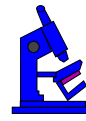




Chapter 4



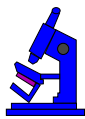
Lecture 4 - Preliminaries



Chocolate fish people



- ✓ Andreas Schuth
- ✓ Chong Jun Yong
- ✓ Ashley Ng *
- ✓ Wu Yongzheng *
- ✓ Zhang Huaixing *
- ✓ Terence Sangeet



The extended Euclidean algorithm

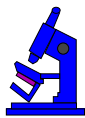


$$x * a + y * p = \gcd(x, y)$$

Now - if p is a prime, then $\gcd(x, y) = 1$, and so

$$x * a + y * p = 1$$

WRONG!



The extended Euclidean algorithm



$$x * a + y * p = \gcd(a, p)$$

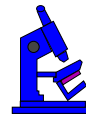
Now - if p is a prime, then $\gcd(a, p) = 1$, and so

$$x * a + y * p = 1$$

RIGHT!



Last session



- Math preliminaries
 - Fermat's little theorem
 - Euler



This session



- Physical preliminaries
- Entropy



This session



- Physical preliminaries
- Entropy



Preliminaries - physical

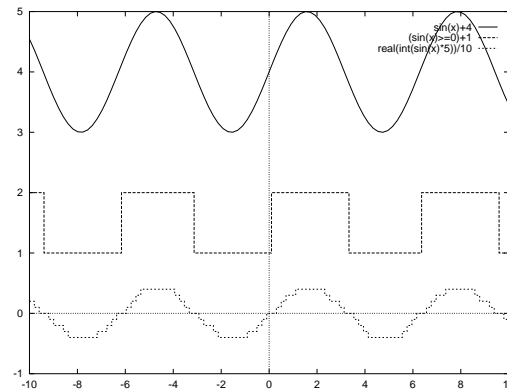


Consider:

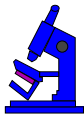
- Is the data **analog** or **digital**?
- What **limits** are placed on it?
- How is it to be **transmitted**?
- How can you be sure that it is **correct/accurate**?



Analog and digital



The plot is **amplitude versus time**.



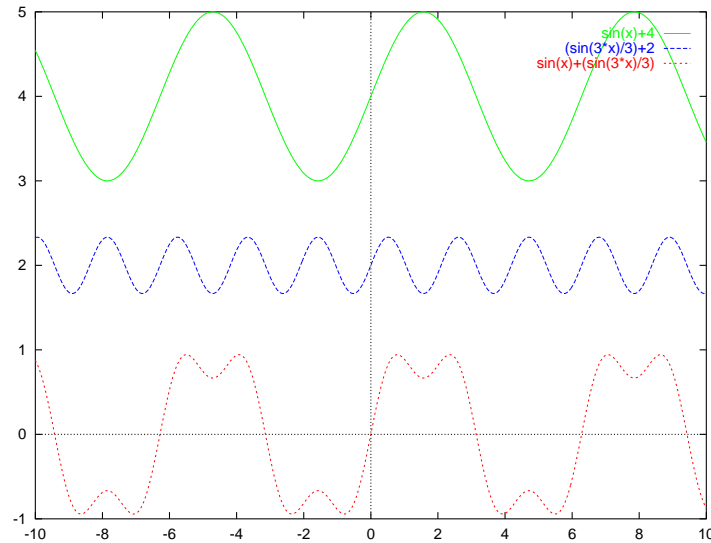
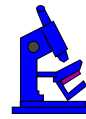
Analog and digital



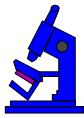
- ✓ Repetition rate (if it repeats) is called the *frequency*, and is measured in *Hertz*
- ✓ The *peak to peak signal level* is called the *amplitude*.
- ✓ The *simplest* analog signal is called the *sine* wave.
- ✓ By mixing we may create *any desired periodic waveform*.



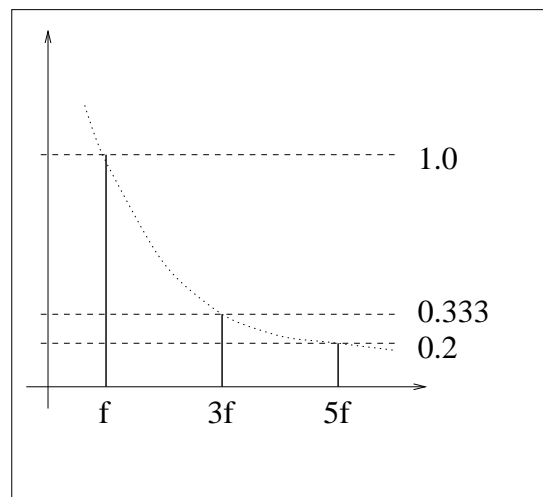
Analog and digital



The plot is **amplitude versus time**. (Time domain)



Analog and digital



The plot is **frequency versus time**. (Frequency domain).



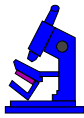
Analog and digital



If we were to continue in the same progression, the resultant waveform would be a **square wave**:

$$\sum_{n=1}^{\infty} \frac{1}{n} \sin(2\pi n f) \text{ (for odd } n) \Rightarrow \text{square wave, frequency } f$$

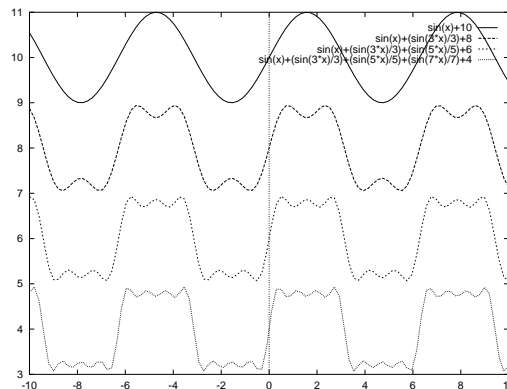
This representation method is known as *Fourier Analysis* after Jean-Baptiste Fourier.



Fourier analysis



$$\frac{4}{\pi} (\sin(2\pi ft) + \frac{1}{3} \sin(6\pi ft) + \frac{1}{5} \sin(10\pi ft) + \frac{1}{7} \sin(14\pi ft) + \dots)$$





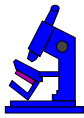
Fourier analysis



Transformation between equivalent time domain and frequency domain representations.

A piecewise continuously differentiable periodic function in the time domain may be transformed to a discrete aperiodic function in the frequency domain.

$$\begin{aligned} \text{smooth, repeating} &\leftrightarrow \text{pointy, notrepeating} \\ f(t) &\leftrightarrow F(\omega) \end{aligned}$$



Fourier analysis



Time domain		Frequency domain	Description
Continuous, periodic	\rightleftharpoons	Discrete, aperiodic	Fourier series
Continuous, aperiodic	\rightleftharpoons	Continuous, aperiodic	Fourier transform
Discrete, periodic	\rightleftharpoons	Discrete, periodic	Discrete Fourier series
Discrete, aperiodic	\rightleftharpoons	Continuous, periodic	Discrete Fourier transform



Accuracy

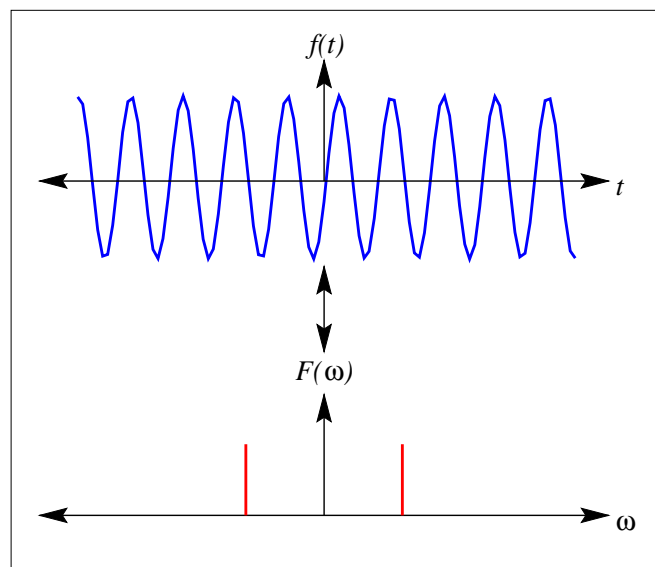


Relationship between the **bandwidth** of a channel, and *how accurate* a signal is.

Another way of stating this is to point out that the higher frequency components are important - they are needed to re-create the original signal faithfully. If we had two 1,000Hz signals, one a triangle, one a square wave - if they were both passed through the 1,000Hz bandwidth limited channel above, they would look identical (a sine wave).

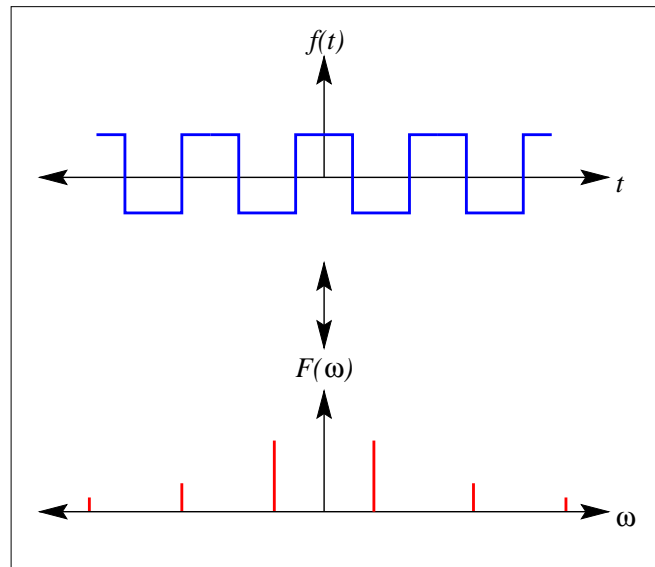


Example transforms

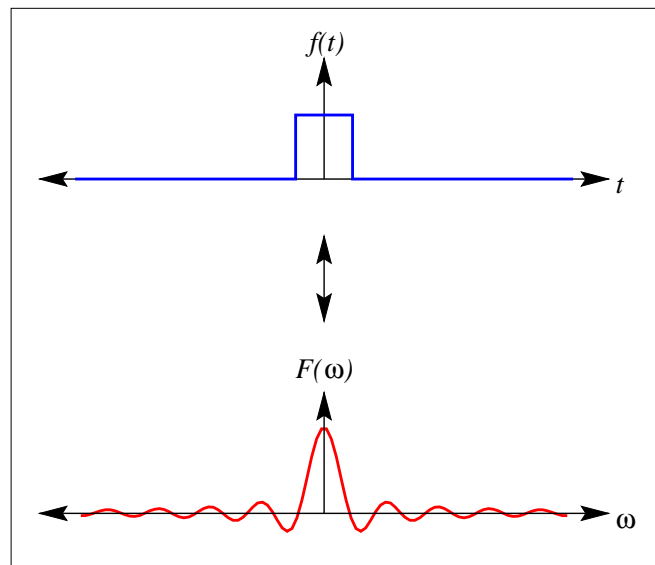




Example transforms

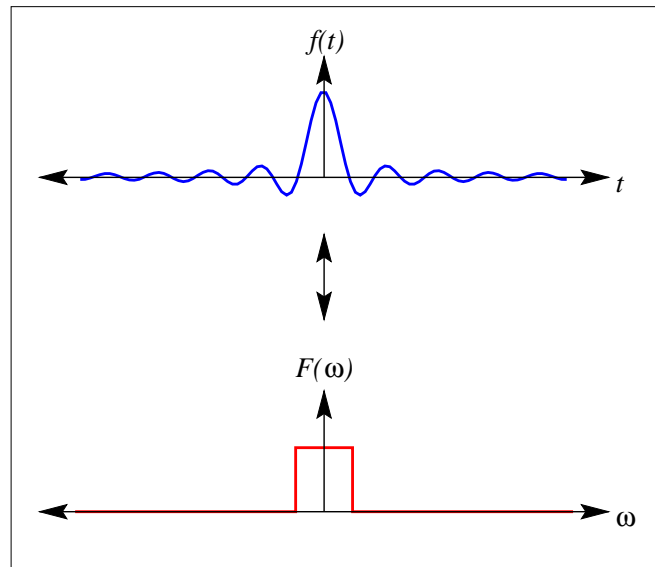


Example transforms





Example transforms



Convolution



The Fourier transform of the convolution $f(t) \star g(t)$ is the product of the Fourier transforms of the functions $F(\omega)$ and $G(\omega)$, *and vice versa*.

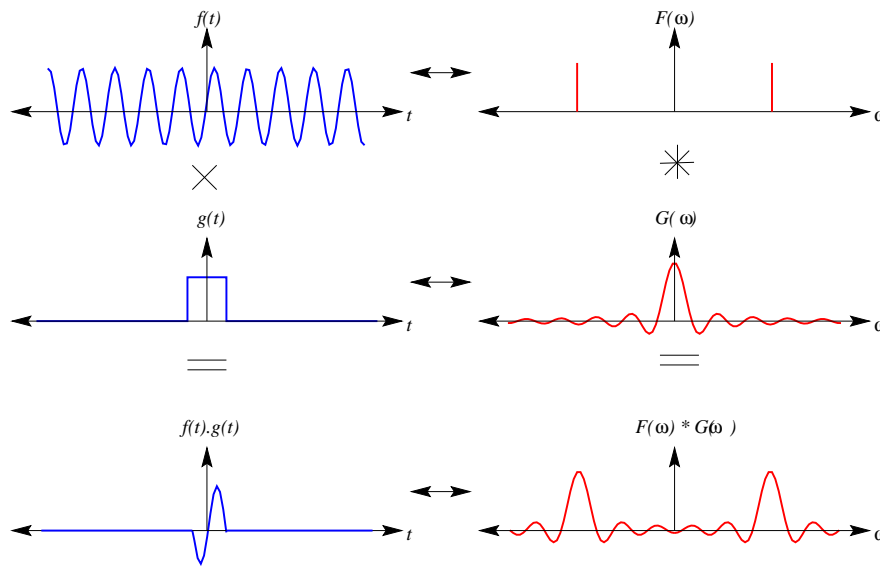
$$f(t) \star g(t) \leftrightarrow F(\omega) \times G(\omega)$$

$$f(t) \times g(t) \leftrightarrow F(\omega) \star G(\omega)$$

We can use convolution to easily predict the functions that result from complex signal filtering or sampling.



Convolution



Modulation



A **baseband** signal is one in which the data is **directly** converted to a signal and transmitted. When the signal is imposed on **another signal**, the process is called **modulation**.

We may **modulate** for several reasons:

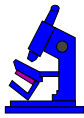
- The media may not support the baseband signal
- We may wish to use a single transmission medium to transport many signals



Modulation methods



- Frequency modulation - frequency shift keying (FSK)
- Amplitude modulation
- Phase modulation - phase shift keying (PSK)
- Combinations of the above (QAM)



Baseband digital encoding

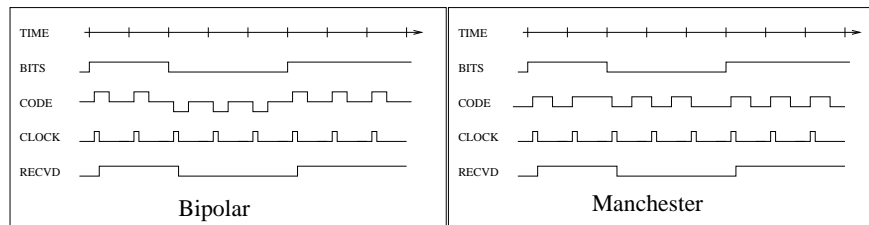


The simplest encoding scheme is just to use a *low level* for a *zero* bit, and a *high level* for a *one* bit. As long as both ends of a channel are synchronized in some manner, we can transfer data.

On the other hand, if the ends of the channel are not synchronized we might use a simple encoding scheme, such as *Bipolar* or *Manchester* encoding, to transfer synchronizing (clock) information on the same channel.



Baseband digital encoding



- ✓ In **Bipolar** encoding, a 1 is transmitted with a positive pulse, a 0 with a negative pulse. Sometimes called *return to zero* encoding.
- ✓ In **Manchester** encoding, there is a transition in the center of each bit cell.



Summary



- ✓ Data commonly transferred digitally
- ✓ **Trade-off** between bandwidth, accuracy of **any** signal



Information theory



The term **information** is commonly understood. Consider the following two sentences:

1. The sun will rise tomorrow.
2. The Fiji rugby team will win against the All Blacks (New Zealand rugby team) the next time they play.

Question: Which sentence contains the most information?



Information theory



✗ Temperature today is OK, Temperature today is OK, Temperature today is OK, Temperature today is OK, Temperature today is OK, Temperature today is OK, Temperature today is OK, Temperature today is OK, ...

... total information here is close to zero!

?

More information means less predictable
Less information means more predictable



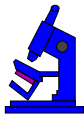
Information theory



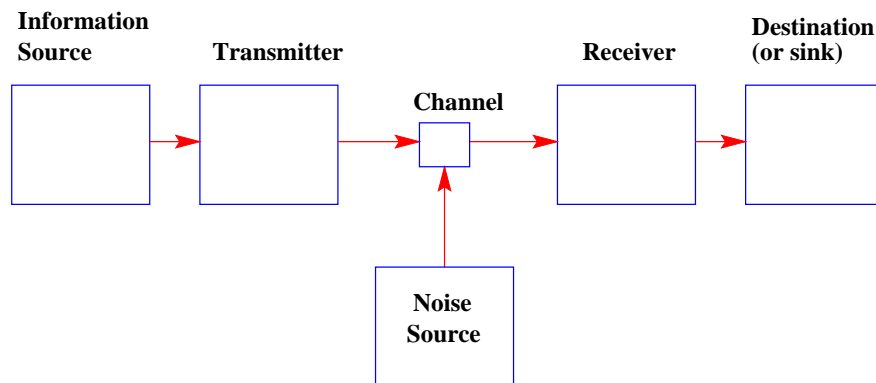
Nyquist (1924) and Hartley (1928) laid the foundations:

- ✓ Hartley showed that the information content is proportional to the *logarithm* of the number of possible messages. Integers between 1 and n need $\log_2 n$ bits.
- ✓ Shannon developed a mathematical treatment of communication and information in an important paper at

<http://cm.bell-labs.com/cm/ms/what/shannonday/paper.html>



Information theory model



The relevance of Shannon to *secrecy* is in another important paper at

<http://www.cs.ucla.edu/~jkong/research/security/shannon.html>



Entropy



In our communication model, the **units** of transmission are called **messages**, constructed from an alphabet of (say) n symbols $x \in \{x_1, \dots, x_n\}$ each with a **probability** of transmission P_x .

We associate with each symbol x a quantity H_x which is a **measure** of the **information** associated with that symbol.

$$H_x = P_x \log_2 \frac{1}{P_x}$$



Entropy



$$H_x = P_x \log_2 \frac{1}{P_x}$$

If the probability of occurrence of each symbol is the same, we can derive Hartley's result, that the average amount of information transmitted in a single symbol (the **source entropy**) is

$$H(X) = \log_2 n$$

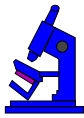
where X is a label referring to each of the source symbols x_1, \dots, x_n .



Entropy units



Our **units** for entropy can be *bits/second* or *bits/symbol*, and we also sometimes use unit-less *relative* entropy measures (relative to the entropy of the system if all symbols were equally likely).



Entropy - same probability



Symbols	Entropy of each symbol	Bits needed
2	$H_x = \frac{1}{2} \log_2 2 = \frac{1}{2}$	$2 * \frac{1}{2} = 1$
4	$H_x = \frac{1}{4} \log_2 4 = \frac{1}{2}$	$4 * \frac{1}{2} = 2$
8	$H_x = \frac{1}{8} \log_2 8 = \frac{3}{8}$	$8 * \frac{3}{8} = 3$
16	$H_x = \frac{1}{16} \log_2 16 = \frac{4}{16}$	$16 * \frac{4}{16} = 4$
21	$H_x = \frac{1}{21} \log_2 21 = \frac{4.39}{21}$	$21 * \frac{4.39}{21} = 4.39$



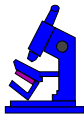
Entropy - different probability



However, if the **probability** of occurrence of each symbol is **not the same**, we derive the following result, that the source *entropy* is

$$H(X) = \sum_{i=1}^n P_{x_i} \log_2 \frac{1}{P_{x_i}}$$

Shannon's paper shows that H determines the channel capacity required to transmit the desired information with the *most* efficient coding scheme.



Entropy - different probability



If we had a source emitting two symbols, 0 and 1, with probabilities of 1 and 0, then the entropy of the source is

$$\begin{aligned} H(X) &= \sum_{i=1}^n P_{x_i} \log_2 \frac{1}{P_{x_i}} \\ &= \log_2 1 + 0 * \log_2 0 \\ &= 0 \text{ bits/symbol} \end{aligned}$$

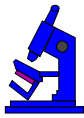


Entropy - different probability



If we were transmitting a sequence of letters A, B, C, D, E and F with probabilities $\frac{1}{2}, \frac{1}{4}, \frac{1}{16}, \frac{1}{16}, \frac{1}{16}$ and $\frac{1}{16}$, the entropy for the system is

$$\begin{aligned} H(X) &= \frac{1}{2} \log_2 2 + \frac{1}{4} \log_2 4 + \frac{4}{16} \log_2 16 \\ &= 0.5 + 0.5 + 1.0 \\ &= 2 \text{ bits/symbol} \end{aligned}$$



Encoding the letters



A fixed size 3-bit code, and then a more complex code:

Symbol	3-bit code	Complex code
A	000	0
B	001	10
C	010	1100
D	011	1101
E	100	1110
F	101	1111

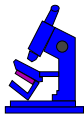


Analysis of encoding



The average **length** of the binary digits needed to encode a typical sequence of symbols **using the 3-bit code** is

$$\begin{aligned}L(X) &= \sum_{i=1}^n P_{x_i} \bullet \text{sizeof}(x_i) \\&= \frac{1}{2} * 3 + \frac{1}{4} * 3 + \frac{4}{16} * 3 \\&= 1.5 + 0.75 + 0.75 \\&= 3 \text{ bits/symbol}\end{aligned}$$



Analysis of encoding



The average **length** of the binary digits needed to encode a typical sequence of symbols **using the complex encoding** is

$$\begin{aligned}L(X) &= \sum_{i=1}^n P_{x_i} \bullet \text{sizeof}(x_i) \\&= \frac{1}{2} * 1 + \frac{1}{4} * 2 + \frac{4}{16} * 4 \\&= 0.5 + 0.5 + 1.0 \\&= 2 \text{ bits/symbol}\end{aligned}$$

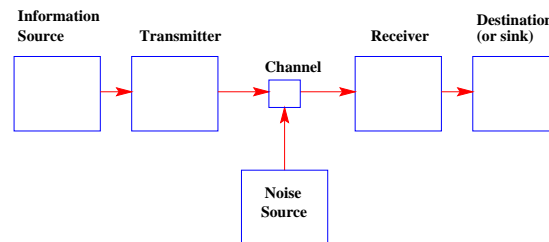
i.e. it is more efficient, averaging only 2 bits for each symbol transmitted.



Entropy and transmission rate



If our source was transmitting 0 and 1 bits with equal probability, but the received data was corrupted 50% of the time, we might reason that our rate $r(X)$ of information transmission was 0.5, because half of our data is getting through correctly.



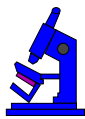
Entropy and transmission rate



However, a better argument is to consider the difference between the entropy of the source and the conditional entropy of the received data:

$$r(X) = H(X) - H(X | y)$$

where $H(X | y)$ is the *conditional* entropy of the received data.



Entropy and transmission rate



$$\begin{aligned}H(X | y) &= 0.5 * \log_2 2 + 0.5 * \log_2 2 \\ &= 1\end{aligned}$$

$$\text{and } H(X) = 1 \quad (\text{shown before})$$

$$\begin{aligned}\text{so } r(X) &= H(X) - H(X | y) \\ &= 0 \text{ bits/symbol}\end{aligned}$$

This is a much better measure of the amount of information transmitted.



Redundancy



The ratio of the entropy of a source $H(X)$ to what it would be if the symbols had equal probabilities $H'(X)$, is called the *relative entropy*. We use the notation $H_r(X)$, and

$$H_r(X) = \frac{H(X)}{H'(X)}$$

The *redundancy* of the source is $1 - H_r(X)$

$$R(X) = 1 - H_r(X)$$

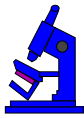


Redundancy



- ✓ If we look at English text a symbol at a time¹, the **redundancy** is about **0.7**.
- ✓ This indicates that it should be simple to **compress** English text **by** about **70%**.
- ✓ This sort of redundancy is a *unitless* **relative** redundancy

¹That is, without considering letter *sequences*.



Unicity distance



Defined by Shannon - an **approximation** to the amount of ciphertext such that the the sum of the source entropy and the encryption key entropy is the same as the number of ciphertext bits used.

- ✓ Ciphertexts longer have only one meaningful decryption
- ✓ Ciphertexts shorter may have more than one meaningful decryption (and hence be stronger, as a hacker will not know which one is correct)



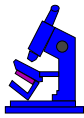
Unicity distance



- ✓ The **longer** the unicity distance, the **better** the cryptosystem
- ✓ Unicity distance U is the entropy of the key divided by the redundancy of the source, and is approximately

$$U \approx \frac{\log_2 K}{R \log_2 P}$$

(K is the key size, R is the redundancy, P is the number of symbols).



Unicity distance



26 letter alphabet, and $26!$ keys

$$\begin{aligned} U &\approx \frac{\log_2 26!}{0.5 \log_2 26} \\ &\approx \frac{88}{0.7 * 4.7} \\ &\approx 27 \end{aligned}$$

So given a ciphertext of 27 symbols, a unique decoding is possible.

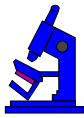


Unicity distance

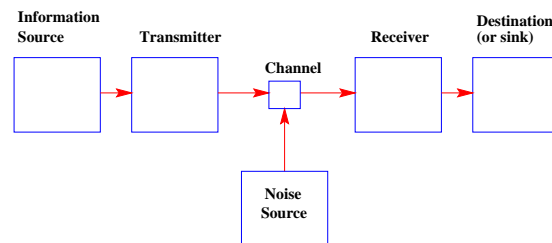


In general

- ✓ Longer key length then longer unicity distance
- ✓ Redundancy inversely proportional to unicity distance
- ✓ Estimates the minimum amount of ciphertext for which there is only a single plaintext solution on doing a brute force attack...



Shannon and Nyquist



$$\text{Maximum BPS} = W \log_2\left(1 + \frac{S}{N}\right) \text{ bits/sec}$$



Shannon and Nyquist example



If we had a telephone system with a bandwidth of 3,000 Hz, and a S/N of 30db (about 1024:1)

$$\begin{aligned} D &= 3000 * \log_2 1025 \\ &\approx 3000 * 10 \\ &\approx 30000 \text{ bps} \end{aligned}$$

This is a typical maximum bit rate achievable over the telephone network.



Nyquist



The maximum data rate over a limited bandwidth (W) channel with V discrete levels is:

$$\text{Maximum data rate} = 2W \log_2 V \text{ bits/sec}$$

For example, two-Level data cannot be transmitted over the telephone network faster than 6,000 BPS, because the *bandwidth* of the telephone channel is only about 3,000Hz.



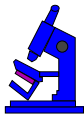
Nyquist example



If we had a telephone system with a bandwidth of 3,000 Hz, and using 256 levels:

$$\begin{aligned} D &= 2 * 3000 * \log_2 256 \\ &= 6000 * 8 \\ &= 48000 \text{ bps} \end{aligned}$$

In these equations, the assumption is that the relative entropies of the signal and noise are a maximum (that they are random).



Maximum entropy

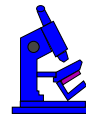


In practical systems, signals rarely have maximum entropy, and we can do better - there may be methods to compress the data².

²**Note:** we must also differentiate between lossy and lossless compression schemes. A signal with an entropy of 0.5 may not be compressed more than 2:1 unless you use a lossy compression scheme. JPEG and Wavelet compression schemes can achieve huge data size reductions without visible impairment of images, but the restored images are not the same as the original ones - they just look the same. The lossless compression schemes used in PkZip, gzip or GIF files (LZW) cannot achieve compression ratios as high as that found in JPEG.



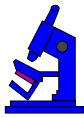
Huffman encoding



An immediate question of interest is “*What is the **minimum length bit string** that may be used to **compress** a string of symbols?*”.

The **Huffman** encoding minimizes the bit length given the frequency of occurrence of each symbol³. The resultant bit string in the best case will be the length predicted from the calculation of the source entropy.

³Note that it presupposes knowledge about these frequencies.



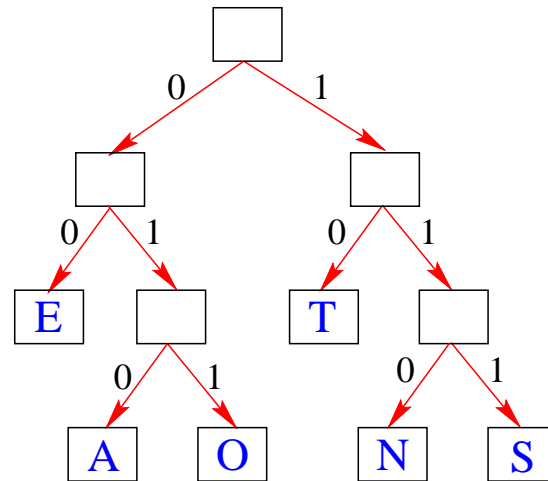
Huffman encoding



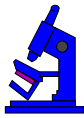
- ✓ How can we get **knowledge** about the **frequency** of (say) the **letters** in the **English** language?



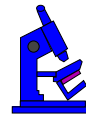
Huffman encoding



Less common characters use longer bit strings.



Huffman encoding



Our [algorithm](#) for encoding is simple - we calculate the tree encoding knowing the frequency of each letter:

Symbol	Coding
E	00
T	10
A	010
O	011
N	110
S	111

To [decode](#), traverse the tree taking a left or right path according to the bit. The leaf has our symbol.



Case study - MNP5 and V.42bis



MNP5 and V42.bis are compression schemes commonly used on modems.

MNP5 suffers from the unfortunate property that it will *expand* data with maximum or near-maximum entropy (instead of compression).

V42.bis does not have this property - it uses a large dictionary, and will not try to compress an already compressed stream.



MNP5



MNP5 uses two different compression methods, switching between them as appropriate. The methods are:

- Adaptive frequency encoding
- Run-length encoding

Run length encoding sends the bytes with a byte count value, and doubles the size of a data stream with maximum entropy.

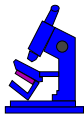


Adaptive frequency encoding



3-bit header	Body size	Total code size	Number of codewords
000	1 bit	4 bits	2
001	1 bit	4 bits	2
010	2 bits	5 bits	4
011	3 bits	6 bits	8
100	4 bits	7 bits	16
101	5 bits	8 bits	32
110	6 bits	9 bits	64
111	7 bits	10 bits	128

$\frac{3}{4}$ of our codewords are *larger* than they would be if we did not use this encoding scheme



Summary of topics



In this section, we introduced the following topics:

- Physical preliminaries, Fourier analysis and convolution
- Entropy
- Encoding



Further study



- Textbook Chapter 32
- Shannon's paper on secrecy systems [?] at <http://www.cs.ucla.edu/~jkong/research/security/shannon.html>.

