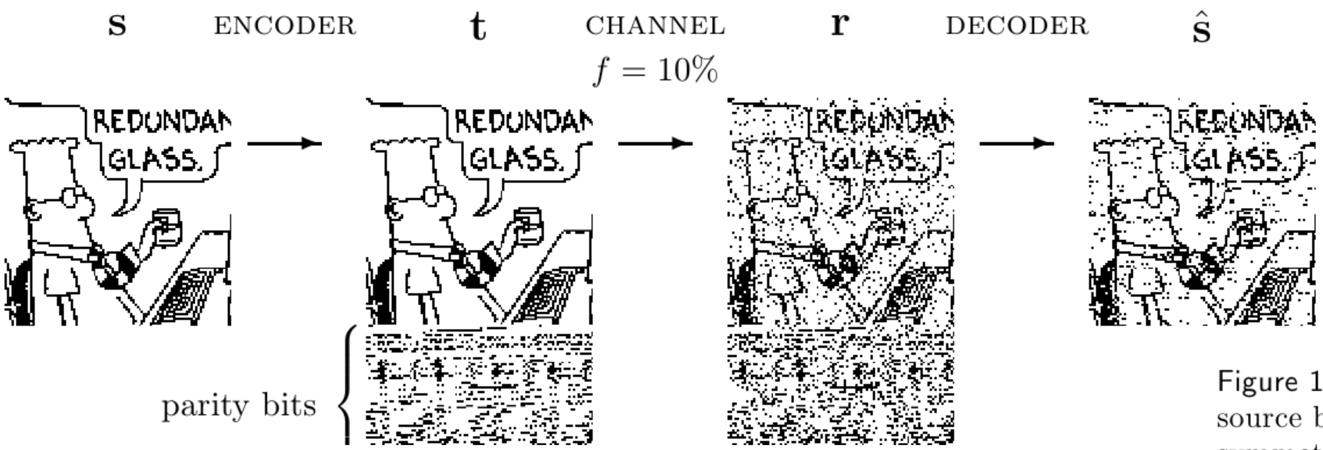
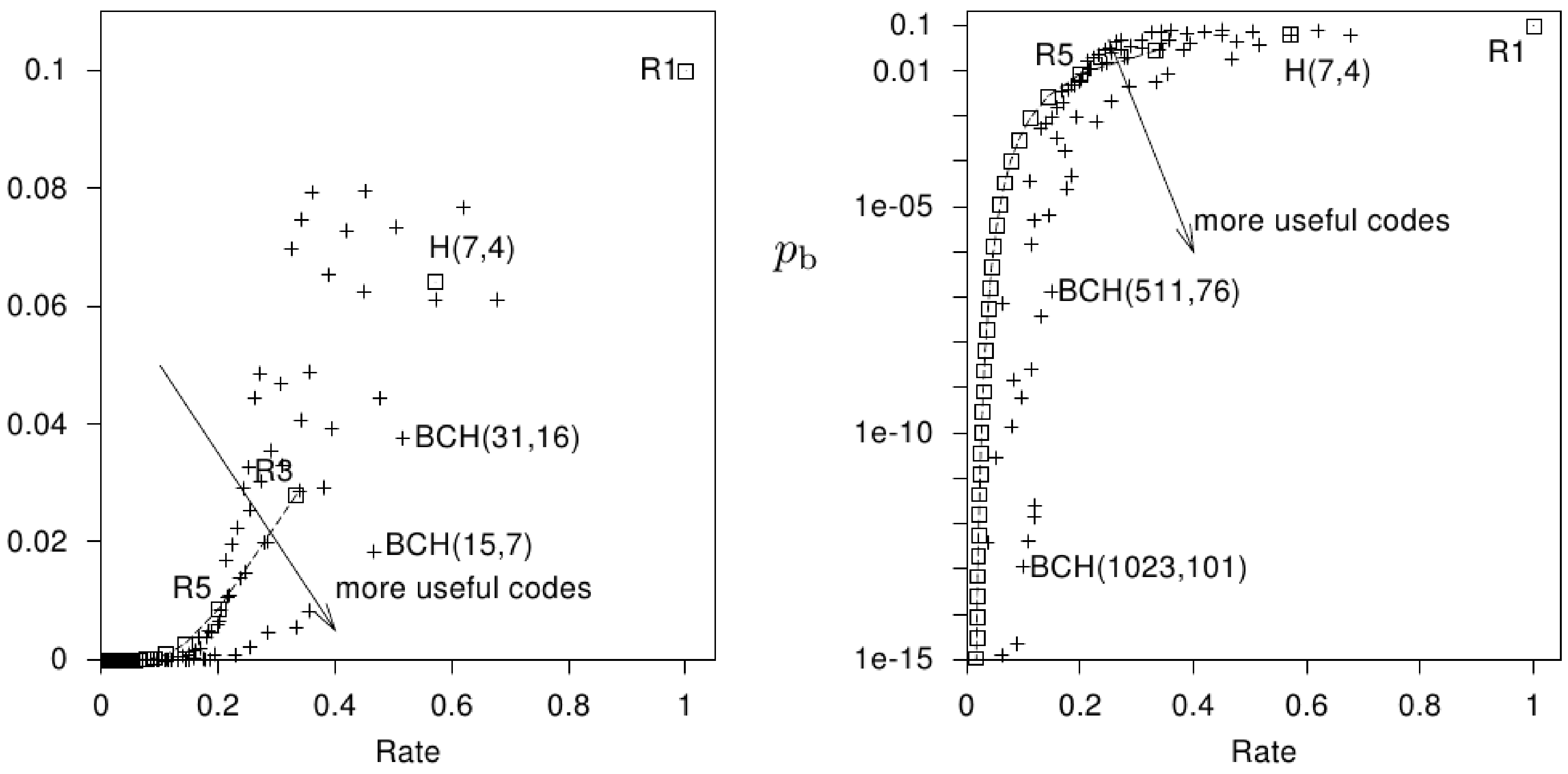
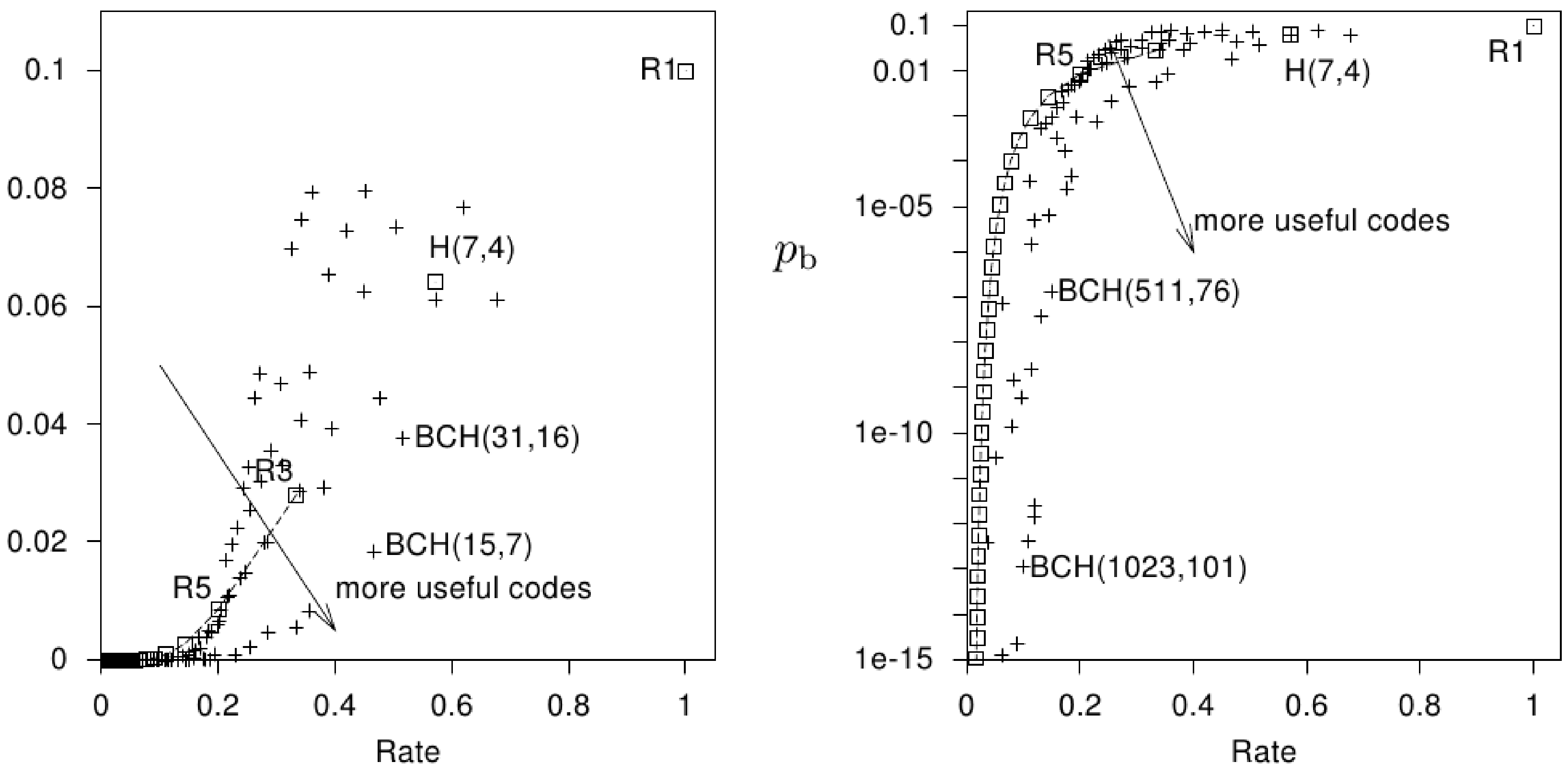


Figure 1.17. Transmitting 10 000 source bits over a binary symmetric channel with $f = 10\%$ using a (7,4) Hamming code. The probability of decoded bit error is about 7%.

WHERE DOES THE (7,4)-HAMMING CODE FIT

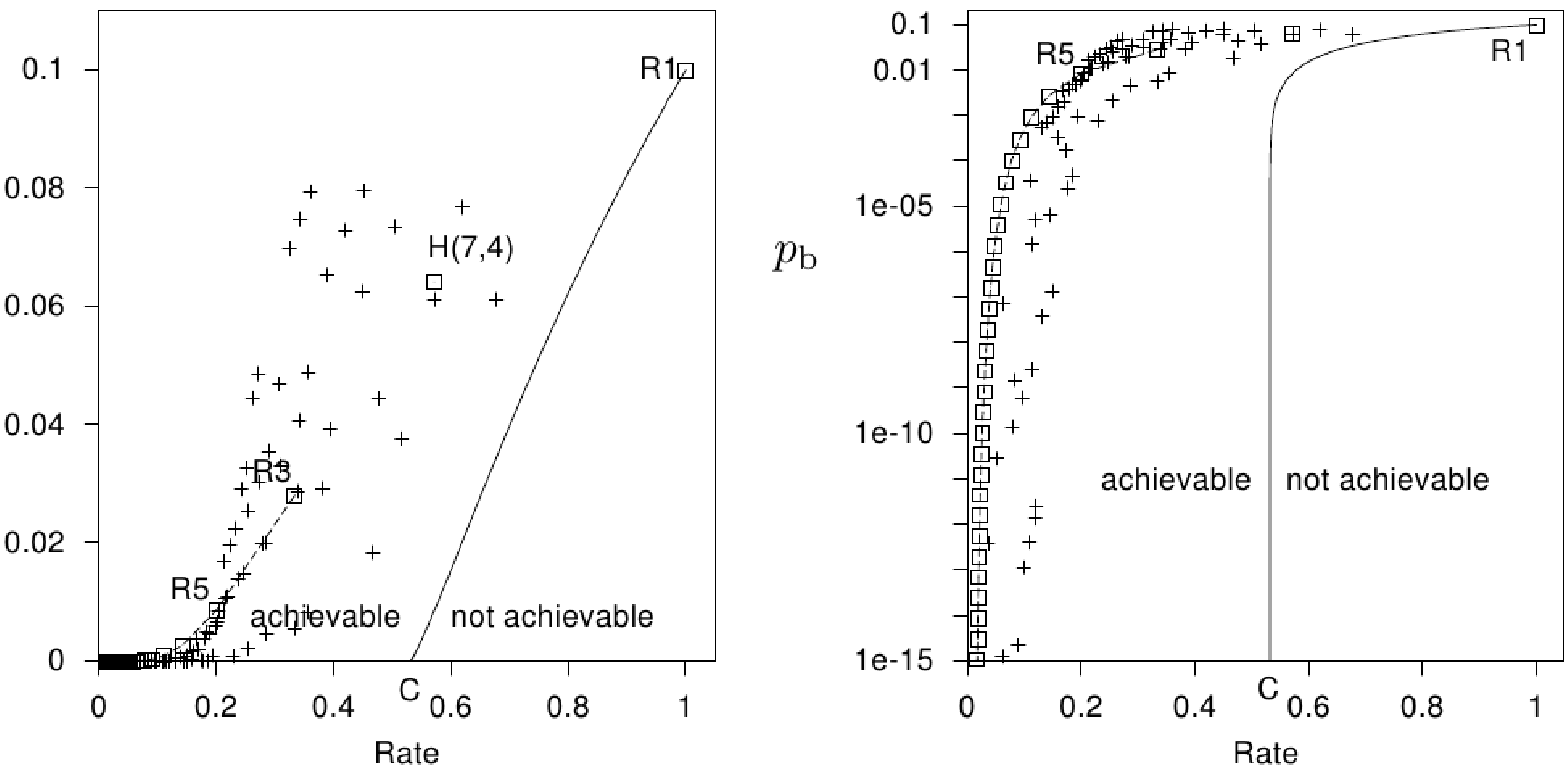


ARE THERE AT ALL BETTER CODES?



- Maybe to have the error arbitrarily close to 0 we must accept an arbitrarily small rate? (after all… no pain, no gain…)

SHANNON'S NOISY CHANNEL THEOREM



For every channel, there is a number C called the Capacity such that we can find a code with arbitrarily small amount of error and R = C!

(large |s|=n)

Let's go back to our Mars transmission example

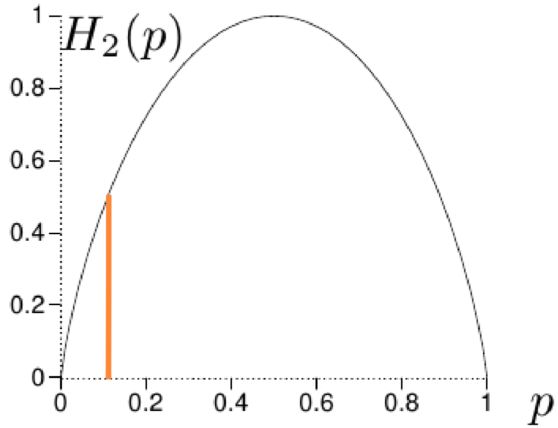
- Binary symmetric channel (BSC) with noise $f = 0.1$

Let's go back to our Mars transmission example

- Binary symmetric channel (BSC) with noise $f = 0.1$
- Repetition code for error $p_{err} = 10^{-15}$ need to send 60GB for every 1GB of actual data!

LET'S GO BACK TO OUR MARS TRANSMISSION EXAMPLE

- Binary symmetric channel (BSC) with noise $f = 0.1$
- Repetition code for error $p_{err} = 10^{-15}$ need to send 60GB for every 1GB of actual data!
- Shannon: capacity of BSC is $C = 1 - H_2(f)$



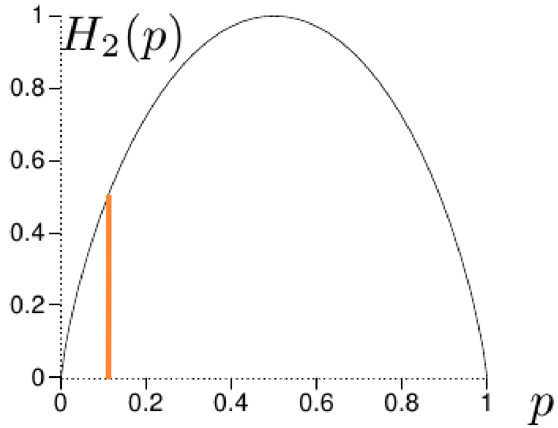
LET'S GO BACK TO OUR MARS TRANSMISSION EXAMPLE

- Binary symmetric channel (BSC) with noise $f = 0.1$
- Repetition code for error $p_{err} = 10^{-15}$ need to send 60GB for every 1GB of actual data!
- Shannon: capacity of BSC is $C = 1 - H_2(f)$

Here $C \approx 0.47$

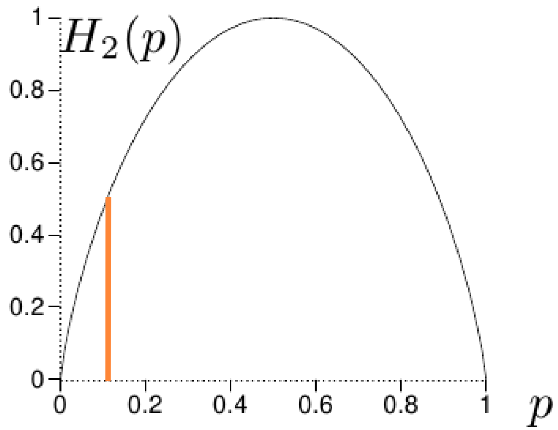
$$R = \frac{1\text{GB}}{N} = 0.47$$

$N \approx 2.2\text{GB}$ is enough!



SHANNON ENTROPY

$$C = 1 - H_2(f)$$

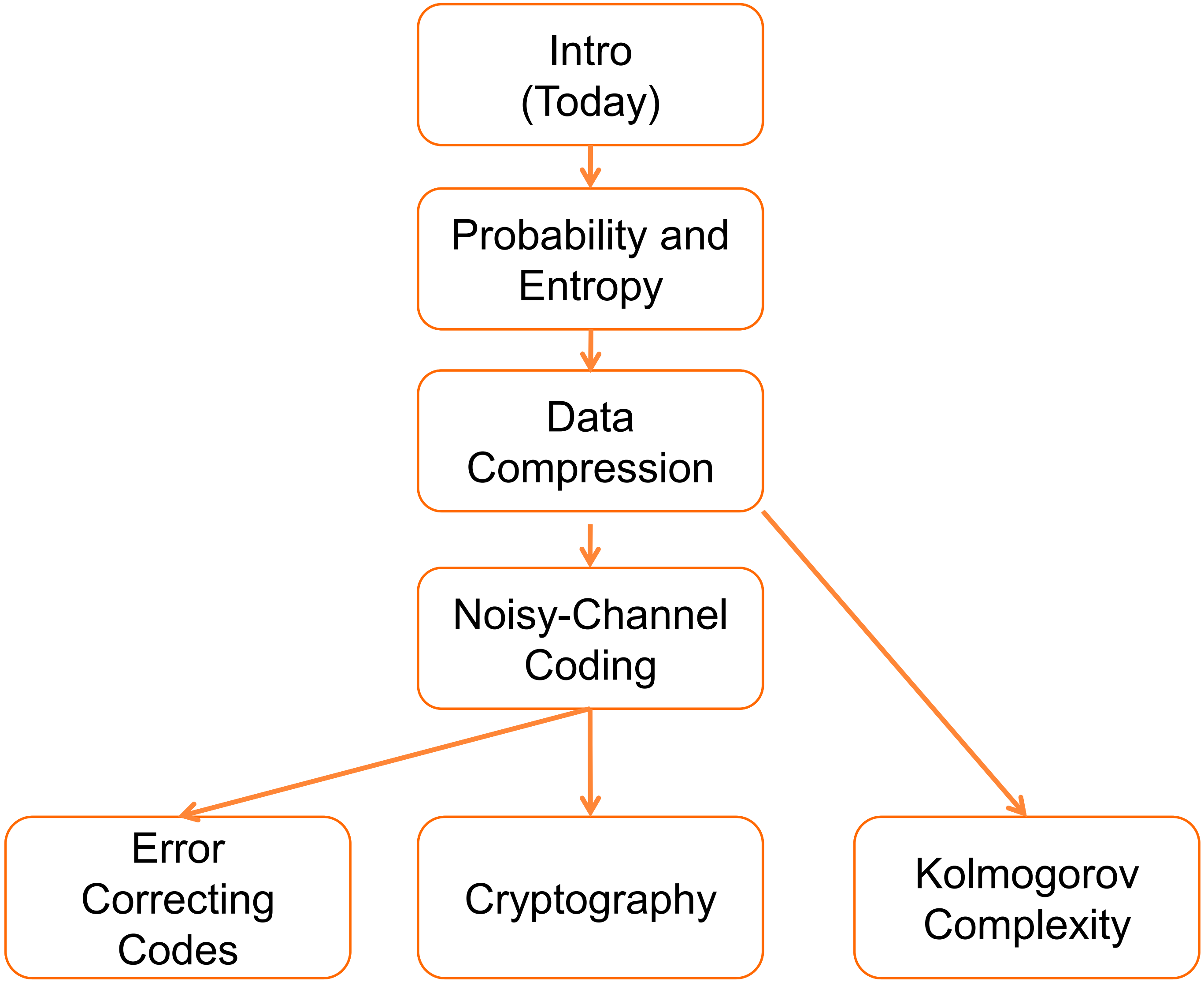


- Binary entropy $$H_2(f) = -f \log_2 f - (1-f)\log_2(1-f)$$
- Shannon entropy $$H(p) = -\sum_x p_x \log_2 p_x$$

What have we learned about information theory?

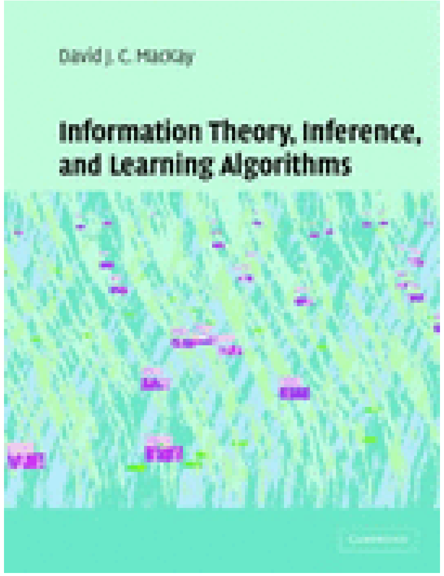
- Encoding and decoding information
- Repetition code
- (7,4)-Hamming code
  - Encodes 4 bits into 7 bits
  - Can detect and correct any error of a single bit!
- Shannon's noisy channel coding theorem
  - Information can be communicated over a noisy-channel at a non-zero rate with arbitrarily small error probability!

Where do we go from here?



Reading for this lecture

- Chapter 1 in the book



Information Theory, Inference and Learning Algorithms
by David J. C. MacKay
Cambridge University Press, 2003

- Homework due by the beginning of next lecture, hand out in person at next lecture or upload to IVLE Workbin (please label upload with your name and assignment number)