Fundamentals of information transmission and coding (a.k.a. data communication)

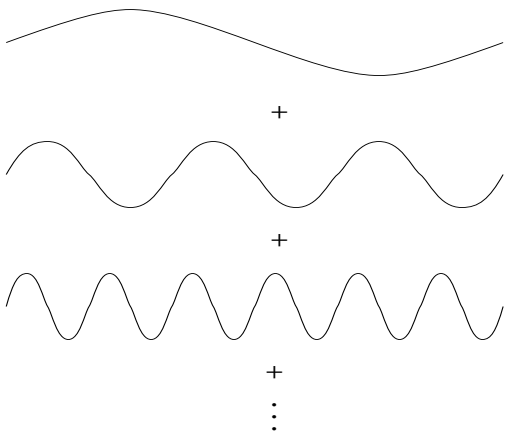
Signals and functions

Elementary operation of communication: send *signal* on medium from point A to point B.

- media: copper wire, optical fiber, air, etc.
- signals: voltage and currents, light pulses, radio waves, etc.

Signal can be viewed as a time-varying function $s(t)$.

If $s(t)$ is "sufficiently nice" (Dirichlet conditions), then $s(t)$ can be represented as a linear combination of complex sinusoids.



Fourier expansion and transform:

$$s(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} S(\omega) e^{i\omega t} d\omega,$$

$$S(\omega) = \int_{-\infty}^{\infty} s(t) e^{-i\omega t} dt.$$

$\longrightarrow$ time domain vs. frequency domain

Luckily, most "interesting" functions are far from random, in fact, *bandwidth limited*.

Speech: 20Hz–20kHz; telephone systems: 300Hz–3300Hz

$\longrightarrow$ bandwidth 3000Hz

Random function (i.e., white noise) has constant spectrum.

$\longrightarrow$ unbounded bandwidth

Digital data vs. analog data

Digital data: bits.

Analog data: voice, image; except for telephony, analog data is oftentimes *digitized* so that bits are the starting point.

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.

Information transmission over noisy channel

Channel capacity C: maximum achievable "reliable" data transmission rate (bps) over a noisy channel.

Channel Coding Theorem (Shannon): Given signal power P_S, noise power P_N, and channel subject to white Gaussian noise (detailed conditions omitted),

$$C = \frac{1}{2}\nu \log(1 + \frac{P_S}{P_N}) \text{ bps.}$$

Here P_S/P_N: *signal-to-noise ratio*.

By the Sampling Theorem,

$$C = W \log(1 + P_S/P_N) \text{ bps.}$$

Signal-to-noise ratio is expressed as

$$\text{dB} \;=\; 10 \log_{10}(P_S/P_N).$$

Example: Assuming a decibel level of 30, what is the channel capacity of a telephone line?

Answer: First, $W = 3000$Hz, $P_S/P_N = 1000$. Using Channel Coding Theorem,

$$C \;=\; 3000 \log 1001 \;\approx\; 30\text{kbps}.$$

Compare against 28.8kbps modems.

Digital vs. analog signals

In essence, *square wave* versus everything else.

Two forms of *transmission*:

- digital transmission: data transmission using square waves
- analog transmission: data transmission using all other waves

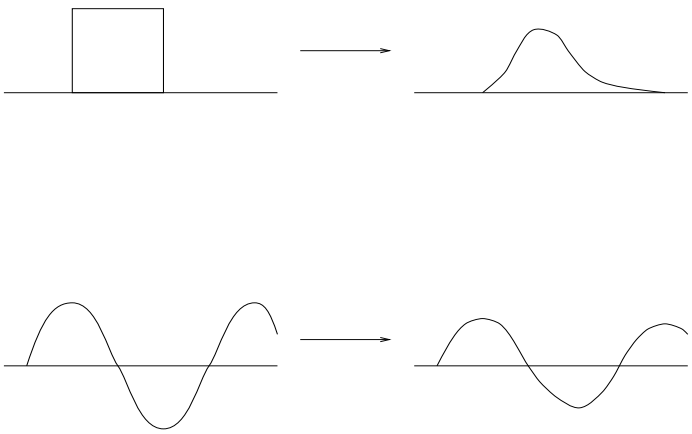
Analog data via analog transmission is straightforward; e.g., telephone at home.

Need to consider:

- digital data via analog transmission
- digital data via digital transmission

First, why is digital transmission "superior" to analog transmission?

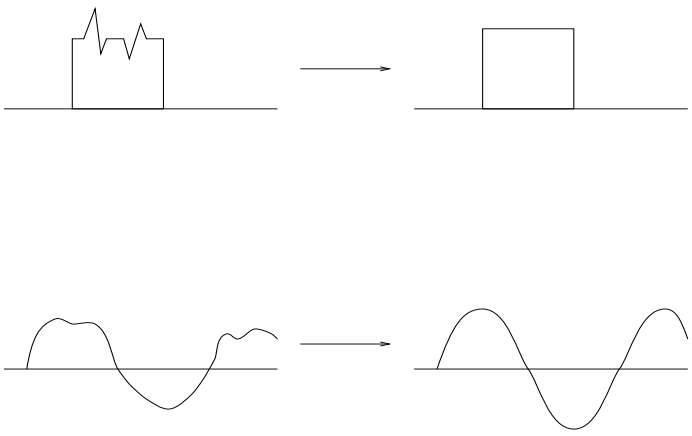
Common to both: problem of *attenuation*.



Rejuvenation of signal via amplifiers (analog) and repeaters (digital).

More problematic: effect of noise.

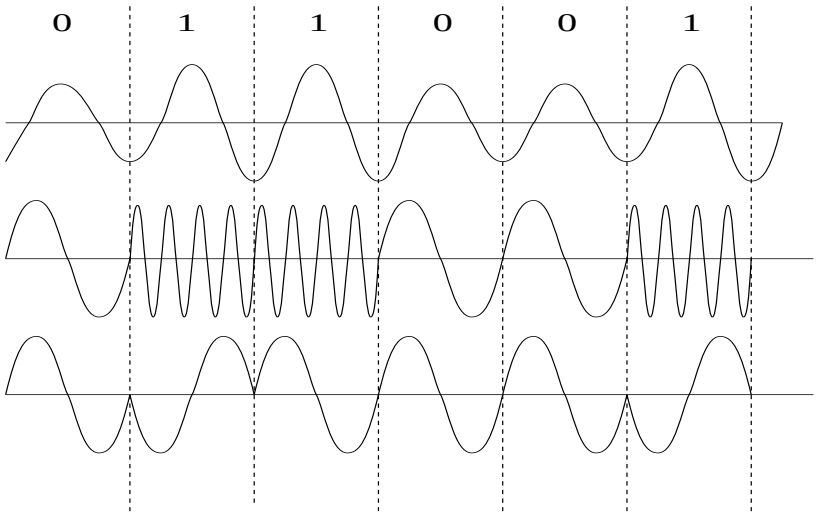
- Analog: Amplification also amplifies noise—filtering out just noise, in general, is a hard problem; e.g., voice.
- Digital: Repeater just generates a new square wave. Noise cannot be confused with data (unless too high).



Analog transmission of digital data

Three pieces of information to manipulate: amplitude, frequency, phase.

- Amplitude modulation (AM)—sensitive to power fluctuations.
- Frequency modulation (FM)—allow full duplex communication; need four frequencies.
- Phase modulation (PM)—more sophisticated.



Baud rate vs. bit rate

Baud rate: Unit of time within which carrier wave can be altered for AM, FM, or PM.

At every signalling event, potentially more than 1 bit of information can be encoded; e.g., PM with multiple phases, say, four.

Thus 2 bits of information per signalling period.

$\longrightarrow$ bit rate (bps) $= 2 \times$ baud rate

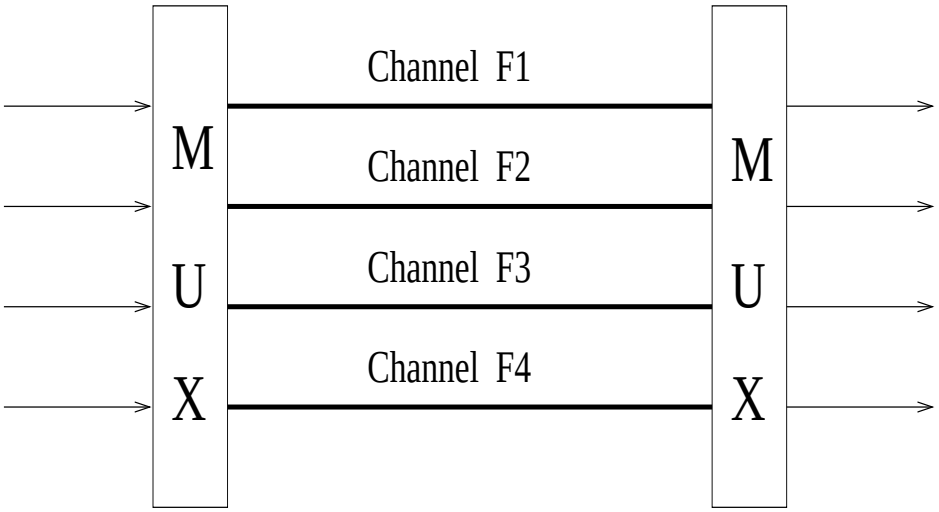
Combine the three; e.g., QAM—8 phase angles and 2 amplitudes for a total of 16 detectable events (CCITT v.29 standard, 9600bps, 2400baud).

$\longrightarrow$ 4 bits per baud

Broadband vs. baseband

Presence/absence of *carrier wave*; allows many channels to co-exist at the same time.

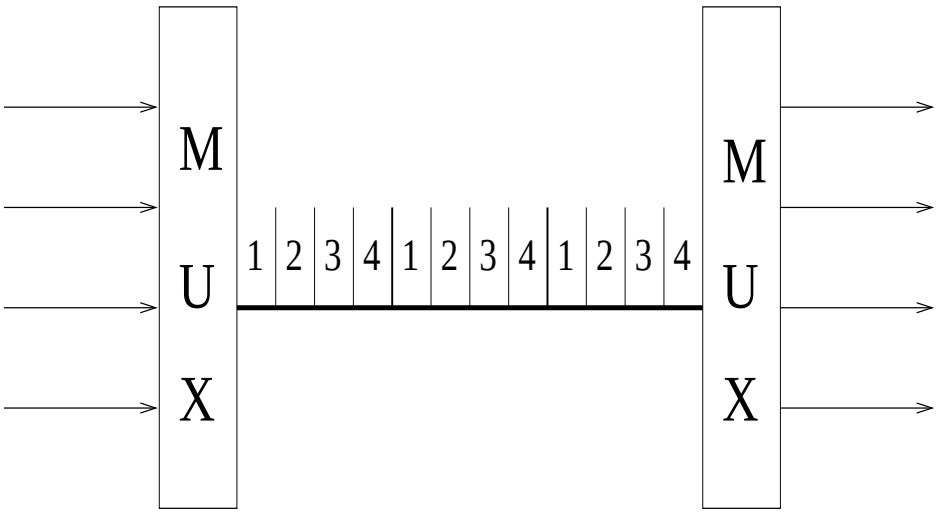
$\longrightarrow$ frequency division multiplexing (FDM)



Clearly, BW of medium $>$ 4 $\times$ BW of signal.

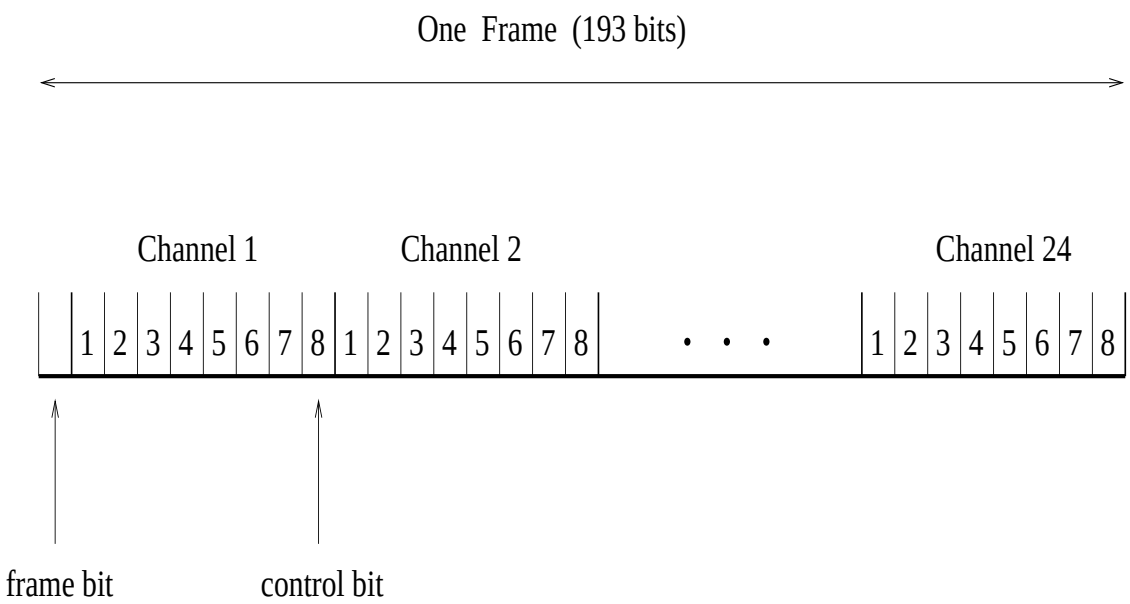
In the absence of carrier wave, can still use multiplexing:

$\longrightarrow$ time-division multiplexing (TDM)



Clearly, bit rate of medium > data rate of signal; however, mostly used for digital transmission of digital or analog data (PCM, codec).

Example: T1 carrier.



Assuming 4kHz telephone channel bandwidth, Sampling Theorem dictates 8000 samples per second (125 μsec/sample).

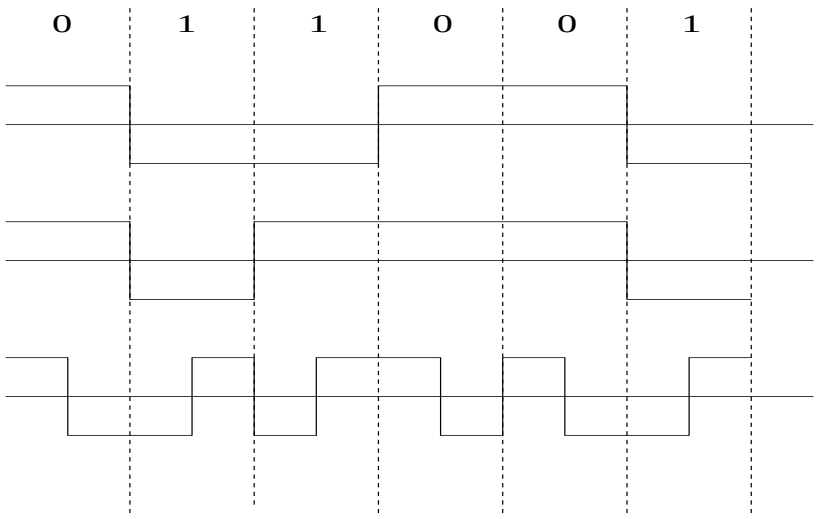
Bandwidth $= 8000 \times 193 = 1.544$Mbps

Digital transmission of digital data

Direct encoding of square waves using voltage differentials; e.g., -15V–+15V for RS-232-C.

NRZ-L (non-return to zero, level), NRZI (NRZ invert on ones);

Manchester (biphase or self-clocking codes).



Trade-offs:

- NRZ codes—long sequences of 0's (or 1's) causes synchronization problem; need extra control line (clock) or sensitive signalling equipment.
- biphase codes—synchronization easily achieved through self-clocking; however, other things being equal, achieves only 50% efficiency vis-a-vis NRZ codes.

4B/5B code

Encode 4 bits of data using 5 bit code where the code word has at most one leading 0 and two trailing 0's.

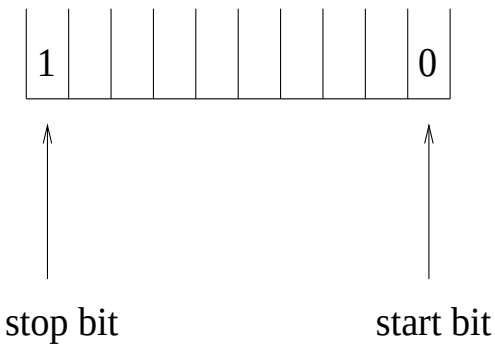
$0000 \leftrightarrow 11110$, $0001 \leftrightarrow 01001$, etc.

$\longrightarrow$ using NRZI, at most three consecutive 0's

$\longrightarrow$ efficiency: 80%

Synchronous vs. asynchronous transmission
("framing problem")

Asynchronous: e.g., ASCII character transmission between dumb terminal and host computer.

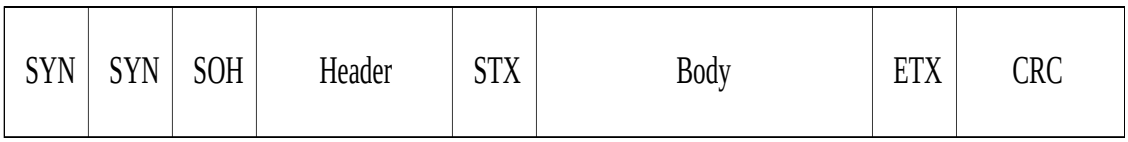


Each character is an independent unit; receiver needs to know bit duration.

Overhead problem; assuming 1 start bit, 1 stop bit, 8 data bits, only 80% efficiency.

$\longrightarrow$ inefficient for long messages

Synchronous: "Byte-oriented scheme"; e.g., BISYNC



Two problems:

- How to maintain synchronization if |Body| is large?
- Control characters within Body of message.

  $\longrightarrow$ inefficient for short messages

  $\longrightarrow$ efficiency approaches 1 as $|\text{Body}| \rightarrow \infty$

"Bit-oriented scheme"; e.g., HDLC

Use fixed *preamble* and *postamble*; simply a bit pattern.

$\longrightarrow$ 01111110

How to avoid confusing 01111110 in the data part?

$\longrightarrow$ bit stuffing

Error-detection and correction

General theory: subject of *Information Theory*. E.g., Hamming codes, Huffman codes, Shannon-Fano codes, Reed-Solomon, etc.

$\longrightarrow$ practical error detection

Parity: Odd or even parity; single bit error detection.

(Internet) Checksum: Group message into 16-bit words; calculate their sum (one's complement); append "checksum" to message.

Cyclic redundancy check (CRC): Polynomial arithmetic over finite field.

Represent n-bit string $a_{n-1}a_{n-2}\cdots a_0$ as a polynomial of degree $n-1$:

$$M(x) = a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \cdots + a_1x + a_0.$$

Fix some generator polynomial $G(x)$ of degree k. Choice of $G(x)$ is important.

Let $R(x)$ be the remainder of $x^k M(x)/G(x)$.

Let $T(x) = x^k M(x) - R(x)$.

$\longrightarrow$ $T(x)$ is the code word

Assume $T(x) + E(x)$ arrives at the receiver.

If $E(x) = 0$ (no errors), then $T(X)/G(x) = 0$.

If $E(x) = x^i$, $0 \leq i \leq n + k - 1$ (i.e., a single error at position i), then assuming $G(x)$ contains at least two terms, $G(x)$ will fail to divide $E(x)$.

$\longrightarrow$ detection of error

If $E(x) = x^i + x^j$, $i > j$, then first express $E(x) = x^j(x^{i-j} + 1)$.

Assuming x does not divide $G(x)$, to detect double errors it is sufficient that $G(x)$ not divide $x^{i-j} + 1$.

Fact: $G(x) = x^{15} + x^{14} + 1$ will not divide $x^r + 1$ for all $k < 32768$.

If number of errors is odd, then $x + 1$ does not divide $E(x)$. Hence, choose $G(x)$ such that $x + 1$ is a factor of $G(x)$.

All error bursts of size $\leq k$ are detected.