University of Pittsburgh

ECE 1150: Computer Networks

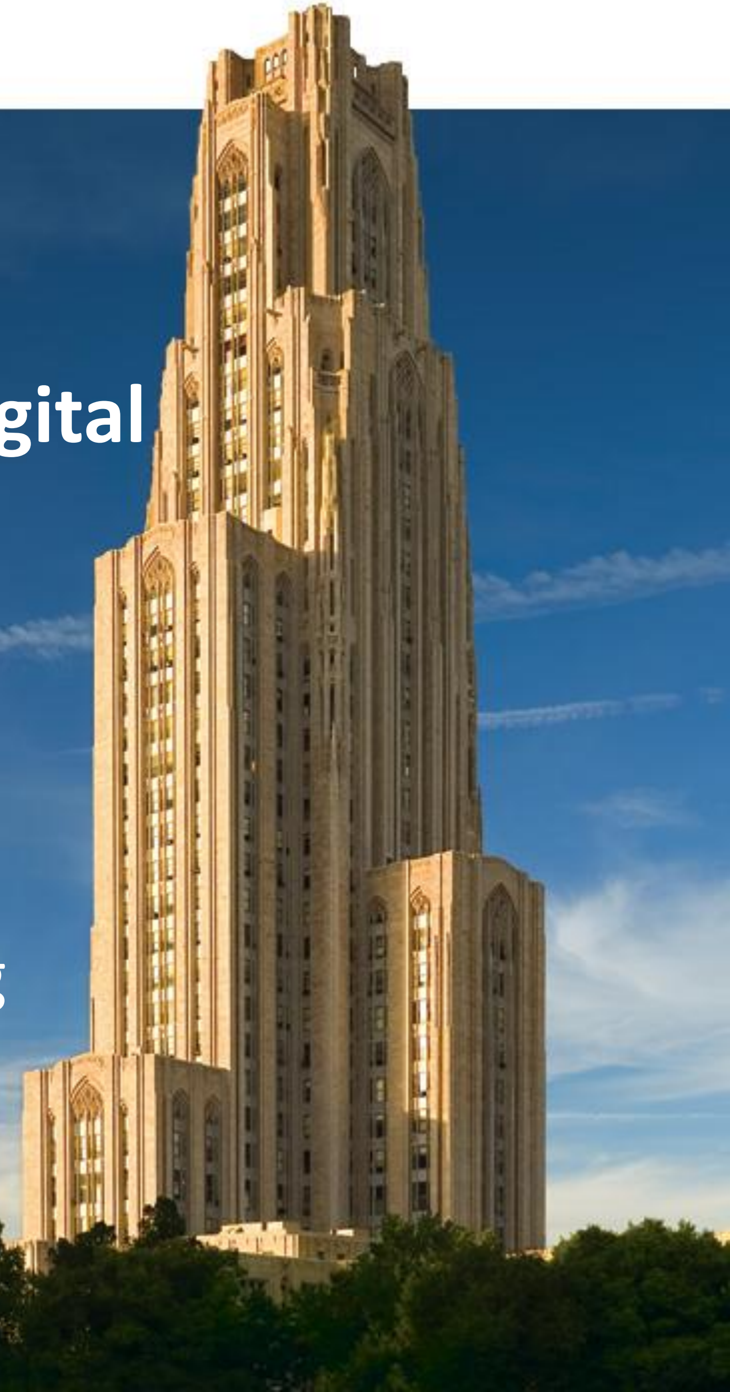
Physical Layer – Processing for Digital Transmission

Mai Abdelhakim, PhD

ECE Department

Swanson School of Engineering

University of Pittsburgh



Previous Units

- Transmission medium/Channel
- Signals
- Impairments of channels on signal
  - Losses (attenuation, path loss)
  - Noise
- Channel capacity

Objectives of This Unit

- Describe main processing in the physical layer
- Why we need **encoding**
- Why we need **sampling**
- What is **quantization**
- **Baseband line codes**
- Explain **modulation** (more about it next unit)

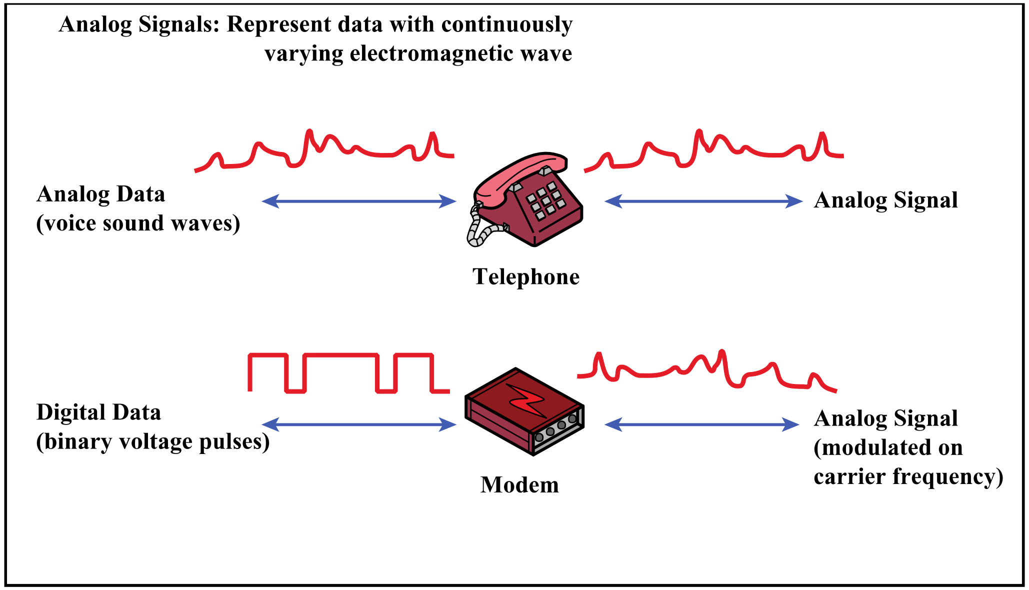
References:

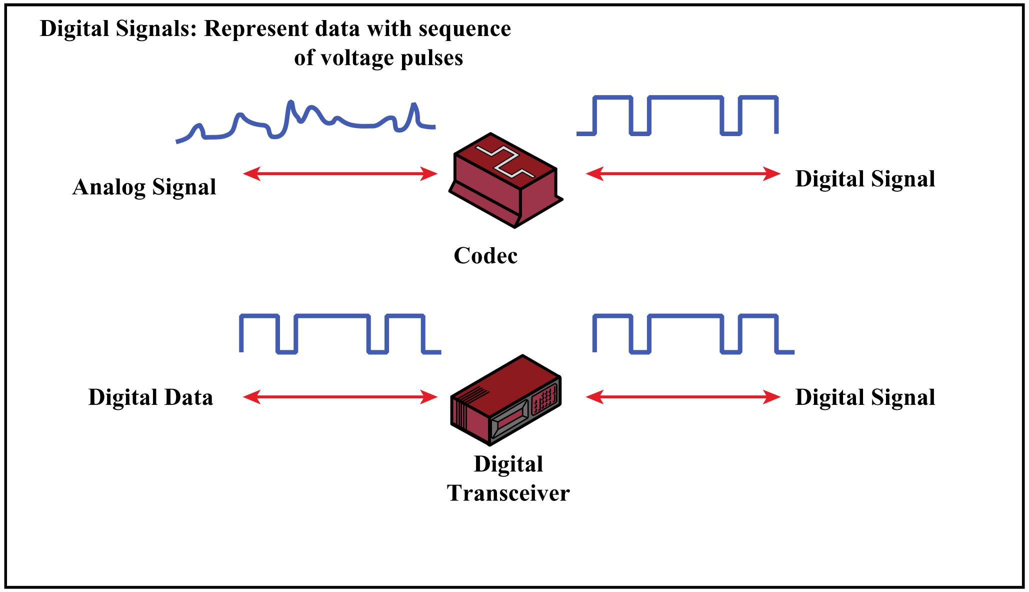
- Textbook, Agrawal, chapter 2
- Fitzgerald et al., Business Data Communication & Networking, Chapter 3

Origins of Data

- Analog data: come from sensors of various kinds (e.g., microphones)

- Digital data: come from computers and digital devices of various kinds





Analog and Digital Signaling of Analog and Digital Data

Data and Transmission

Data is analog (from microphones) or digital (file on computer)

Data type

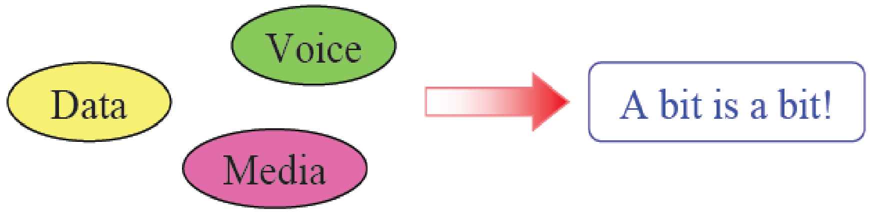
Transmission type = Signal type	Analog	Digital
Analog	Modulation	Modulation (Modems)
Digital	Signal Conversion (CODEC), PCM	Line Codes

In this unit, we will talk about digital transmission of digital and analog data

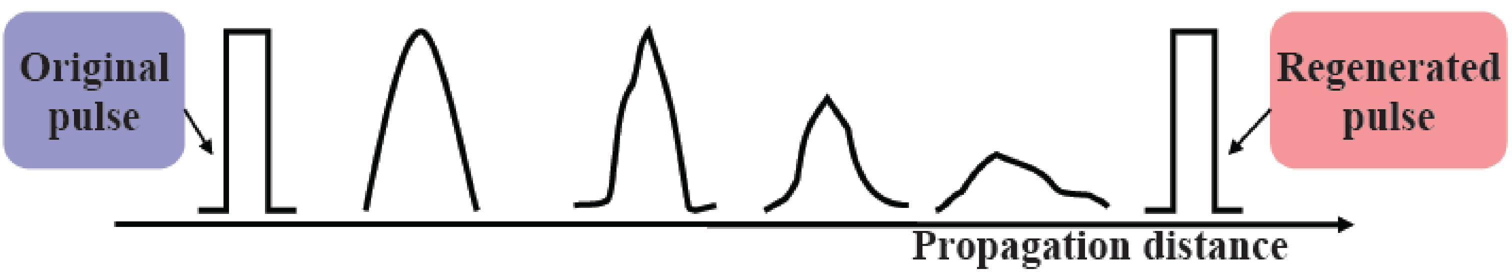
- Digital signal for analog of digital data

Advantages of Digital Systems

- Different kinds of information are treated the same way

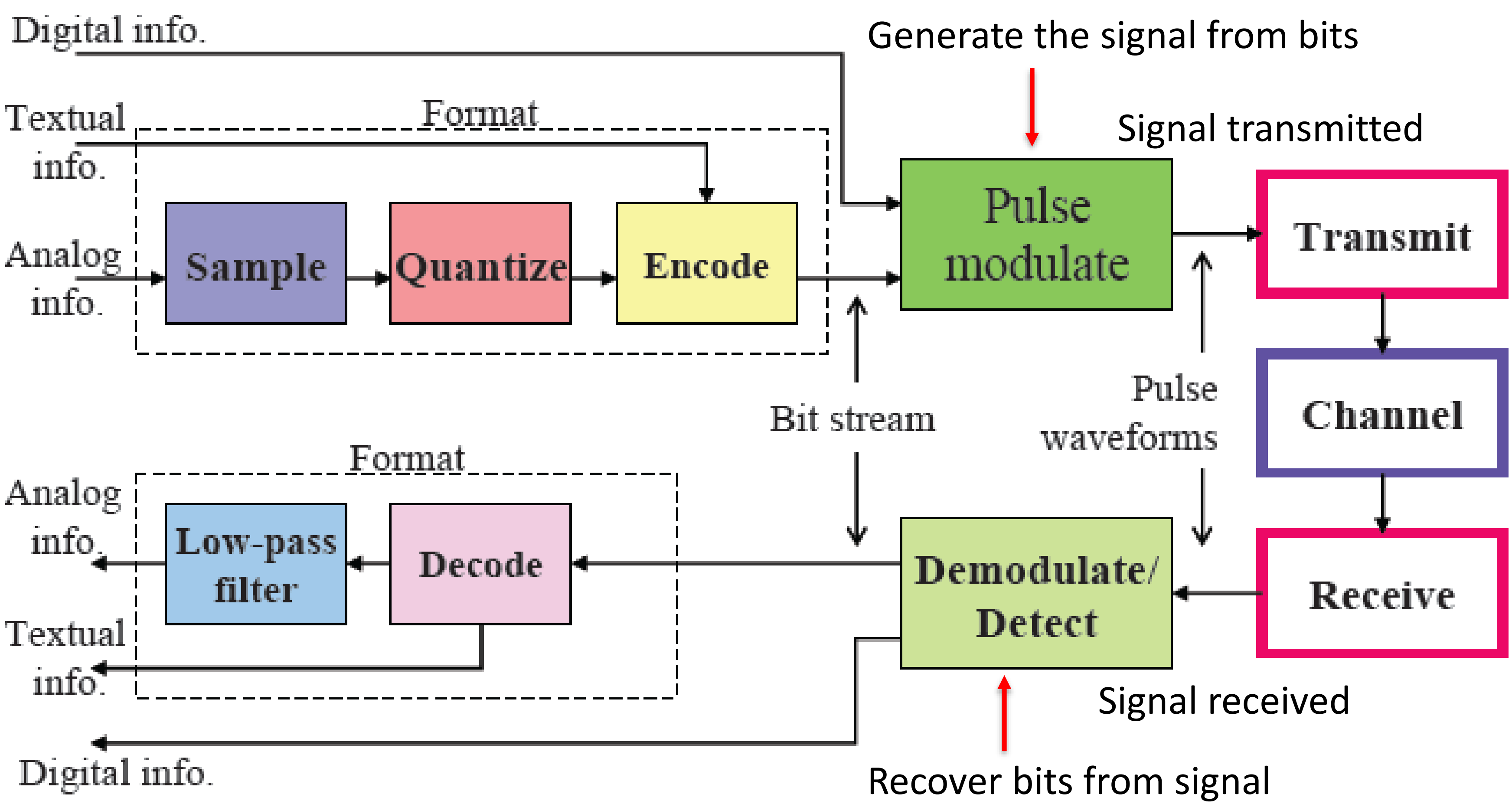


- More immune to noise.
  - Efficient regeneration of signal (can be done with repeaters)



- Advances in digital signal processing and coding makes digital transmission more efficient. E.g. data compression

Typical Baseband Digital Communication System – Physical Layer Processing



Encoding

- Problem: How can we send text? how to **ensure that the sender and receiver understand** messages?
  - **Coding scheme** is needed to ensure sender and receiver understand messages
  - Examples of coding schemes: ASCII, Unicode

Encoding

Mapping

- **Object set**
  - Alphanumeric characters
    - Lower case: a-z
    - Upper case: A-Z
    - Numbers: 0-9
  - Special characters (`~!@#$%^&*()-_=+\|{[]}:;"",<.>/?)
  - Control characters

- **Code set**
  - Binary codes: Sequence of {0,1}
  - Can be anything in principle
- Code types
  - Fixed Length codes (e.g., ASCII)
  - Variable length codes (e.g., Morse)

ASCII Encoding

- ASCII stands for: **American Standard Code for Information Interchange**
  - **Developed** by the **American National Standards Institute** (**ANSI**)
- Convert characters to binary
- A character is represented by a group of bits (8 bits)
  - Can represent $2^8 = 256$ characters
- Used in most microcontrollers
- http://www.asciitable.com

Letter	Binary
h	01101000
e	01100101
l	01101100
o	01101111

Example Using 8-bit ASCII

Each character is mapped to 8 bits with ASCII

Jones

J	o	n	e	s
01001010	01101111	01101110	01100101	01110011

Unicode Encoding

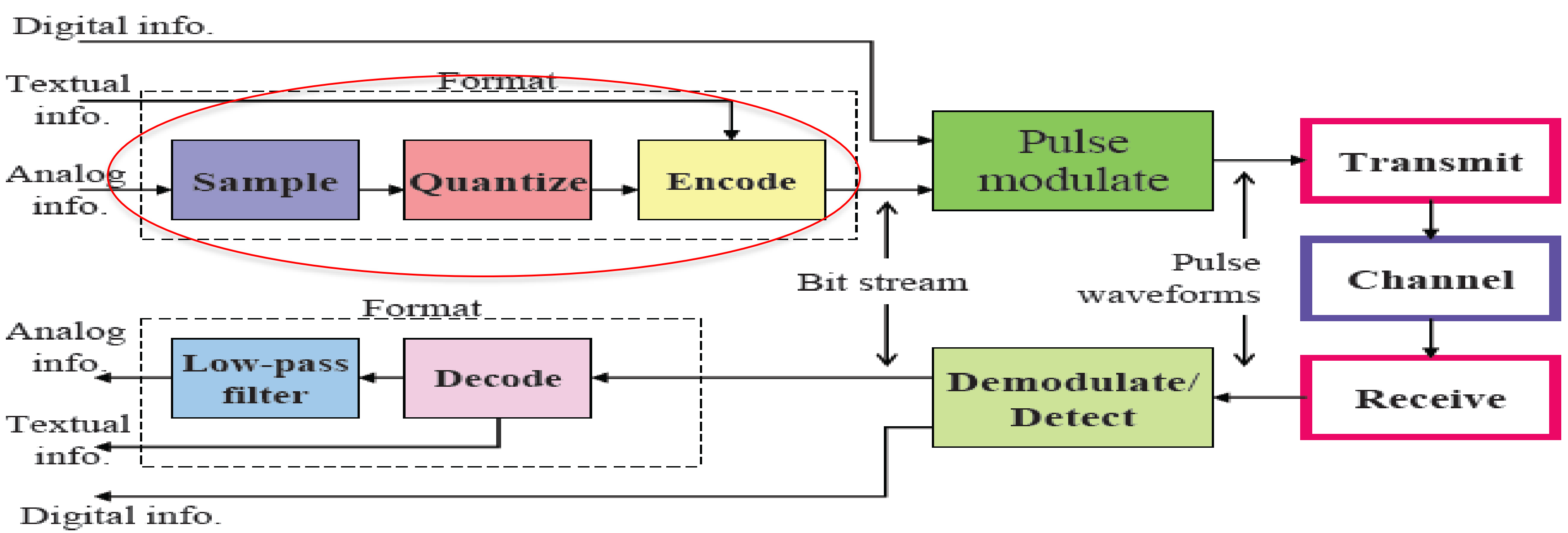
- Many versions of Unicode
  - UTF-16: use 16 bits per character
    (UTF: Unicode Transformation Format)
    - How many characters can UTF-16 represent? What about UTF-32?
- Used in operating systems, e.g. Windows
- Represent characters in almost all languages
  - Including over 75,000 Chinese characters
  - Incudes other characters
    - Greek characters α – ω

Analog Transmission over Digital System

- What about if the data is analog?
  - To transmit it with a digital system we need to first convert it into digital
  - This is called **Analog to Digital conversion**
  - CoDec (coder/decoder): Analog data to digital format

Analog to Digital

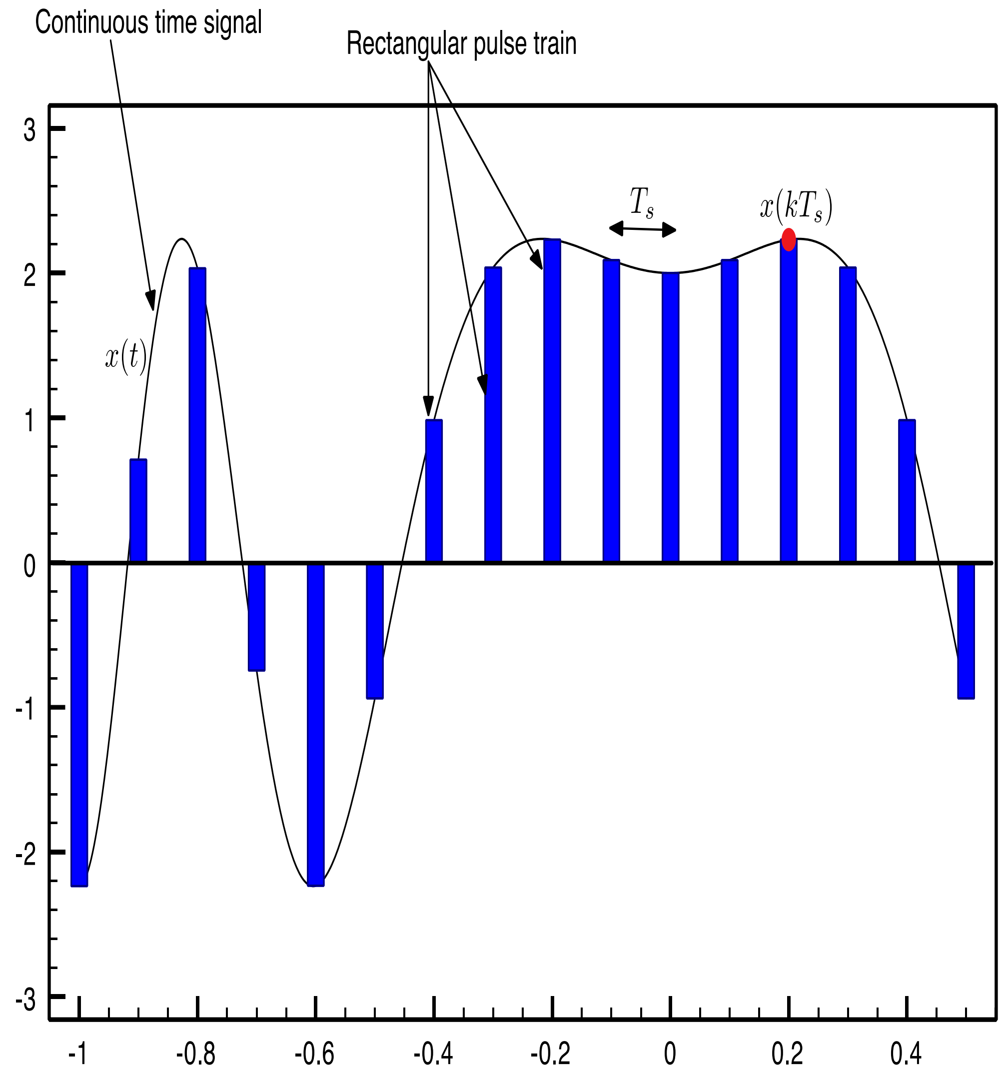
- Analog to Digital Conversion is made over steps
  - Sampling
  - Quantization
  - Encode



Sampling

- Sample the analog signal at regular time intervals
  - Ts = sampling time
  - F_s= sampling rate= 1/(Ts)
    - Also called sampling frequency

- Take a sample every 1/Fs seconds of the analog signal



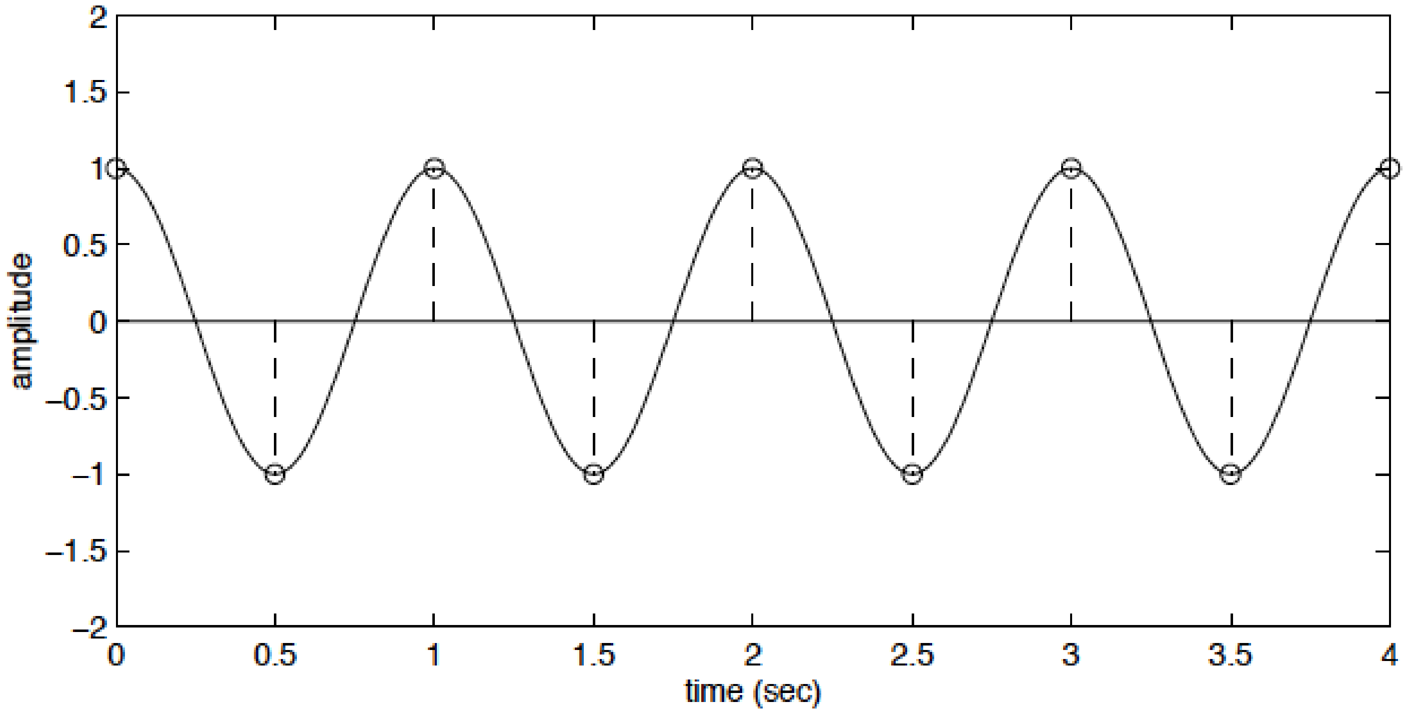
Sampling Theorem

- **Nyquist's Theorem:** Signal must be sampled at least at a rate that is **twice** the **maximum frequency** component of the signal

  - If F_m is the maximum frequency component in a signal

  - To capture variation in signal, sampling frequency (F_s) must be **$F_s \geq 2\ F_m$**

Examples on Sampling

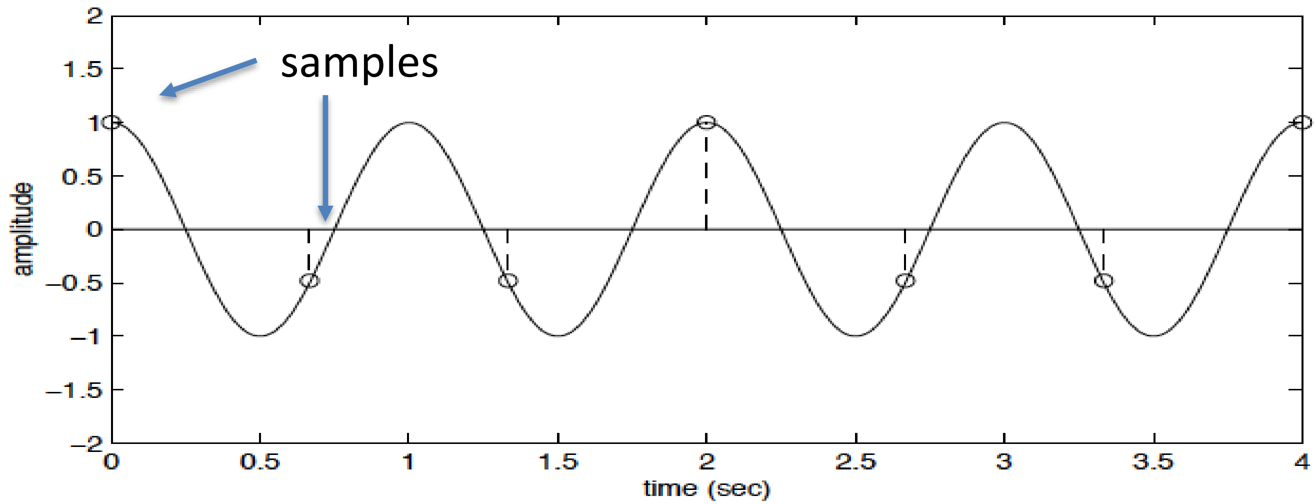


Signal has a frequency 1 Hz and is sampled at 2 Hz (sample every ½ seconds).

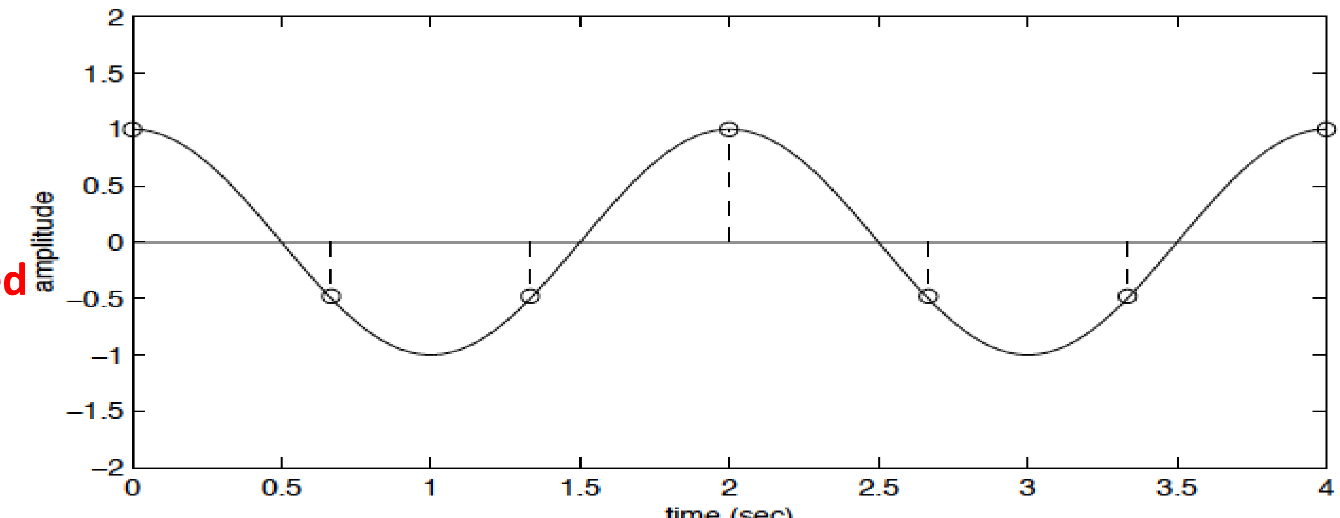
Variations can be captured at the receiver (since Fs = 2Fm)

What if the sampling frequency is less than twice the maximum frequency $F_s < 2 F_m$, e.g. Fs= 1.5 Hz?

Sampled signal



Signal recovered from samples



Recovered signal is distorted version of the original signal

<u>This is known as Aliasing</u>: Signal is misidentified when sampled at a rate lower than twice its maximum frequency

Tophat

Sampling

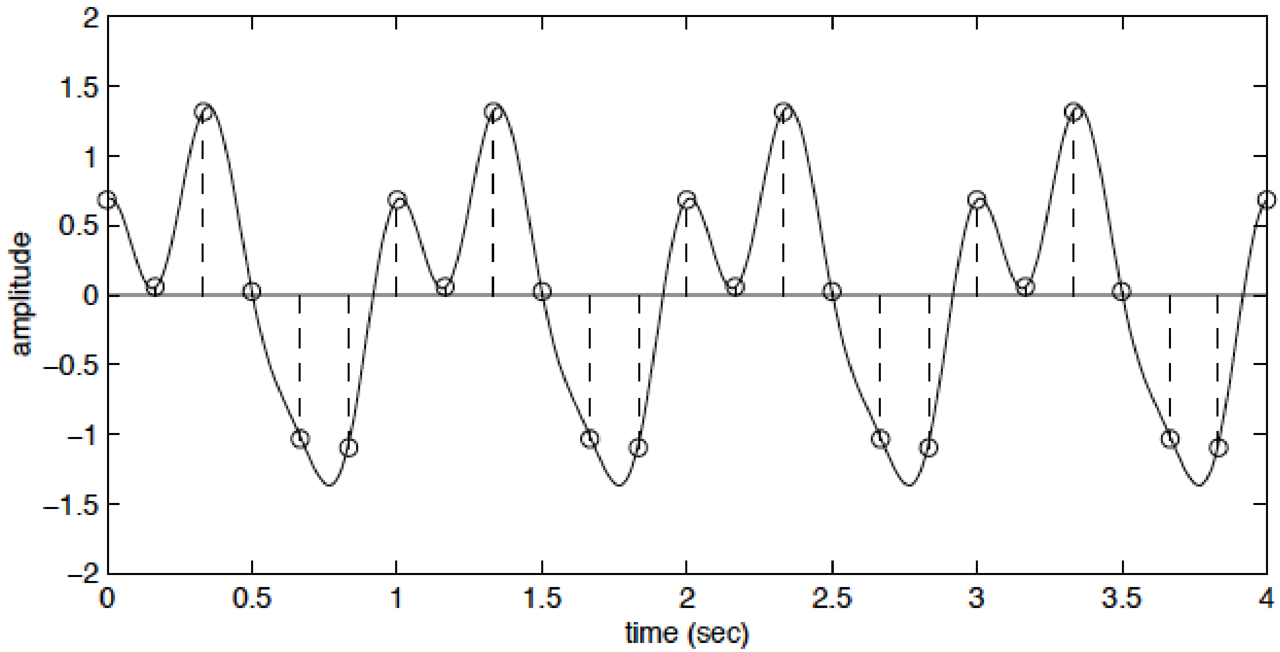
To avoid aliasing, a signal of $\cos(20\pi t)$ should be sampled at a rate no less than

A	40 samples/sec
B	20 samples/sec
C	10 samples/sec

Example

- The frequency components of a signal are at: 1Hz, 2Hz, 3 Hz (obtained by Fourier series)
- What is the minimum sampling frequency?

$$Fs=2F_m= 6\ Hz$$



Quantization

Quantization

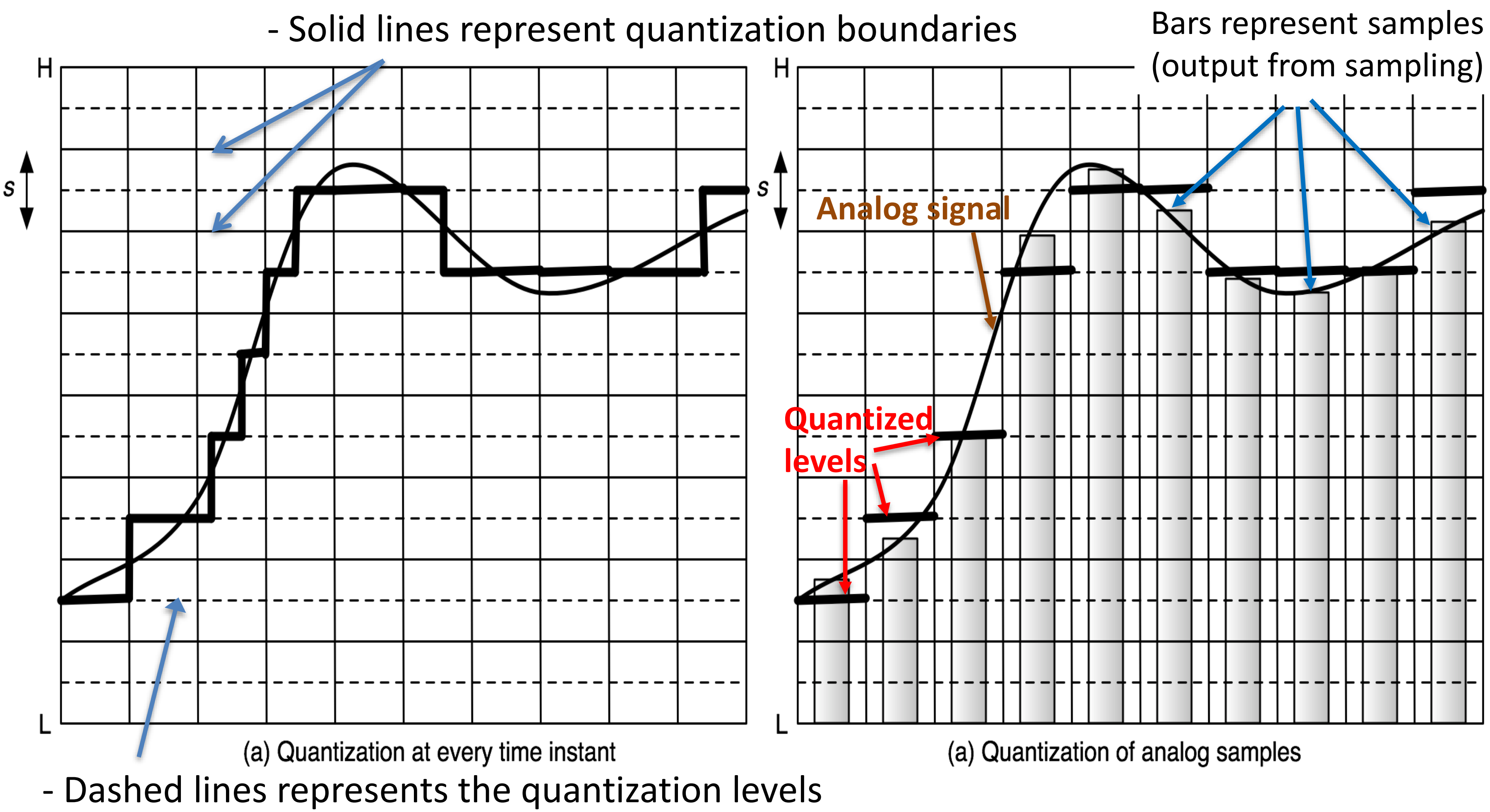
- After sampling the data, quantization takes place

- Main objective: **arbitrary values of samples** are mapped to a **finite set of amplitudes**

Any values => finite set of values

Quantization

Quantize the samples into a finite number of levels



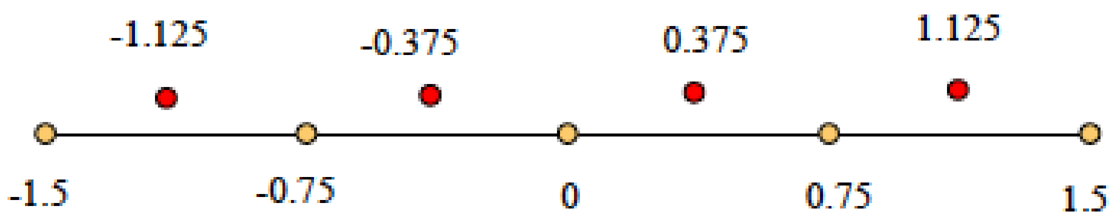
quantization error = exact value – approximation

Example

- For the sequence: {1.2, -0.2, -0.5, 0.4, 0.89, 1.3}. Quantize it using uniform quantizer with range of (-1.5, 1.5) with 4 levels. What is the quantized sequence?

Example

- The size of each quantization interval
  – S=(H-L)/N=(1.5- (-1.5))/4=3/4=0.75
- The quantization levels are at midpoint of interval



- Map the sequence to quantization levels.

Given sequence is {1.2, -0.2, -0.5, 0.4, 0.89, 1.3}.
Then quantized sequence is:

{1.125, -0.375,-0.375,0.375,1.125,1.125}

Quantization Types

- Uniform (covered)
  - Quantization regions are of same length
  - Quantization levels are at midpoints
- Non-uniform (out of scope)
  - Quantization region need not be of same length

After Quantization – Encoding

- After we quantized the signal, we do encoding
  - Convert the finite quantization levels into bits

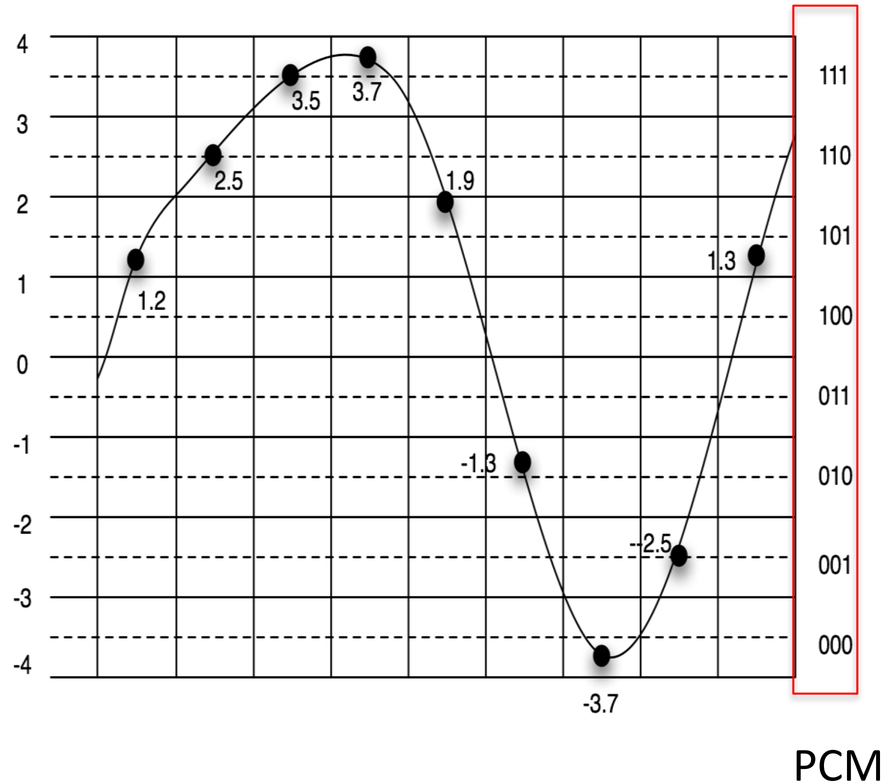
- Encoding is to **represent the quantized values in bits**

- If we have M quantization levels, then the number of bits to represent each level is

$$\mathbf{b = Log_2(M)}$$

After Quantization – Encoding: Pulse Code Modulation (PCM)

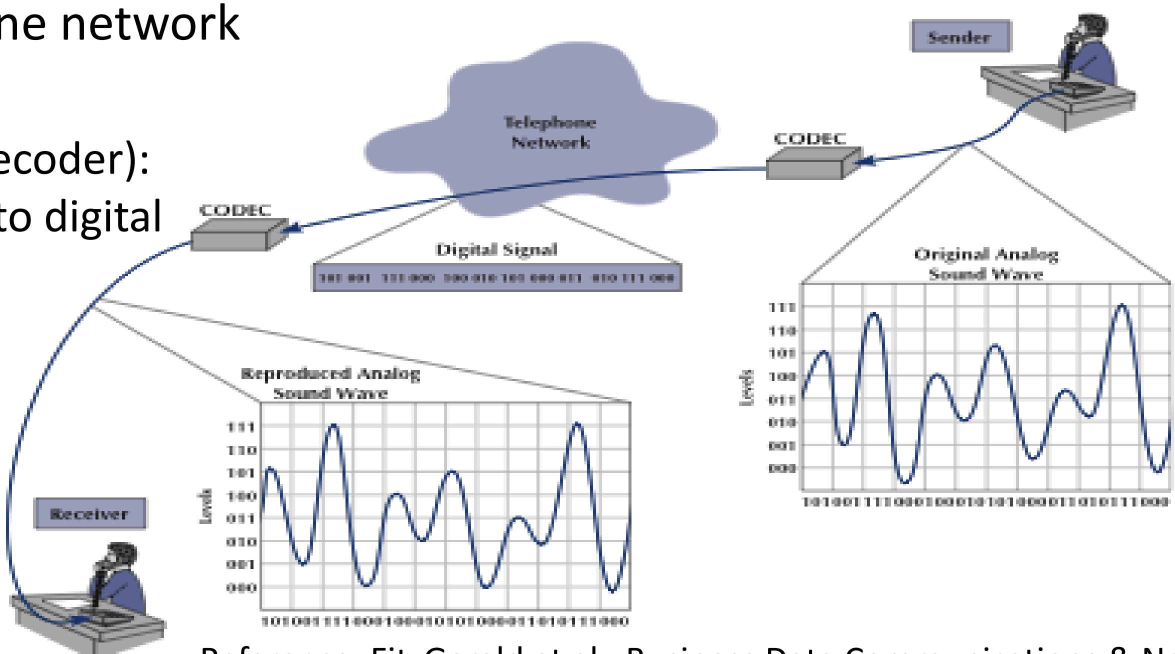
- The conversion of a **quantized signal into bits** is called "**pulse code modulation**" or **PCM**
  - We use n-bits to represent the samples
  - In the figure we have 8 quantization levels
    - Need $\log_2(8) = 3$ bits to represent each level



Telephone Network

- Common carriers (telephone companies) now convert phone networks to digital using PCM
- Local loop (last mile to user) is analog
  - Wires from home to telephone switch carries analog signal
  - Switch contains CODEC to convert signals to digital then transmit it over telephone network

CODEC (coder/decoder): converts analog to digital

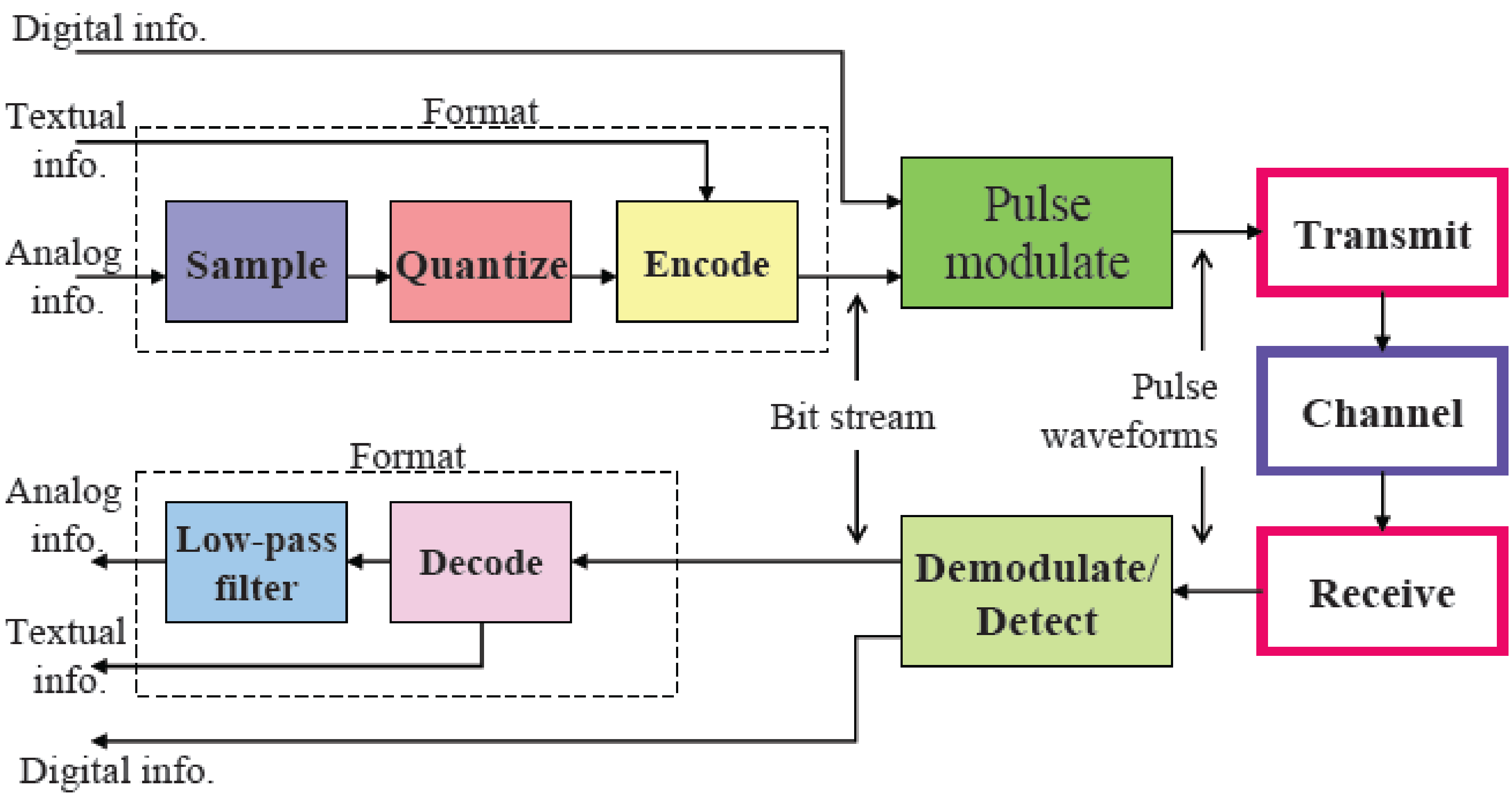


Reference: FitzGerald et al., Business Data Communications & Networking

Voice over Internet Protocol

- Digital phones with built-in CODECs to convert signals into digital
- Use packet switching
- Can be directly connected to the LAN network
  - Similar to any computer
  - No need for separate network for voice
- Many protocol standards
  - E.g. standard G.722
    - Sample 8000 times per sec
    - 8 bits per sample (encoding)

Revisit: Typical Digital Baseband Communication System



Transmission Approaches

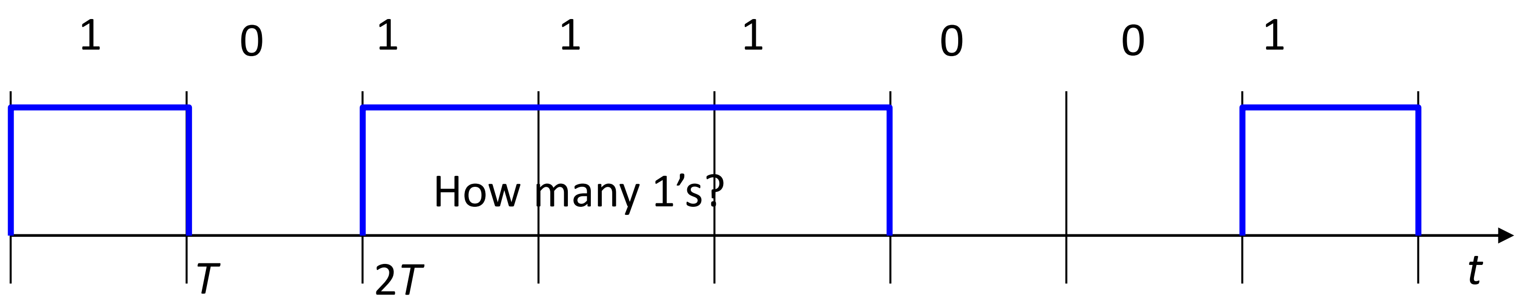
- Two primary transmission approaches
  - Baseband: supports frequency = 0
    - Signals have frequency close to zero
    - Example: Ethernet, Voice on copper cable in landlines
  - Passband: <u>does not</u> support frequency = 0
    - AM/FM radio, Cellular Telephone Signals, Coaxial cable

Baseband Pulse Modulation: Line Coding

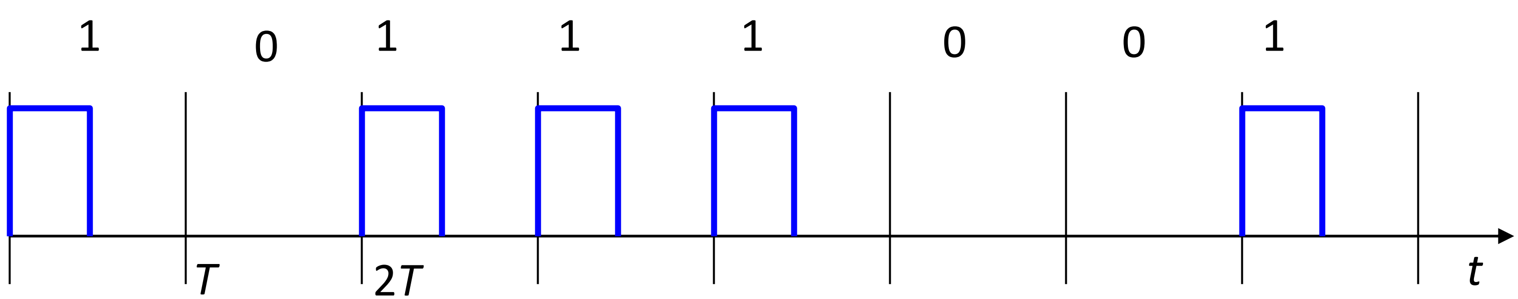
- Now, we got the bits (through sampling, quantization, encoding), we need to generate the signals
- In baseband, we use line codes
- Characteristics of line codes:
  - **Unipolar**
    - Signal values are positive voltage or zero
  - **Bipolar (or Antipodal)**
    - Both positive and negative voltage values (usually identical in magnitude) exist
  - **Non Return-to-Zero (NRZ)**
    - Each digital value is represented by a voltage pulse that is constant for the entire symbol (or bit) duration
  - **Return-to-Zero (RZ)**
    - Voltage pulses return to zero before the end of the symbol (or bit) duration

Line Coding

- **Unipolar**: Signal values are positive or zero voltage
- **Unipolar & non-return-to-zero (NRZ)**

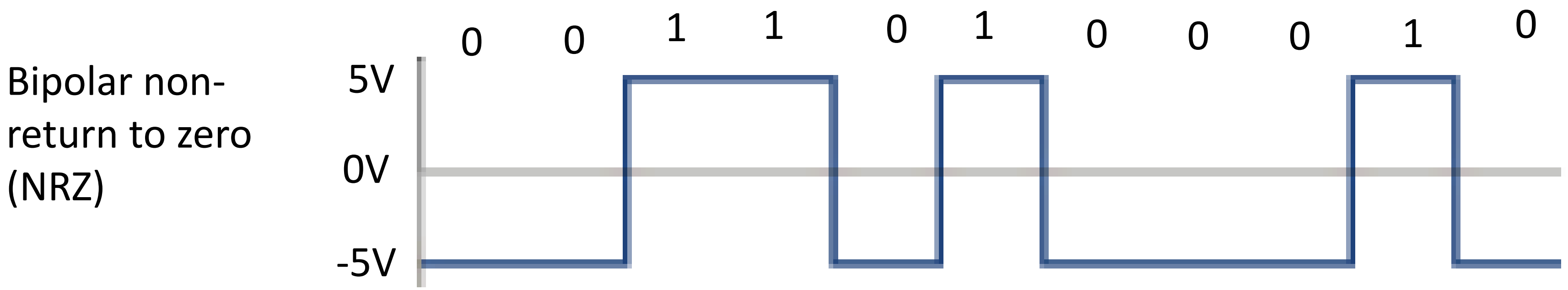


- **Unipolar & return to zero (RZ) ..**
  – Transition to zero helps in decoding the signal (now it is clearer how many zeros)



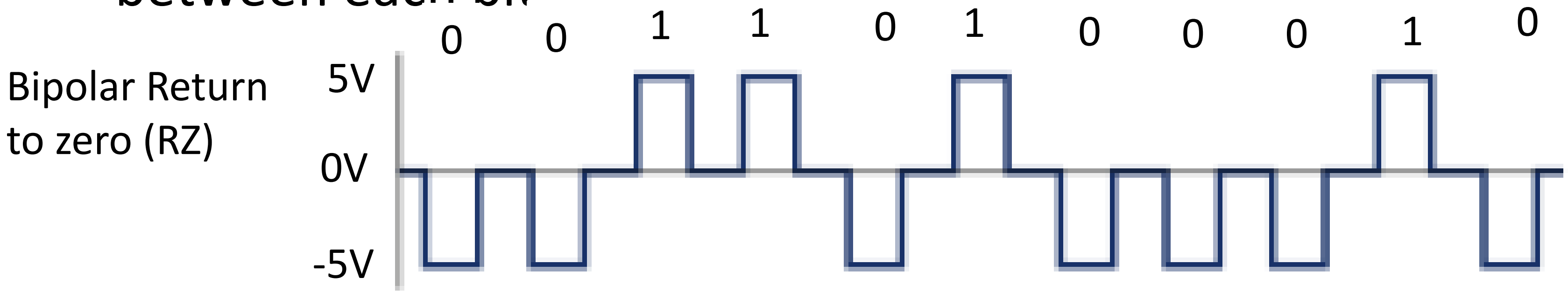
Line Coding

- **Bipolar NRZ -** voltage is positive or negative, but not zero
  – Fewer errors than unipolar because signals are more distinct



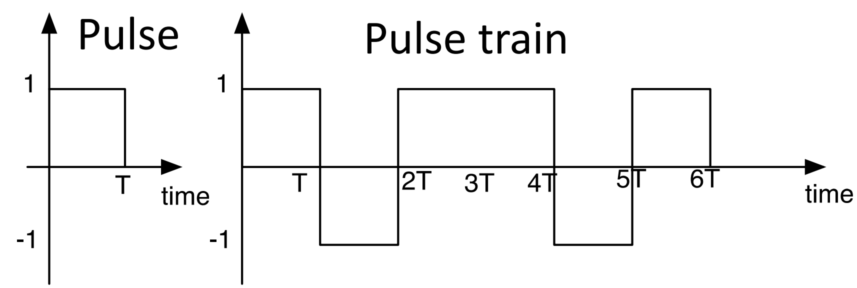
Bit ‘1’ is represented by 5 Volts and ‘0’ is represented by -5 Volts

- **Bipolar RZ -** voltage is positive or negative, returning to zero between each bit

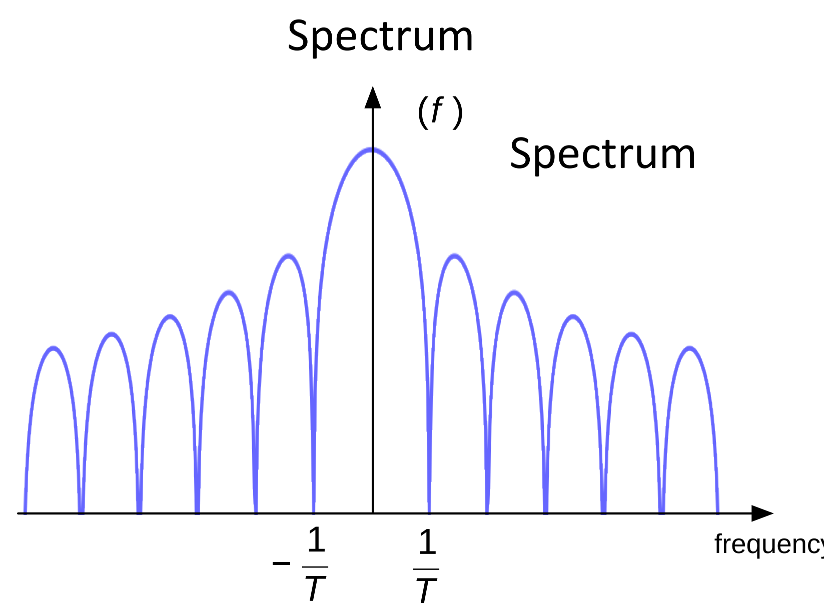


Comparison

- Less errors with bipolar
  - Sampling threshold for distinguishing ‘1’ from ‘0’
    - Bipolar has a zero threshold for binary

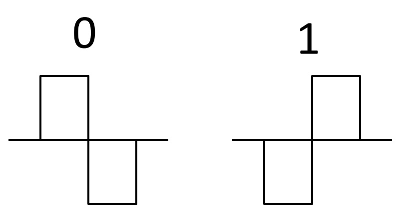


- Bandwidth
  - RZ needs more bandwidth
  - More transitions => higher frequency => more bandwidth

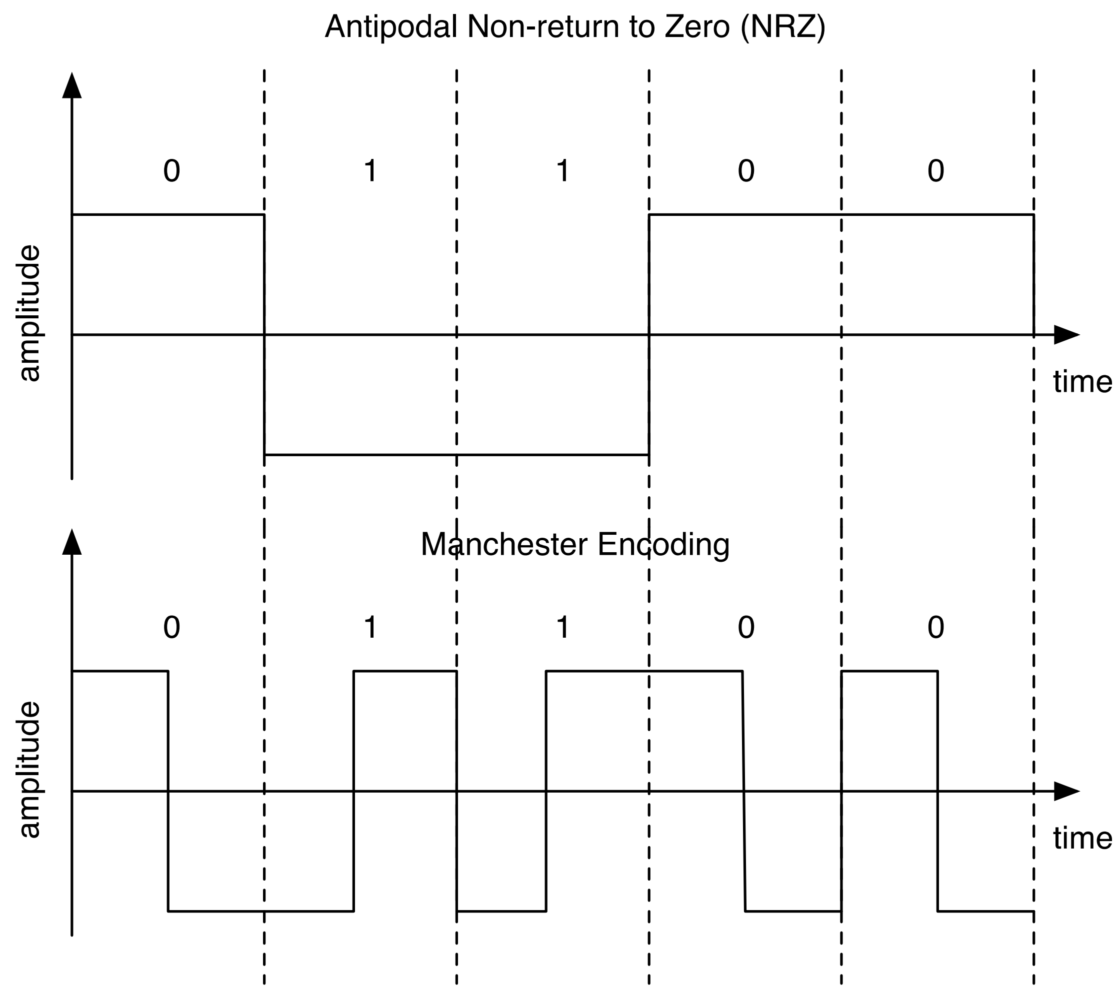


Manchester Code

- Used in **Ethernet**
- **Special type of bipolar**
- **Transitions from high to low in the middle for ‘0’ and from low to high for ’1’ (Or vise versa)**



- Bandwidth like RZ



Tophat: Q_Manchester Encoding

Transmission Modes

- Two transmission modes: parallel and serial

- **Serial transmission:** bits are transmitted sequentially over a link (e.g. wire)

- **Parallel transmission:** multiple bits transmitted simultaneously
  - Used inside computers
    - 8 bit structure computers: 8 bits are transferred in parallel between memory & processing unit using 8 separate wires

Key takeaways

- Digital transmission of digital data
  - Encoding
    - Example: ASCII, Unicode
- Digital transmission of analog data
  - Sampling
  - Quantization
  - Pulse code modulation
- Baseband digital transmission: Line codes (bipolar, unipolar, RZ, NRZ, Manchester)
- Parallel and serial transmission modes