

Physical Layer

Lecture 3 Physical Media and Attenuation

The physical layer is about communication between adjacent nodes.

→ What signals do we send?

→ How do we deal with distortion?

Say A is sending a message to B.

(binary: 0 → 0V, 1 → 5V)

Ideally, B would receive exactly what A sends.

What could go wrong?


→ There could be magnetic flux through the circuit, messing up the voltages.

due to currents between other people talking

$\text{emf} \propto \frac{dB}{dt}$

"Cross-talk"

How to fix?

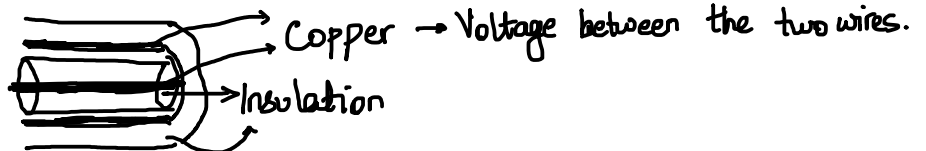
- Reduce area of loop.
- Twisted-pair.  reduces flux. If done well, eliminates most cross-talk

CAT-3 cables have 10 Mbps over 100m.

CAT-5 cables have 100 Mbps over 100m.

CAT-6 cables have 1 Gbps over 100m.
10 Gbps over 50m.

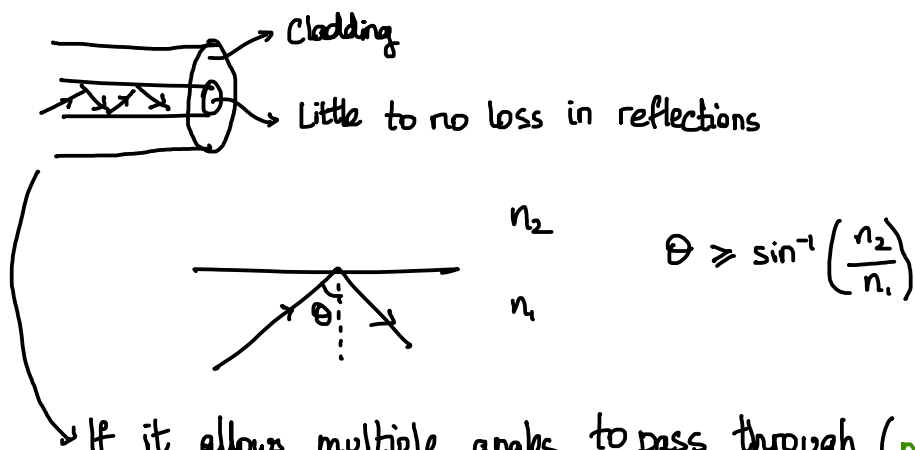
- Ethernet connectors have many wires coming in.
(8 wires — 4 twisted pairs)
- Co-axial cables have a layer of shielding / insulation.



(0.2 in) ThinNet Coax give 100Mbps over 200m

(0.4 in) ThickNet Coax give 100 Mbps over 500m

- Optic fibres allow very high data rate. They use light rays instead of electric signals, thus reducing attenuation.
- fired from laser.



→ If it allows multiple angles to pass through (**modes**), it is called **multi-mode fibre**.

They are not the best because different modes can overlap. In single-mode fibre, only one mode passes through.

→ Reduce diameter or increase refractive index
More expensive.

Say A transmits amplitude A_{in} and B receives amplitude A_2 .

The **attenuation**, measured in dB, is equal to

$$10 \log_{10} \left(\frac{P_{in}}{P_{out}} \right) = 20 \log_{10} \left(\frac{A_{in}}{A_{out}} \right)$$

↓
Powers

It is usually measured in dB/1000 ft.

Note that it is additive over distance.

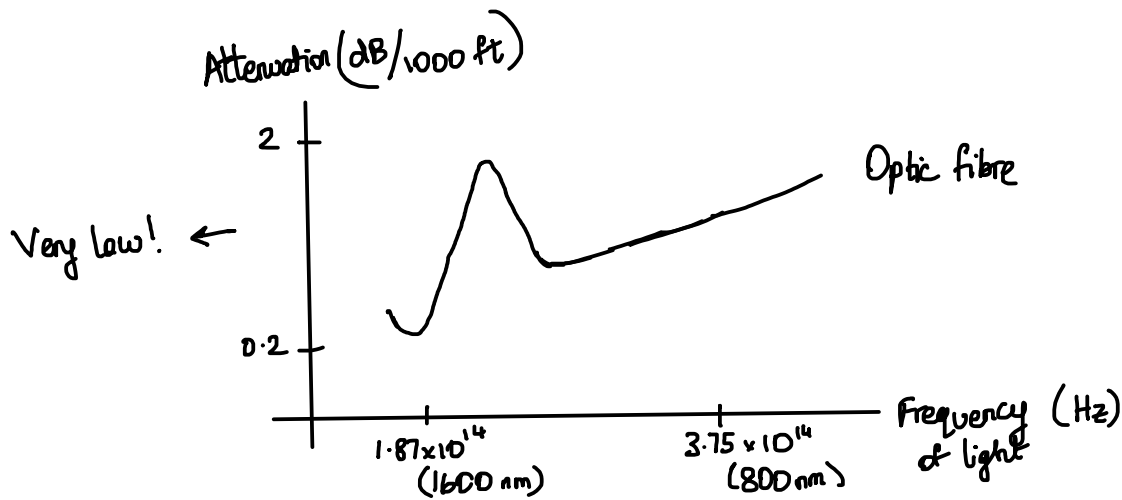
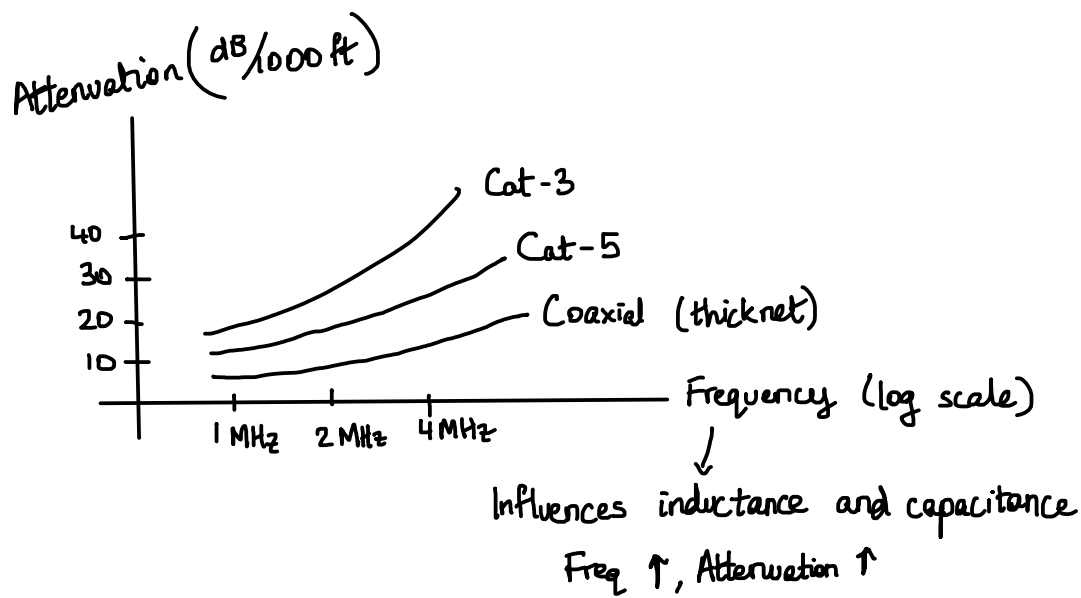
$$\begin{array}{c} A_{in} \quad \quad A' \quad \quad A_{out} \\ \hline \underbrace{\quad x \text{ dB} \quad \quad y \text{ dB} \quad}_{(x+y) \text{ dB} \text{ attenuation total}} \end{array} \quad \left(\frac{A_{out}}{A_{in}} \right) = \left(\frac{A_{out}}{A'} \right) \cdot \left(\frac{A'}{A_{in}} \right)$$

If P goes to $P/2$, attenuation is $\sim 3\text{dB}$.

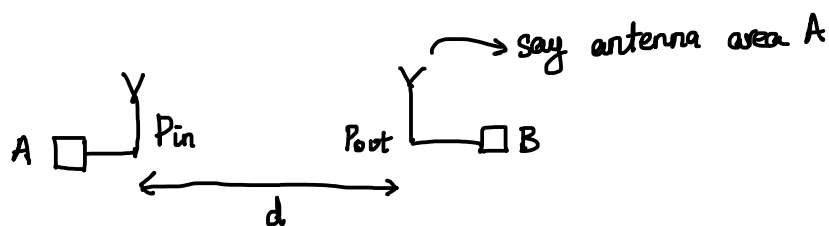
We also measure absolute power in dBm, dBW.

Suppose power P .

$$10 \log_{10} \left(\frac{P}{1 \text{ mW}} \right) \rightarrow \text{Power in dBm.}$$



Say A and B are communicating wirelessly. A sends signals via antenna.



If power is spread isotropically in all directions

$$P_{out} \propto \frac{P_{in} \cdot A}{d^2}$$

In real life, it's even worse
at d^α for some $2 < \alpha < 5$.

d^2 → Huge power loss with distance.

To solve this, we sometimes use directional antennas.

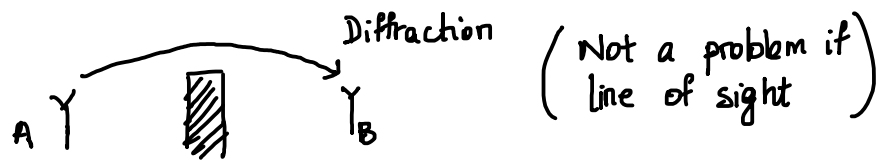
(dish antennas for example)

Nowadays, we use MIMO (multiple-input and multiple-output) or more recently, massive MIMO.

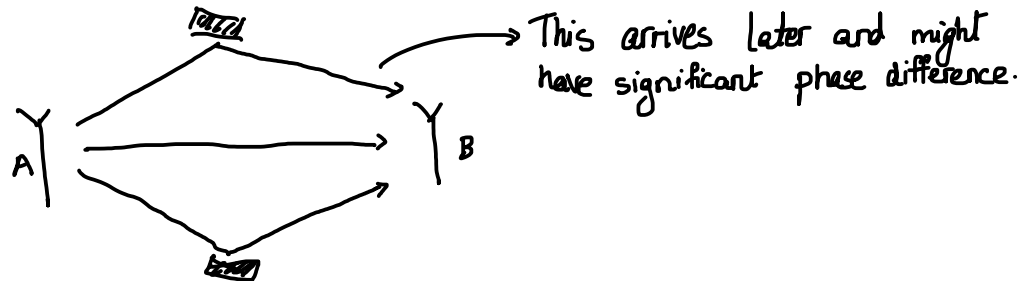
In the wireless case, we could have

→ Interference (counterpart of cross-talk)

→ Obstructions



→ Multipath (counterpart of multimode)



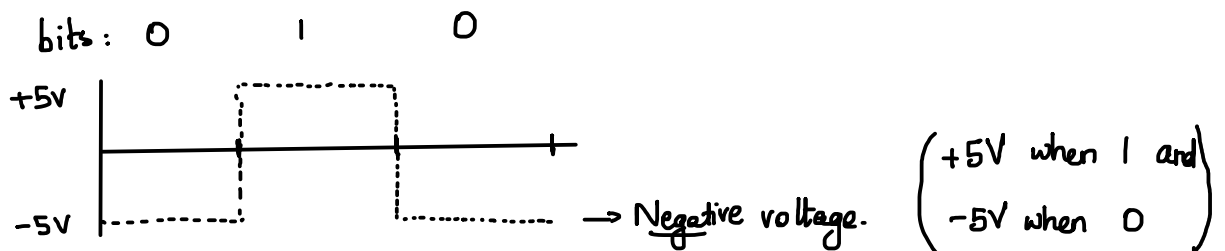
Lecture 4 Line Coding

Given a set of bits, we have to convert them into signals. How?

Wired situation.

We use **Line coding**.

- **Non-return to zero (NRZ)**

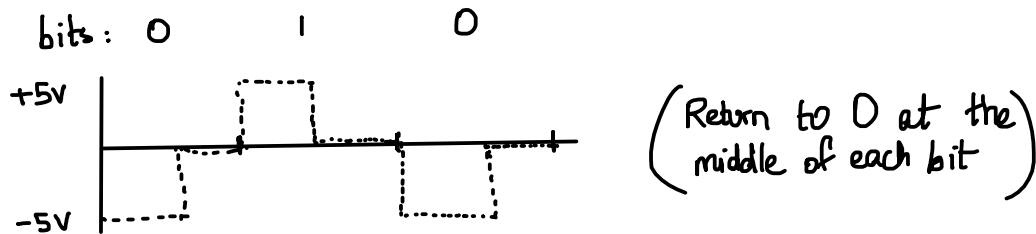


Issues with NRZ:

- **Clock recovery** - 001100 and 010 are essentially the same. How would we get the actual clock duration of the sender? The clock may be slightly slower/faster due to imperfections and may also have "drift".
↳ getting even slower/faster with time
We are unable to recover the clock from the signal.
- **Baseline wander** - Instead of being exactly -5 and +5, there may be some offset. The **DC offset** is the average of the two offsets - the "new 0". This leads to errors other than the usual errors itself.

If we use a high pass filter, it partially resolves it. However, if there is a long string of 0s or 1s, it gets messed up.

- Return to Zero (RZ)



We never have a constant signal for a long time, so high pass filters can remove baseline wander.

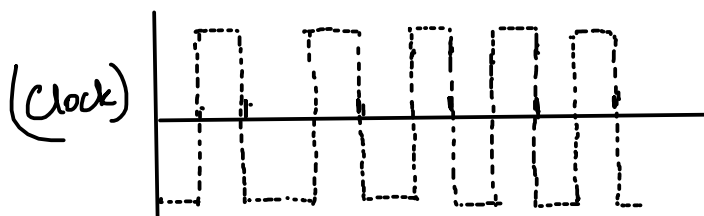
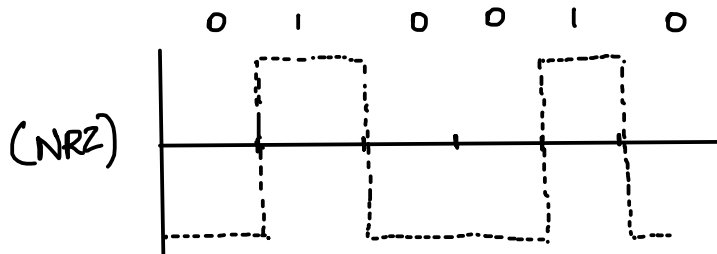
However, the issue is that we now have three levels.

- Manchester Coding (802.3 IEEE)

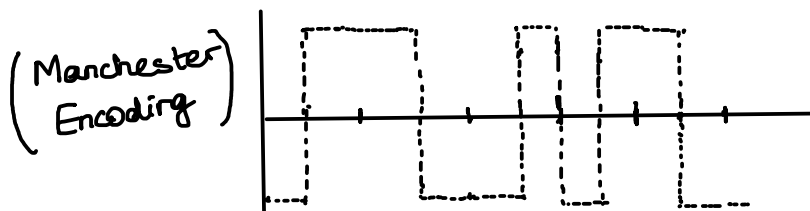
↳ standard for ethernet

This is what is used usually.

We start with NRZ.



We then XOR the two.



The regular transitions eliminate the issue of clock recovery.
Like RZ, we can use a high pass filter to eliminate baseline wander.
(but only two voltage levels)

Note that the encoding of a 0 has a positive transition
1 has a negative transition

We can recognize the bit if we know where it starts.

1. We need to know where each bit begins.
2. We mustn't mess up the polarity.

To do 1, we use a known signal known as the **preamble** to synchronize. Before the actual message, we insert this special signal.
It can be thought of as a physical layer header.

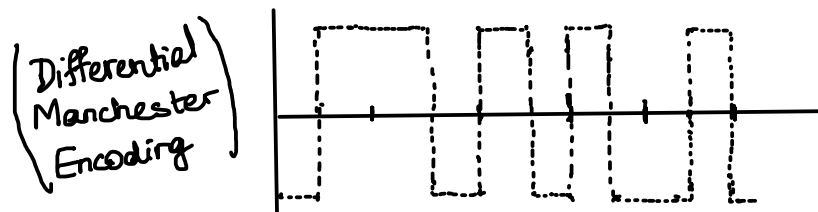
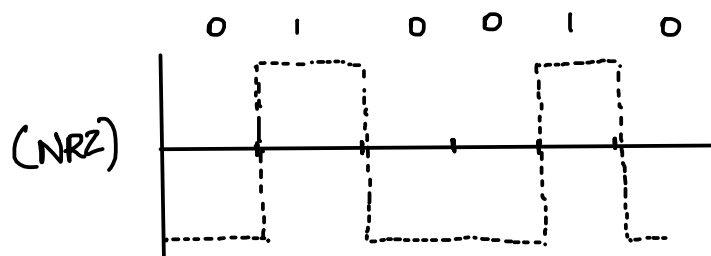
- **Differential Manchester Encoding** (802.5 IEEE)

It is used in token ring LANs (see DLL)

For a 0, the first half is opposite of the last half of the previous bit.

1, the first half is equal to the last half of the previous bit.

Within each bit, there is a clock-like signal.



This has the advantages of Manchester encoding and further, the polarity issue is now gone.

(just see if it inverts or stays the same)

- 4B/5B Encoding:

Recall that in NRZ, we run into issues if the same bit occurs multiple times.

What if we manually change it once in a while?

What this does is that for an input of 4 bits, it gives out 5 bits. For example,

$$\begin{array}{lcl} 0000 & \rightarrow & 01010 \\ 1000 & \rightarrow & 10010 \\ & \vdots & \\ & & \end{array} \left. \vphantom{\begin{array}{lcl} 0000 \\ 1000 \end{array}} \right\} 2^4 \text{ such things.}$$

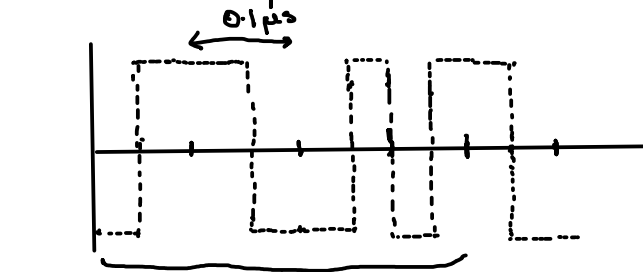
This ensures that there are at most 3 consecutive 0s/1s in the encoded bit string.

This is disadvantageous because there is some degree of redundancy.

Baud Rate:

Recall that the bit rate is the maximum possible number of bits transferred per second.

The Baud rate is the maximum allowable number of symbol changes made to the transmission medium per second.



for example, 5 symbol changes
Worst case, one change every $0.05 \mu\text{s}$

$$\text{Baud rate} = 2 \times 10^7 \text{ symbols s}^{-1}$$

In some sense, Baud rate captures what potential we can use the medium to. The allowable frequency range is related.

↳ recall that as freq. ↑, attenuation ↑.

An allowable signal should have most of the Fourier transform contained within the allowable frequencies.

Observe that in Manchester, Baud rate $>$ bit rate $\xrightarrow{\text{and}}$ twice as much
in NRZ, Baud rate = bit rate.

Lecture 5 Modulation in Wireless Networks

Suppose we have a sender with an antenna. He wants to send a signal.
How would he do that? A free-for-all is infeasible.

The government steps in.

They auction certain ranges of frequencies for explicit purposes.

Then even if different frequencies interfere, we can use a bandpass filter.

Some bands like WiFi are unlicensed, but then interference is an issue.

Say we are limited to $(f_0 \pm 10 \text{ MHz})$

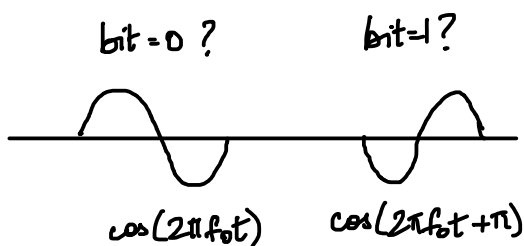
$$\cos(2\pi f_0 t) \xrightarrow{\text{F.T.}} \text{graph of a single impulse at } f_0$$

But doesn't convey any information.

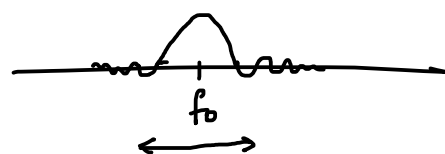
This is where modulation enters the picture.

Carrier wave: $\cos(2\pi f_0 t)$

$A \cos(2\pi f_0 t + \theta)$
 Amplitude \downarrow Frequency \downarrow Phase \rightarrow 3 types of modulation.
 (F.M. used in analog)



You can show that the Fourier transform looks something like



Width depends on how frequently you change phase.

\hookrightarrow Usually far more than just 1 wavelength

If we change less frequently, low rate of transmission but restricted to a narrower band.

Say $s(t)$ is transmitted and

$$r(t) = \sum_{i=1}^{\infty} a_i s(t - \tau_i) + n(t)$$

\nearrow due to reflections \nearrow noise
 \downarrow attenuation \downarrow delay \searrow if added,
 "additive white gaussian noise"
 (AWGN)

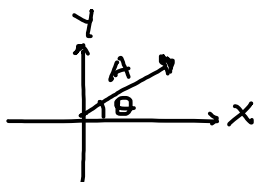
is the received signal.

(most general case)

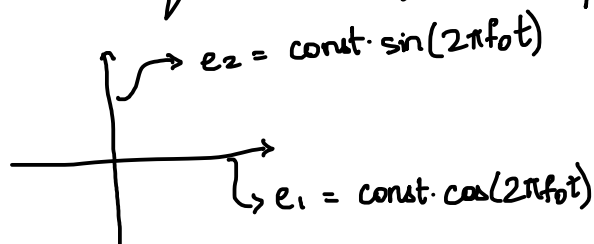
The **bit error rate** is the fraction of transmitted bits received erroneously. (interpreted)

We can think of signals in a vector space.

Focusing on frequency f_0 , we can represent a wave $A \cos(2\pi f_0 t - \theta)$ by the vector of magnitude A at angle θ . (in \mathbb{R}^2)



What would be the equivalent of the dot product here?



square-integrable

In these functional vector spaces, the inner product of functions $f, g: [0, T] \rightarrow \mathbb{R}$ is

$$\langle f, g \rangle = \int_0^T f(t)g(t) dt$$

If we send the signals over time $T = N/f_0$,

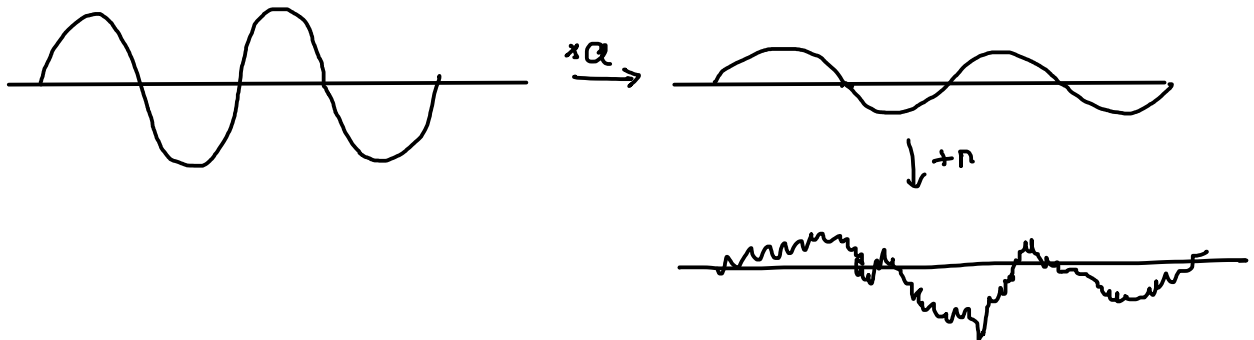
since $\langle e_1, e_1 \rangle = \langle e_2, e_2 \rangle = 1$,

$$e_1 = \sqrt{\frac{2f_0}{N}} \cos(2\pi f_0 t) \quad \text{and} \quad e_2 = \sqrt{\frac{2f_0}{N}} \sin(2\pi f_0 t)$$

and $\langle e_1, e_2 \rangle = 0$.

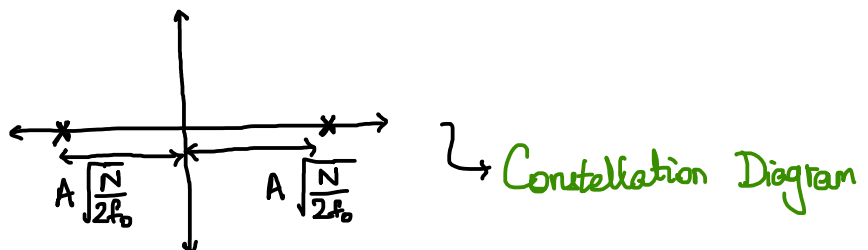
Suppose $r(t) = \underbrace{a \cdot s(t)}_{\text{attenuation}} + \underbrace{n(t)}_{\text{AWGN}}$

$$\begin{aligned} r_x &= a \langle s, e_1 \rangle + \underbrace{\langle n, e_1 \rangle}_{n_x} \\ r_y &= a \langle s, e_2 \rangle + \underbrace{\langle n, e_2 \rangle}_{n_y} \end{aligned}$$



$$g = A \cos(2\pi f_0 t - \theta) = \frac{A}{\sqrt{2f_0/N}} (\cos\theta e_1 + \sin\theta e_2)$$

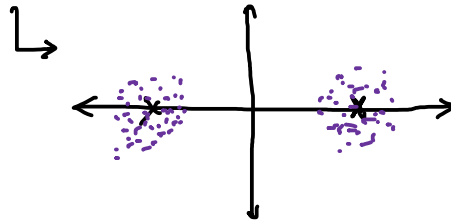
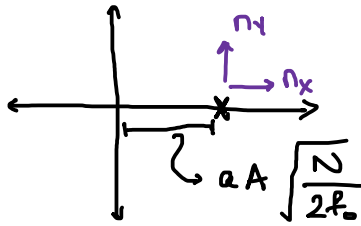
Suppose we transmit $s(t) = A \cos(2\pi f_0 t)$ if bit 0
 $s(t) = -A \cos(2\pi f_0 t)$ if bit 1
 for time $0 \leq t \leq T = \frac{N}{f_0}$.



Say bit 0, so $s(t)$. We receive $as(t) + n(t)$

$$r_x = as_x + n_x$$

AWGN n is such that n_x and n_y are iid Gaussian.



(If we transmit the same bit several times)
Looks like a constellation.

How do we do detection?

Say $r(t) = a s(t) + n(t)$

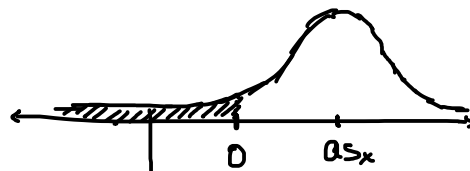
→ Calculate r_x and r_y .

→ Find the constellation point closest to (r_x, r_y) .

(In this example, just see if $r_x > 0$ or $r_x < 0$)

↓
bit 0

↓
bit 1

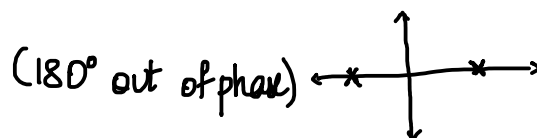


This tail probability is the probability of interpreting it incorrectly (if 0 is transmitted)

The **signal-to-noise ratio** (SNR) is the ratio of signal power to noise power.

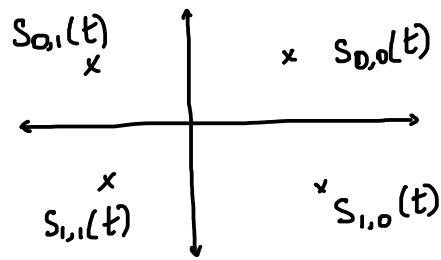
The above modulation scheme is known as **Binary Phase Shift Keying**.

(BPSK)



could switch 0 and 1 or even keep them on Y-axis.

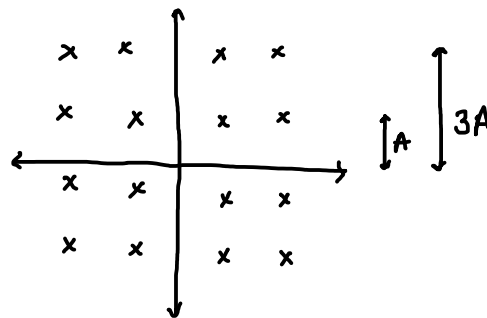
In Quadrature PSK (QPSK),



Can send two bits of data at a time.

Transmission / detection is nearly identical to BPSK.

In QAM-16,



Four bits of data at a time

QAM-256 has 8 bits of data.

What constellation do we use?

1. Allowed transmit power
2. Received signal power

Bit Error Rate is a function of SNR.

Say QPSK vs QAM-16 with same received signal power (and same attenuation).

↓ ↓

Slower Faster

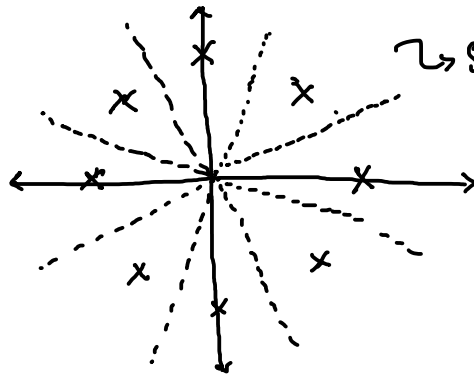
but lower BER but higher BER

For example, our WiFi card constantly sees the SNR and dynamically changes the scheme used.

Each of these x's in the constellation diagram is called a **symbol**

The transmitted signal is just the concatenation of the signals corresponding to each of the bits.

8-PSK is



↳ sectors according to which we decide what to interpret received signal as.

QAM-16's sector division is slightly trickier (what is it?)
(and not even uniform, in fact)

PSK is convenient because amplitude doesn't even enter the picture.