Computer networks: data is digital (i.e., bits)

→ high-speed (broadband transmission): analog

But some data or information starts out analog

→ e.g., voice, audio, video

→ must make digital to send over computer networks

→ i.e., analog-to-digital conversion

Not the end of the story:

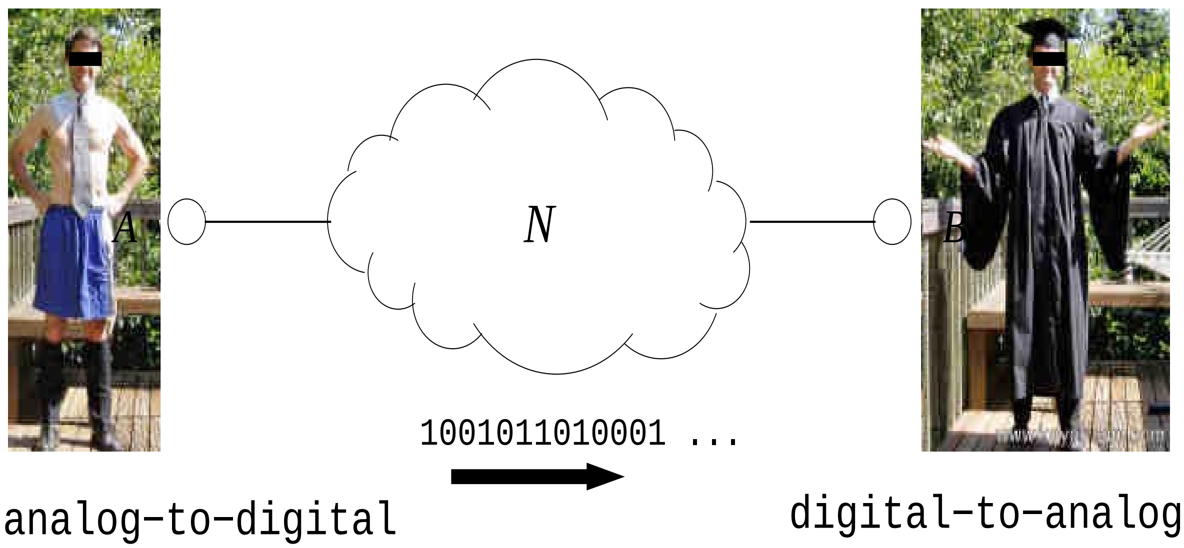
→ consumer of digitized analog data: human

→ at the end: must convert back to analog!

→ fidelity issue (aka garbage-in-garbage-out)

→ key issue when doing analog-to-digital

Problem: How to avoid



Problem: How can we convert analog information to digital data so that when we convert back to analog (after transmission over computer networks) the analog information looks the same as its original?

$\longrightarrow$ other benefits of digitizing?

What does digitizing mean?

Two things:

- time: from continous time to discrete time

  → called sampling

  → good quality video (e.g., movie theatre)?

- strength: amplitude is discretized

  → 8 and 16 bits: popular and sufficient

  → note: logarithmic scale

So, can one always digitize without losing fidelity?

⟶ no

⟶ when is this possible?

Fidelity can be preserved when analog signal is

$\longrightarrow$ bandlimited

Note: complicated-looking analog signal are just sums of scaled sine curves (building blocks)

Bandlimited means:

$\longrightarrow$ high frequency sine curves are not needed

$\longrightarrow$ can ignore sines with frequency $> \omega_*$

What use is it to us?

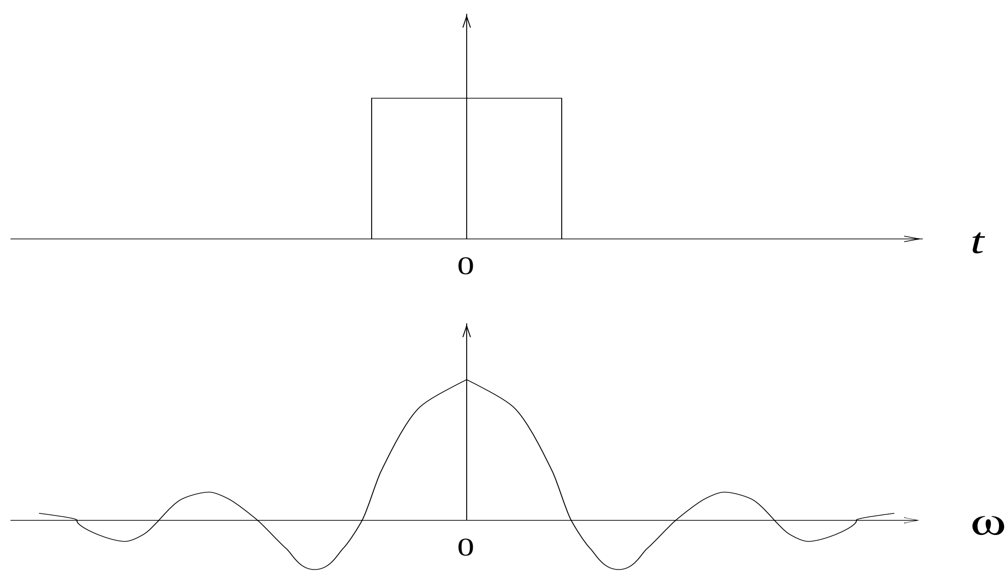
$\longrightarrow$ most signals in nature are bandlimited

$\longrightarrow$ so are signals from engineered (man-made) systems

Square wave (man-made):

$\longrightarrow$ nature doesn't provide many square waves

$\longrightarrow$ $s(t)$ and $S(\omega)$ profiles



$\longrightarrow$ strictly speaking: not bandlimited!

$\longrightarrow$ what to do?

"Laid back" approach: approximation

$\longrightarrow$ square wave: cut the tails off $S(\omega)$

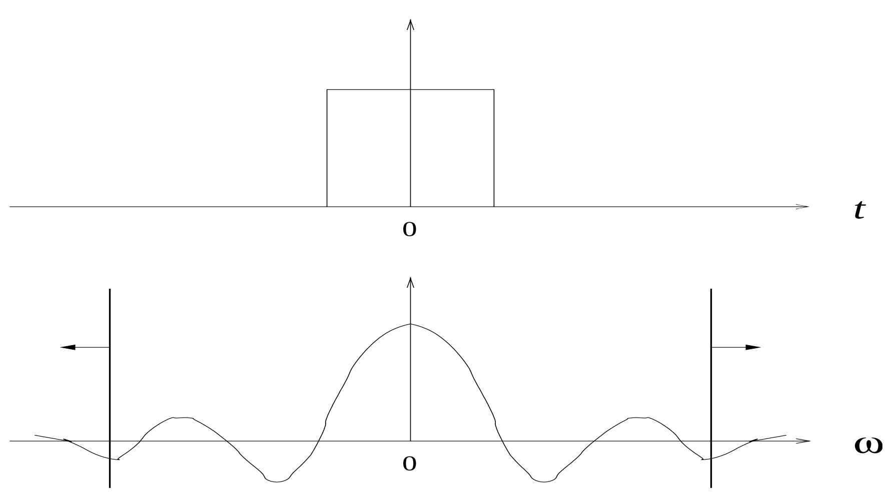
$\longrightarrow$ let's approximate!

$\longrightarrow$ when $S(\omega) \approx 0$, can treat as $S(\omega) = 0$

$\longrightarrow$ i.e., $S(\omega) = 0$ for $|\omega|$ sufficiently large

$\longrightarrow$ now: bandlimited

$\longrightarrow$ what will the square without tail look like?



Ex.: human auditory system

$\longrightarrow$ 20 Hz–20 kHz

$\longrightarrow$ speech is intelligible at 300 Hz–3300 Hz

$\longrightarrow$ broadcast quality audio; CD quality audio

Telephone systems: engineered to exploit this property

$\longrightarrow$ bandwidth 3000 Hz

$\longrightarrow$ throw out: sines above 3300 Hz

$\longrightarrow$ that's why voice quality is not good

$\longrightarrow$ we're missing: 3300–20 kHz sine waves!

$\longrightarrow$ CD quality: much better

But why throw out voice data in 3300–20 kHz range?

Intuition behind sampling:

$\longrightarrow$ signal varies rapidly: more samples

$\longrightarrow$ signal varies slowly: can do with less samples

$\longrightarrow$ think of camera shutter speed in sports

$\longrightarrow$ e.g., baseball (or golf)

What about a single sine wave with period T?

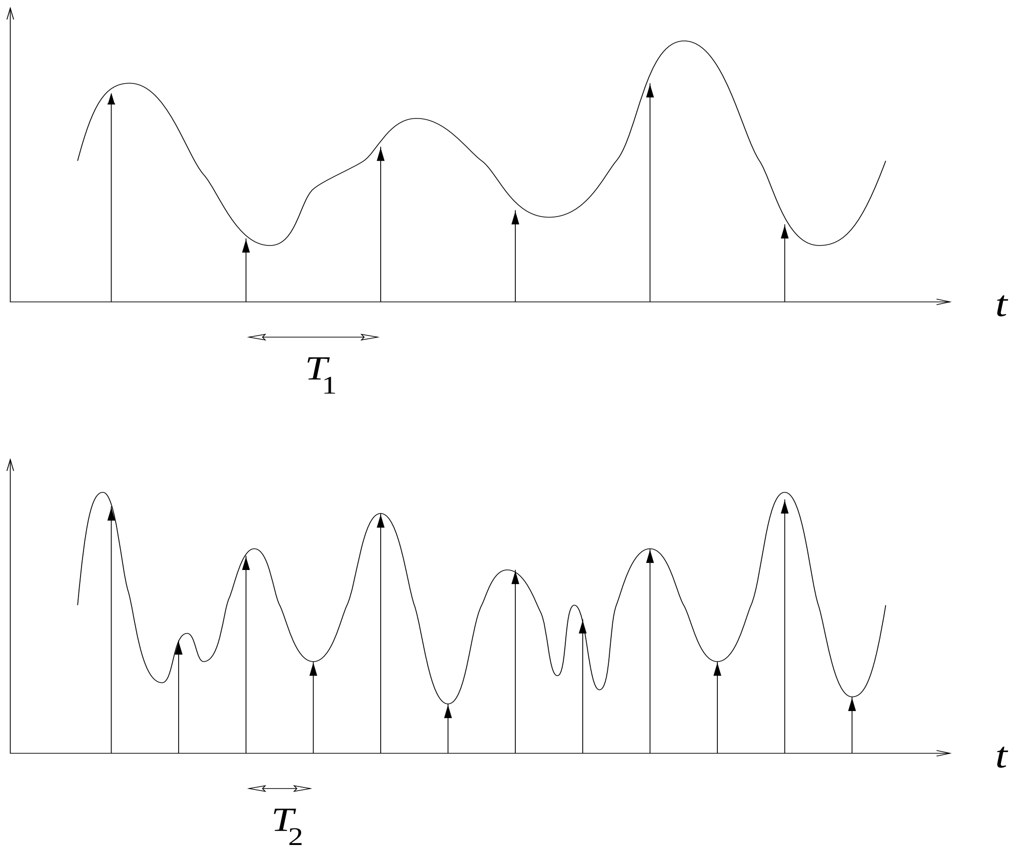
$\longrightarrow$ how many samples (snapshots) are needed

$\longrightarrow$ express sample count in terms of T

What about bandlimitedness?

$\longrightarrow$ is there a relationship to sampling?

Slowly vs. rapidly varying signal:



If a signal varies quickly, need more samples to not miss details/changes.

$$\text{we have:} \quad \nu_1 = 1/T_1 \;\; < \;\; \nu_2 = 1/T_2$$

Sampling criterion for guaranteed fidelity:

Sampling Theorem (Nyquist): Given continuous *bandlimited* signal $s(t)$ with $S(\omega) = 0$ for $|\omega| > W$, $s(t)$ can be reconstructed from its samples if

$$\nu > 2W$$

where ν is the sampling rate.

$\longrightarrow$ ν: samples per second

Remember simple rule: sample twice the max bandwidth

$\longrightarrow$ e.g., in T1 line: 8000 samples per second

$\longrightarrow$ along with 8 bits $(= 7 + 1)$ gave 1.544 Mbps

$\longrightarrow$ why 8000 samples per second?

Compression

Information transmission over noiseless medium

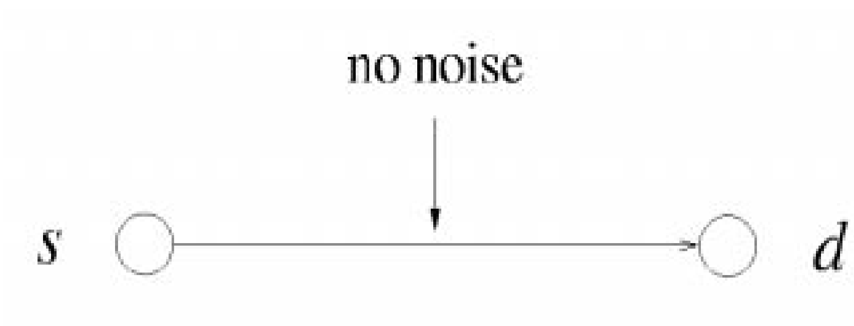
$\longrightarrow$ medium or "channel"

$\longrightarrow$ fancy name for copper wire, fiber, air/space

Sender wants to communicate information to receiver over noiseless channel.

$\longrightarrow$ can receive exactly what is sent

$\longrightarrow$ idealized scenario



Set-up:

$\longrightarrow$ take a system perspective

$\longrightarrow$ e.g., modem manufacturer

Need to specify two parts: property of data source—what are we supposed to send?—and how compression is done.

$\longrightarrow$ need to know what we're dealing with

$\longrightarrow$ if we want to do a good job compressing

$\longrightarrow$ two parts

Part I. What does the (data) source look like:

- source s emits symbols from finite alphabet set Σ

  $\rightarrow$ e.g., $\Sigma = \{0, 1\}$; $\Sigma =$ ASCII character set

- symbol $a \in \Sigma$ is generated with probability $p_a > 0$

  $\rightarrow$ e.g., books have known distribution for 'e', 'x' ...

  $\rightarrow$ let's play "Wheel of Fortune"

Part II. Compression machinery:

- code book F assigns code word $w_a = F(a)$ for each symbol $a \in \Sigma$

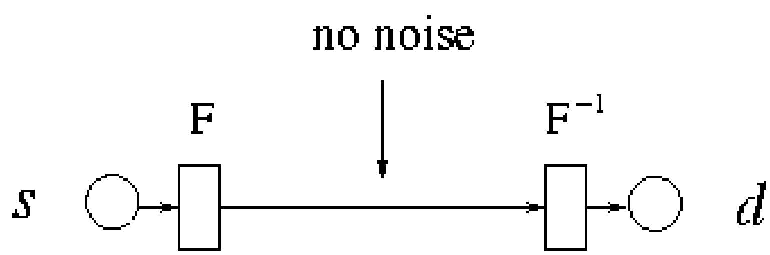
  $\rightarrow$ w_a is a binary string of length $|w_a|$

  $\rightarrow$ F could be just a table

- F is invertible

  $\rightarrow$ receiver d can recover a from w_a

  $\rightarrow$ F^{-1} is the same table, different look-up



Ex.: $\Sigma = \{A, C, G, T\}$; need at least two bits

- F^1: $w_A = 00$, $w_C = 01$, $w_G = 10$, $w_T = 11$
- F^2: $w_A = 0$, $w_C = 10$, $w_G = 110$, $w_T = 1110$

$\longrightarrow$ pros & cons?

Note: code book F is not unique

$\longrightarrow$ find a "good" code book

$\longrightarrow$ when is a code book good?

"Hoodness" measure: average code length L

$$L = \sum_{a \in \Sigma} p_a |w_a|$$

$\longrightarrow$ average number of bits consumed by given F

Ex.: If DNA sequence is 10000 letters long, then require on average $10000 \cdot L$ bits to be transmitted.

$\longrightarrow$ good to have code book with small L

$\longrightarrow$ very practical concern

Optimization problem: Given source $\langle \Sigma, \mathbf{p} \rangle$ where $\mathbf{p}$ is a probability vector, find a code book F with least L.

$\longrightarrow$ practically super-important

$\longrightarrow$ shrink-and-send

$\longrightarrow$ lossless shrinkage

Limit to what is achievable to attain small L.

$\longrightarrow$ kind of like speed-of-light

First, define entropy H of source $\langle \Sigma, \mathbf{p} \rangle$

$$H = \sum_{a \in \Sigma} p_a \log \frac{1}{p_a}$$

Ex.: $\Sigma = \{A, C, G, T\}$; H is maximum if $p_A = p_C = p_G = p_T = 1/4$.

$\longrightarrow$ when is it minimum?

Source Coding Theorem (Shannon): For all code books F,

$$H \leq L_F$$

where L_F is the average code length under F.

Furthermore, L_F can be made to approach H by selecting better and better F.

Remark:

- to approach minimum H use blocks of k symbols

  → e.g., treat "THE" as one unit (not 3 separate letters)

  → called extension code

- entropy is innate property of data source s
- limitation of ensemble viewpoint

  → e.g., sending number $\pi = 3.1415927\ldots$

  → better way?