

Boar In Blue Linen

www.BrainLubeOnline.com

November 21, 2013

Abstract

This is a living L^AT_EX Print/Press Document Version of the Boar In Blue Linen Document; a tutorial on electronic/radio/computer wave and communications theory.

Originally compiled into L^AT_EX: 11/20/2013

Web Revised: 11/20/2013

L^AT_EX Revised: -/-/-

1 Wave Primer

1.1 How to read this

To use this tutorial: read every word, and click and read every link. I wrote this page as I learned the material, and that makes for the best tutorials, because this is the order that I learned the information. Just make it a project and step away from it, when you stop concentrating. Come back to it when you are rested and do not quit the project until you are done. For the course I followed excerpts from DATA COMMUNICATIONS AND NETWORKING, BEHROUZA. FOROUZAN W/ SOPHIA CHUNG FEGAN. A great book that I thoroughly recommend.

1.2 What are Waves

Wave to your computer! Data comes in waves. The math for basic electromagnetic waves are the same for any other type of waves, this makes the elementary stuff fairly simple. It is important to have a firm grasp on fundamental wave theory, it is necessary to understand these principles to fully master the methods of communication.

There are many great sites available to help you understand waves, the best I have found are located at the Khan Academy, check the videos out at: Introduction to Waves, Waves, and The Doppler Effect.

2 Data Transmission Using Signals

In the realm of data transmission there are two classic types: *Analog*, and *Digital*. Analog is a clock with a contentious second hand. Digital is your clock radio, where the numbers are displayed. Analog transmissions are infinitely divisible, digital transmissions have set, discrete increments. For the electrical transmission of signals it is sometimes necessary to convert digital signals to analog ones, or vise-versa.

2.1 General Signal Waves

A contentious wave that creates a noticeable pattern (by changing frequency or amplitude) is referred to as a *periodic* wave. The time measured that it takes to complete 1 complete cycle of that wave is called the wave's *period*. By contrast, a non-periodic (aperiodic) wave does not have a discernible pattern or cycle.

2.2 Periodic Signal Analog Waves

When we look at a sine wave we are looking at a graph of Time in the X domain, and Amplitude in the Y domain. The *peak amplitude* of a signal is the highest part of the wave. The *period* of the wave is how many seconds it takes to complete 1 cycle of the wave; or written another way, the time (in seconds) it takes from one peak on the X axis to the next. The *frequency* of a

wave is how many cycles per second the wave is completing; or written another way; frequency is how many periods in 1 second. Take a minute to think about that. $f=1/T$ where T (period) = time measured in seconds and f is frequency measured in Hz (1 Hertz = 1 cycle/second). Therefore period (T) may be calculated by $T=1/f$ because frequency and period are inverses of each other.

The *Hertz* measurement is seldom written out beyond 3 digits to the left of the decimal. Instead it is written as factors of 10^3 . 1 cycle per second = 1Hz, 1,000 cycles per second = 1kHz (Kilohertz, or 10^3 Hertz), 1,000,000 cycles/second = 1MHz (Megahertz, 10^6 Hertz), 1,000,000,000 cycles/second = 1GHz (Gigahertz, 10^9), and finally 1,000,000,000,000 cycles/second = 1THz (Terahertz, 10^{12}).

Frequency	1Hz	1kHz	1MHz	1GHz	1THz
Cycles per Second	1	1,000	1,000,000	1,000,000,000	1,000,000,000,000

Thus the relationship with period may be expressed as follows: a period(T) of 1 second(s) = 1Hz and, 1 Millisecond (ms, or 10^{-3} seconds) = 1kHz, 1 Microsecond (us, or 10^{-6} seconds) = 1MHz, 1 Nanosecond (ns, or 10^{-9} s) = 1GHz, and 1 Picosecond (ps, or 10^{-12} s) = 1 THz.

Period (in seconds)	1	.001	.000001	.000000001	.000000000001
Cycles/Second (Hz)	1	1,000	1,000,000	1,000,000,000	1,000,000,000,000

3 Frequency and Time Domains

If the transmission medium never changes it's state than it's frequency is zero. However, if the state of a transmission medium changes instantaneously, than it's frequency is infinite.

Phase is the configuration of a waveform relative to time. If the period of a wave is the measure of one cycle, than the phase is where you start measuring. For instance you can measure a wave from where the amplitude starts to rise, to where it starts to rise again, or, from it's highest point to the next. Where you start to measure (or view) is called the phase. Phase is measured in *degrees* or *radians*. 1 degree equals $2\pi/360$ radians (that's 2 times pi divided by 360). Inversely, 360 degrees equals 2π radians. This means that 1 rad (radian) = $360/2\pi$ degrees (57.295779513082325 degrees).

Here is a great site for explaining how to measure phase:

[HTTP://WWW.INDIANA.EDU/~EMUSIC/ACOUSTICS/PHASE.HTM](http://www.indiana.edu/~emusic/acoustics/phase.htm)

Imagine a circle moving counter-clockwise along the sin wave. To measure phase 0 degrees is where the rising amplitude at the center is. 90 degrees is the top peak of the wave descending. 180 degrees is the descending amplitude at the center of the wave form. 270 degrees is the lowest part ascending, than back to 360/0.

3.1 Wavelength

One "up hump" and one "down hump" is one cycle of a wave, we know this. We also know how to measure the time it takes to complete one cycle (period), further, we know how to count how many cycles occur in one second (frequency). Since we know the period, if we know how fast the wave is moving, we can also tell the wavelength. *Wavelength* (wavelength is represented with the lambda symbol: λ) is the distance a signal travels in 1 period. Or to look at it another way, λ is the length of the waveform. The *propagation* speed of the wave conduit or medium is the speed the wave travels through various materials. a sound wave through air at sea-level travels faster than one at a high altitude because the air is more dense. Likewise, a sound wave travels much faster under water than in air. Electrical signals travel at around the speed of light (c). So for electrical signals our formula can be $\lambda = c/f$.

This means that propagation speed * the period = the wavelength. Also because frequency is an inverse of the period, propagation speed / frequency also equals the wavelength. ($\lambda = c/f = c*\text{period}$).

The speed of light in a vacuum is $3*10^8$ meters per second ($c=3*10^8\text{m/s}$). The speed of an electrical signal through a wire is slightly slower in a wire. However, c is close enough for our purposes. A good example is the wavelength of what us humans perceive as green light, it has a 540nm (1 nm is $1*10^{-9}$ meters) wavelength. if $\lambda = c/f$ we know now that $(540 * 10^{-9}) = 3*10^8/f$, we can rearrange the formula to compute the frequency. To find f we write $3*10^8 / (540 * 10^{-9}) = f$

$3*10^8 = 300,000,000$ AND $(540 * 10^{-9}) = 0.00000054$ SO $300,000,000 / 0.00000054 = 555,555,555,555,555.5555555555555556$ OR 556THz! Thus the color Green has a frequency of 556THz.

Lets put this into conventional radio terms and lets calculate the wavelength of the 144-148MHz ham band. first lets find the center of that frequency delta (Δf), 148 minus 144 gives us 4MHz divided by 2 gives us 2MHz added back to 144 leaves us at 146MHz for the middle of that ham band. $\lambda = c/f$ so $\lambda = 3*10^8/146,000,000 = 2.054794520547945$ meters. This is why they call this the 2-meter band!

This can be illustrated with a software defined radio at this site:

[HTTP://WEBSDR.EWI.UTWENTE.NL:8901/](http://WEBSDR.EWI.UTWENTE.NL:8901/)

3.2 Composite Periodic Waveforms

A simple sine wave is relatively easily to visualize. But the wave does not have to be just a simple sine. The wave can be a composite of many different perturbations in the signal. First let's bring our attention back to this site: [HTTP://WWW.INDIANA.EDU/~EMUSIC/ACOUSTICS/PHASE.HTM](http://WWW.INDIANA.EDU/~EMUSIC/ACOUSTICS/PHASE.HTM). As you can see, a wave laid over top another wave can cause: *constructive interference*, or *destructive interference (phase cancellation)*. The waves can become quite complicated, and we can use this to extract information from the signal. for instance, since we know the beat frequency from the last example from the site above,

and we know the first tone (the *carrier wave* more on this later) we can extrapolate the other tones (frequencies) in the signal. We now need to go to this site: [HTTP://HYPERPHYSICS.PHY-ASTR.GSU.EDU/HBASE/AUDIO/FOURIER.HTML](http://HYPERPHYSICS.PHY-ASTR.GSU.EDU/HBASE/AUDIO/FOURIER.HTML) .

The idea here is counting the occurrences of frequencies within a range or set. This process is called *decomposition*. Counting beats is useful if all the frequencies in the composite wave are of the same amplitude but we can stuff more information in a wave than just that! We can also vary the amplitude of each different frequency of the wave.

In doing this we have a *fundamental frequency* otherwise known as the *first harmonic*. Thus another component frequency of the composite wave that is 3 times the fundamental frequency is the *third harmonic*, and 9f is the ninth harmonic frequency, and so on. These harmonics are always discrete, meaning there will never be a 2.7f harmonic. They will always be discrete spikes on the waveform.

3.3 Bandwidth

Bandwidth is the difference between the highest and lowest frequencies in a wave; it is the range of frequencies portrayed in a given wave. For instance, if a composite wave or signal is comprised of frequencies of 1250MHz (1.2GHz) and 2GHz, then the bandwidth is $2,000,000,000 - 1,250,000,000 = 750,000,000$ or 750MHz. It should be noted that it does not matter if this wave contains 1.3GHz and 1.7GHz signals also, we only care about the highest and lowest frequencies to discover B (bandwidth).

You've seen the sine, now on to...

4 Digital Signals

Digital signals have discrete states, in it's simplest form it could be a state named "1" for any positive voltage or "0" for the lack of a voltage. the name of the states is arbitrary, and the states could just as easily be represented by -1 volts and +7, just depending on what you need. A digital signal may have many different levels though.

It doesn't have to be just 2 signal levels however, there can be many different levels in a signal; perhaps -2 volts, -1 volts, 3 volts, and 5 volts.

In a signal with 2 levels, the signal carries 1 bit per level: 1 or 0.

In a signal with 4 levels, each level carries 2 bits (2-bit/level). (00, 01, 10, 11) Thus a signal with L levels carries $(\log_2 L)$ bits. An example would be that a signal with 8 levels has $\log_2 8 = 3$ bits per level ($3\text{bits}/L$).

4.1 Bitrate

Bitrate is the number of bits per second being sent. Described in *bps* (bits/second). If you need to download 1000 pages in 1 minute, where each page is 48 lines of 80 characters and each character is 8 bits, what is the connection speed bitrate

required? : $\rightarrow 8$ (bits) times 80 (characters) times 48 (lines per page) times 1000 (pages) = 30,720,000 or 30.72 megabits required to be transmitted in one minute. Divide 30,720,000 by 60 because there are 60 seconds in a minute: $30,720,000/60 = 512,000$ bits per second (bps) or 512kbps. Thus you need a 512kbps connection.

4.2 Bit Length

Bit length in a digital signal is tantamount to wavelength in an analog one. Bit length therefore is the length or distance 1 bit occupies in the wire or transmission medium. Bit length = propagation speed * bit duration (propagation speed in air or a wire is close to C the speed of light in a vacuum).

4.3 Bits in relation to Frequency

To discover the maximum bandwidth needed in a 2 level signal one would apply: bandwidth = $N/2$ where N is bitrate. The worst case is the most changes to the levels 10101010 or 01010101 the best case being 00000000 or 11111111. To understand way you must visualize the waveform of the signal. The most changes in a second are the maximum required frequency and the least changes required a second is the minimum frequency. Maximum frequency - Minimum frequency = Bandwidth. It should also be noted that 11111111 or 00000000 possesses a frequency attribute of 0Hz because it does not change.

4.4 Loss

Whenever you transmit a signal, it is subjected to impairment. This is because of attenuation, distortion, and noise. Attenuation is the loss of energy. This energy lost may be transformed into heat, light, or movement (such as vibration in coils). Signal loss or gain is measured in *decibels* (dB). The decibel is a comparative number, positive is gain and negative is loss. $\text{dB} = 10\log_{10} P_2/P_1$ where P_1 is the first test point or location and P_2 is the second test point or location. This is a formula for magnitude; it can work for voltage or power. Voltage being $20\log_{10} P_1/P_2$

The Wiki on logs is located here:

[HTTP://EN.WIKIPEDIA.ORG/WIKI/LOGARITHM](http://en.wikipedia.org/wiki/Logarithm)

Distortion is when the wave takes on a different shape. Noise is any unwanted signal imposed on the desired communication signal; noise causes distortion. The signal to noise ratio (*SNR*) describes how much of a signal is noise. $\text{SNR} = \text{average signal power} / \text{average noise power}$

Converted to dB is: $\text{SNR}_{\text{dB}} = 10\log_{10} \text{SNR}$

5 Data Rate Limits

5.1 Nyquist

The Nyquist bit rate describes the maximum theoretical bit rate of a noiseless channel. $\text{Bitrate} = 2 * \text{bandwidth} * \log_2 L$ where L is the number of signal levels used. So for example, to send 512kbps over a noiseless channel at 40kHz, how many levels does the signal need to possess?

$512,000 = 2 * 40,000 * \log_2 L$ $512,000 = 80,000 * \log_2 L$ $6.4 = \log_2 L$ $2^{6.4} = 84.4$ levels (84.44850628946526 needs 85 levels)

5.2 Shannon

The Shannon capacity describes the highest data rate possible in a noisy channel. It is measured in band capacity.

$$\text{Capacity} = \text{bandwidth} * \log_2(1 + \text{SNR})$$

The example given in the book is this: A 1MHz bandwidth channel with an SNR of 63 → What is the appropriate bitrate and number of signal levels?

First they use Shannon Capacity: $C = B \log_2(1 + \text{SNR})$ $10^6 \log_2(1 + 63)$ $10^6 \log_2 64$ 6Mbps

Then Nyquist is applied to find the signal levels, 4Mbps is used to make sure we are well within our 6Mbps capacity.

$$4\text{Mbps} = 2 * 1\text{MHz} * \log_2 L$$
 $L = 4$ (4 levels)

5.3 Performance

The bandwidth discussed here is a frequency bandwidth. When dealing with communication services it is usually a bitrate bandwidth. The latter being a networking context, and the first being in a communications context. It is also important to discuss *throughput*. The term *bandwidth* describes the potential bit rate of a transmission medium. The term *throughput* describes the **actual** bitrate measured. For example a 5Mbps bandwidth connection is transferring 5,000 frames/minute where each frame is 10,000 bits.

$5,000 \text{ frames} * 10,000 \text{ bits} = 50,000,000 \text{ bits per minute} / 60 = 833,333 \text{ bits/second} = 833\text{Kbps!}$

The line with a bandwidth of 5mbps only has a throughput of 1/6th of the bandwidth. There is an issue somewhere.

5.4 Latency

Latency is the delay, or how long it takes a message to **completely** arrive at it's destination. Counted from the first bit sent to the last bit received. Latency is a composite of 4 elements: *propagation time*, *transmission time*, *queuing time*, and *processing delay*. To get the latency of a transmission you simply add these 4 elements together. Propagation time we have already discussed:

$$\text{propagation time} = \text{distance} / \text{propagation speed}$$

This is the time it takes the first part (change in wave) of the first bit to travel from the transmitter to the receiver. C (the speed of light in a vacuum) is 3×10^8 m/s (meters per second), in a cable it is 2.4×10^8 m/s. So let's say we had a cable around the equator of the Earth and we wanted to send a bit down it and turn right around and receive it. How long would it take to receive it? Well the Earth at the equator is 24,901 miles long (circumference at the earth's greatest point) and 1 mile = 1.60934 kilometers. So that's 40,075 km circumference * 1000 (to get us to 40,074,000 meters) / 2.4×10^8 (the propagation speed of electromagnetic energy in a wire) gives us: .166975 seconds -> propagation time = 166ms! Fast!!!

Transmission time is the time just between the first bit sent and the last bit sent. The last bit travels at the same speed that the first bit does but it has to wait for all the bits in between to be sent first.

So transmission time = message size / bandwidth

Queuing time is simply the time the message must be held until it can be processed, and of course processing time is how long it takes to actually process the message. This will become important when we get into data compression and error correction. However it is also important to think about the input buffers of these systems. You can fill a line with data that can keep coming into the receiver after the receiver's buffer is filled and you have stopped transmitting (because the signals are still traveling in the medium). Thus the number of bits that it takes to fill a link is the *bandwidth delay product*.

The bandwidth delay product or volume = bandwidth * delay and line delay product = propagation time * bit rate

5.5 Elements of Communication

There are *signal elements* and there are *data elements*, they are different and related.

Visit this page, Signal Elements vs. Data Elements:

[HTTP://SPARC.NFU.EDU.TW/~LYC/INTRO_NET/CH04.PDF](http://SPARC.NFU.EDU.TW/~LYC/INTRO_NET/CH04.PDF)

As you can see, it might take 1 signal element to describe one data element or it might even take many in certain situations. This is defined by ratio r

If $r = 1$ that equals 1 data element per 1 signal element, if $r = 1/2$ that is 1 data element per 2 signal elements, and if $r = 2$ that would be 2 data elements for every 1 signal element. $4/3r = 4$ data elements to 3 signal elements. Thus *data rate* is the number of data elements transmitted in one second (bps). *Signal rate* is the number of signal elements sent in 1 second. Signal rate is measured in *baud*. Signal rate is synonymous with pulse rate, modulation rate, or baud rate. The ultimate goal being to increase the data rate while lowering signal rate. Baud is symbols per second.

$S = c * N * 1/r$ Where S is baud, c is a corrections constant, N is the data rate, and r is the data to signal element ratio.

Therefore the goal is a higher bit rate at a lower baud. Baud is also related to bandwidth. The minimum bandwidth required is $B_{\min} = C * N * 1/r$ thus the maximum bit rate is $N_{\max} = 1/c * B * r$

5.6 Digital Amplitude

A *line coding* scheme is a method used for carrying data (methods with signals and signal elements). The average power of the transmission is factored by the receiver, this average is called the *baseline*. If many consecutive 1's or 0's are encountered, this may cause a large drift in the baseline, this is called *baseline wandering*. When this happens, the receiver will not decode (the message) correctly. There are also DC components that you have to watch out for. The idea here is that devices like transformers **only** work with AC power so if the receiver gets too many signals that are of a single polarity (positive or negative, or of no change at all) the line will be effectively DC. There are different line coding schemes that consider this and effectively deal with it. There is also the subject of self-synchronization. The clocks of both the transmitter and the receiver must be the same, and they must be in the correct phase of each other. Good line coding schemes also provide for this by sending sync pulses to align the clocks. If the receiver is slower by just 1 tenth of a percent, then a 1mbps connection is receiving 10kbps more worth of false data, then is being sent. That's a bunch.

6 Line Coding

Even though error detection is primarily done in code, there are a few tricks that can be done at the line level. There are also a few ways to increase the signal's immunity to noise and interference by the way it is line coded. It should also be mentioned that the more levels, or the more complex the line coding scheme, the more expensive the system will be. There is also a technological limit, but it is always getting better.

6.1 Unipolar Scheme

NRZ Unipolar Non Return to Zero is a line coding scheme where 1 voltage defines 1 bit. For example a positive voltage could define bit "1" and a voltage of 0 could define bit "0". The power required for this mode is very high when compared to other modes and comes out to $1/2 V^2$. Because of this, this method is not often used.

6.2 Polar Schemes

Polar NRZ and *NRZ-I*, I for inverted, switches from a 0 to a 1 or a 1 from a 0 if the voltage **changes**. Thus if there is no change (high or low) the bits stay of the same kind. It should also be mentioned that these line coding methods have a baud rate that comes out to around $N/2$ (average). It should also be understood that these methods have a DC component problem since they have no negative voltages they will not work with AC systems.

6.3 RZ Encoding

Return to zero schemes add a negative value component to the equation. This makes it usable in an AC (telephone) system and, most importantly takes care of synchronization and clock creep. In RZ systems, the signal goes to zero in the middle of each clock cycle (the middle of each bit). A negative voltage = bit "0", a positive voltage = bit "1". This is an expensive and inefficient system and is not often used anymore.

6.4 Biphase

In *manchester* RZ and NRZ-I are both components in the manchester encoding method. The voltage will always move from positive to negative within one cycle, but the phase will change by 180 degrees if it is conveying a "1". Thus the clock (receivers clock) knows where to start counting every time the voltage changes and it makes no difference how many consecutive "1's" or "0's" are sent. *Differential manchester* works the same way but only when the signal changes polarity does the bit invert. Same as NRZ-I, it should also be mentioned that with these schemes baseline wandering is also not a problem. However, it does take 2 times the bandwidth of NRZ methods.

6.5 Bipolar Encoding

Bipolar encoding A.K.A. multi-level binary, same as manchester they have a negative voltage, a positive voltage, and 0. In this scheme one bit value (data element) is attributed to the zero state (0 volts), while the other value alternates between negative and positive. *Alternate mark inversion* is one of these systems. 0 voltage = bit "0" and bit "1" alternates between positive and negative voltages. The name comes from telegraphy (think morse code) where "mark" means "1". *pseudoternary* is the inverse of AMI where 1 in no voltage and the 0 bit alternates between positive and negative. These = 1/2N. The advantage of this: even with a long string of 1's the signal never turns into a DC current. 0's are another story, tackled later.

6.6 Multilevel Systems

To get more data with less bandwidth, one way to do it is to add more signal levels to your transmission scheme. You want more data/ baud (data per baud is key). 2 distinct data elements 1 and 0 (represented by M) make the number of patterns possible 2^M (00, 01, 10, 11). M is the length of the pattern, and n is the length of the signal. 2^M can equal L^n in a 1:1 ratio of encoding. With $L = 4$ you could have 2^2 combinations.

7 Analog to Digital Conversion

7.1 PCM

PCM stands for *pulse code modulation*. This is a system to convert analog signals into digital ones. For PCM you must sample the signal first. *Sampling* can be done at different T_s time, this is the sample interval or period. The sampling rate (frequency) is the inverse of the period (f_s). The sampling rate needs to be at least twice as highest frequency in the signal. Sampling is also called *pulse amplitude modulation*. This is the process of measuring the amplitude of a signal at various time intervals.

Nyquist sampling: $f_s = 2 f$

Oversampling is anything over that: $f_s = 4 f$ for example.

Visit this link on Signal Sampling:

[HTTP://WWW.CS.CF.AC.UK/DAVE/MULTIMEDIA/NODE149.HTML](http://www.cs.cf.ac.uk/Dave/Multimedia/node149.html)

Undersampling is, you guessed it: anything less than nyquist and is not enough to accurately describe a wave.

7.2 Quantization

The next step in the PCM conversion is *quantization*. The raw amplitude data at each time interval needs to be processed to be transmitted. Because the data at each point could be of infinite resolution. (.0000001 lets say) So we get the maximum and minimum amplitudes in the signal and divide that by the number of levels we wish to use (the more levels the more accurate the description of the wave) to give us our delta per level: $\Delta = (V_{\max} - V_{\min}) / L$

Encoding this is use converting this quantized data into binary for transmission and is so simple it does not need to be discussed. If you do not understand binary, follow the link at the "Global communication with the Internet" section, on binary and start there.

Because of this process we end up with a needed bitrate = sampling rate(f_s) * bits per sample(N_B)

We can also describe minimum required bandwidth by: $B_{\min} = N/2 * \log_2 L$ Hz.

7.3 Delta Modulation

Δ Modulation is a much simpler encoding and quantization technique than PCM and could come in handy if you're dealing with a low power device such as a micro-controller. A stream of 1's and 0's simply tell the receiver (demodulator) 1 to ascend to the next delta or 0 to descend to the next lower delta. (thus moving a speaker or etc.)

8 Analog Transmissions

8.1 ASK

In *amplitude key shifting*, as the name implies, frequency and phase are constant while the amplitude is varied. At it's simplest implementation there is no signal for bit 0 and a signal amplitude of a frequency for bit 1 or vise versa. this is referred to as on-off keying (OOK). There may be namely different (multilevel) amplitudes keyed to increase bitrate.

8.2 FSK

The next system of transmission is *frequency shift keying*. FSK is where one frequency is bit 0 and another is bit 1 (or bit 01010 depending on how many levels there are in the mode). To calculate the minimum bandwidth required for binary FSK: $B = (1+d) * S + 2\Delta f$

This leads us to multilevel FSK (same math), and if it were infinite in level Δ it would be the same as analog FM radio!!! Because instead of shifting frequencies it is infinitely modulating them, thus *Frequency Modulation*.

8.3 PSK

Phase shift keying is, at it's simplest, where amplitude and frequency remain constant but the phase is shifted 180 degrees for a "1" and back 180 degrees for a 0 bit. There is a multilevel scheme for PSK as well, for instance in *quadrature PSK (QPKS)* two bits can be sent at the same time.

For example:

Circle	45	135	225	315
Phase (in degrees)	45	135	-45	-135
Bit	00	01	11	10

You could in practice be dealing with 16-QAM where there is a Δ of 22.5 degrees each level. How many bits per level with this?

8.4 Radio Propagation

In the electromagnetic realm, frequencies up to 300GHz are considered radio. 300GHz to 400THz is considered *infrared* and from 400THz to 900THz is called "light" as it is the human visible range of the spectrum. The radio band delta may further be broken down because of the properties of the earth and our solar system. The atmosphere of the earth creates a layer that certain radio frequencies may bounce off of. The magnetization and electrical forces of the earth and sun also play a part in radio propagation. Further, The white noise you see on an off TV channel my be RF noise as old as around the time the universe was created!

There are 3 types of radio propagation: *ground propagation*, *sky propagation*, and *line-of-sight propagation*. Frequencies up to 2MHz travel on the ground. Frequencies from 2MHz to 30MHz travel by reflecting off of the ionosphere (atmospheric layers) and frequencies about 30MHz travel by line-of-sight.

9 Error Detection

9.1 Errors

There are many ways an error may be created. Interference such as electromagnetic anomalies like lightning may change the shape of the waveform, turning 0's to 1's, etceteras. A *single bit error* is when only 1 bit of a data packet is effected. This is typically rare and errors will most likely occur in *burst errors*, where a string of bad information is received. There are certain methods of redundancy that may be put into place to combat some errors.

9.2 Detection vs. Correction

Error detection is the process of knowing/finding errors that may have occurred. *Error correction* is the process of getting the correct information back where the error was, in the data stream. *Forward error correction* is where the receiver tries to reconstruct the error laden transmission. Correction by *retransmission* is where the portion of data found to be erroneous is retransmitted.

9.3 Parity

To detect errors, simple methods such as *parity* may be used. Here is a link on the serial protocol with some parity information:

[HTTP://WWW.BRAINLUBEONLINE.COM/SERIAL.PDF](http://www.brainlubeonline.com/serial.pdf) .

Parity bits are appended to each packet sent, and they denote whether the data packet is odd or even. For example if you set the parity to odd, then the 7-bit packet 0110111 would have a parity bit of 0 (because it is already odd) appended to it for a new 8-bit packet 01101110. If the packet to be sent was 0110110 then the transmitter would append a parity bit of 1 to make the 1 count odd: 01101101 This way the receiver knows that if it does not get an odd count, an error has occurred. This works great for single bit errors but what if 00000111 was sent (binary 3 + 1 parity bit to keep the count odd) but this was received: 00011111 (a burst error where the binary 4 and 8 place have been flipped by interference). Since the parity bit still works out to odd the receiver sees only good data but an error as slipped by. This simple parity is no good for bust errors, which are the brunt of errors encountered. We can attack this problem by calculating the *hamming distance*. Hamming distance is simply counting the absolute value of the difference between the amount of 1's in the first packet and the next packet continuously. This can be done with an XOR (exclusive or) operation with two packets:

Packet Size	5-bit	5-bit	3-bit
Arbitrary Packet First	10101	11001	000
Arbitrary Packet Next	11110	10011	011
XOR Product	01011	01010	011
(1-Count) Hamming Distance Base ₁₀	3	2	2

So by sending a packet with the hamming distance we can catch some of the burst errors, but not all of them. The receiver may still receive an error with the same hamming distance of the of the correct packet. Further, this method must transmit a great deal of information taking much of the time share needed for the real data to be transmitted. Enter *two-dimensional parity checking*:

The Parity is set to *even* for this example:

2-dimensional Parity Checking	Row	Row	Row	Row	Row	Row	Row	Row	Row Parity
Packet ₁	1	0	1	1	1	1	0	1	
Packet ₂	0	1	1	0	1	0	1	0	
Packet ₃	1	1	1	0	1	0	1	1	
Packet ₄	0	0	1	1	0	1	1	0	
Column Parity	0	0	0	0	1	0	1	0	

With this method you can send less error checking data per packet (or set of packets). In some cases the computer can find the exact bit that is incorrect. Play with different scenarios though, you will find that there are still cases that a burst error can slip by undetected.

10 Network Topology

To get us started, lets inject some basic communications nomenclature. Communication in simplex mode means that the communication is unidirectional. It can only travel from transmitter to receiver. Like your FM stereo at home. Communication in *half-duplex* mode describes communication between transceivers, but not at the same time. Like citizen band walkie-talkies, push to talk, wait to listen, but not at the same time. Transmissions in *full-duplex* (also refereed to as simply duplex) describes communications between transceivers that can transmit and receive simultaneously (such as a telephone).

When a set of transceivers, receivers, and or transmitters are linked together, this as a whole is usually referred to as a *network*. The elements that comprise the network such as a computer (a transceiver because it transmits and receives), or a printer, are referred to as *nodes*. There are 4 common physical topologies for computer networks: *Mesh*, *Star*, *Bus*, and *Ring*.

10.1 Mesh

Common to computer networking, there are 2 types of connections: *point-to-point* and *multipoint*. A point-to-point connection is a dedicated link of wire

or radio between 2 devices. These transmissions can be one way or two way communications, simplex, half-duplex, or full-duplex in nature. Multipoint is where many devices share a single link.

In a mesh topology network, each device on the network has a dedicated point-to-point link to every other device on the network. This means that to connect each device (node) to every other node you need $n(n-1)$ links. This is how this works, imagine there are five people, they all need to shake hands. Number 5 needs to shake hands with 4, 3, 2, and 1. Number 4 however, needs only shake hands with 3, 2, and 1... And so on. This math gives on like for uploads and one link for downloads. If you are dealing with a duplex transmission conduit (one line connects node to node rather than 2) the formula is $n(n-1)/2$. This can be compared to the handshake example by visualizing that if person 4 initiates a handshake with person 5, than person 5 does not need to initiate a handshake with person 4.

10.2 Star

In a star topology, each node on the network is connected to a central device such as a hub through a dedicated point-to-point link. This topology will be discussed further later.

10.3 Bus and Ring

In bus and ring topologies, each node is connected to a single multi-point link. This topology will also be discussed further later.

11 Global Communication with the Internet

Many LANs use a 48-bit (6-byte) physical address. It is usually viewed as a colon delimited set of 12 hexadecimal digits, that are two digits in length.

02:3A:1F:BB:2C:E7 If you need a tutorial on hexadecimal or binary notation go here:

[HTTP://WWW.BRAINLUBEONLINE.COM/COUNTING_ON_BINARY/FRAMER.HTM](http://WWW.BRAINLUBEONLINE.COM/COUNTING_ON_BINARY/FRAMER.HTM)

The physical address is not to be confused with the logical address (usually a 32-bit number set, this will soon be increased), we will go over both of these later. There is also another important number called a port address. This tells various equipment what type of data is being received.