

Lecture 23: Modulation

CSE 123: Computer Networks
Alex C. Snoeren



HW 4 DUE FRIDAY





Lecture 23 Overview

- Signaling
 - ♦ Channel characteristics
- Signaling constraints
 - ♦ Inter-Symbol Interference
 - ♦ Shannon's Law
- Encoding schemes
 - ♦ Clock recovery
 - ♦ Manchester, NRZ, NRZI, etc.



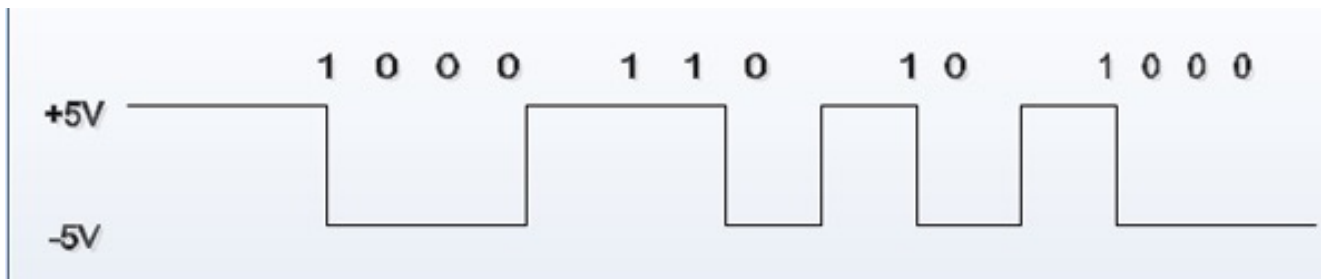
Two Main Tasks

- First we need to transmit a signal
 - ◆ Determine how to send the data, and how quickly
- Then we need to receive a (degraded) signal
 - ◆ Figure out when someone is sending us bits
 - ◆ Determine which bits they are sending
- A lot like a conversation
 - ◆ “WhatintheworldamIsaying” – needs punctuation and pacing
 - ◆ Helps to know what language I’m speaking

Binary signaling with Voltage



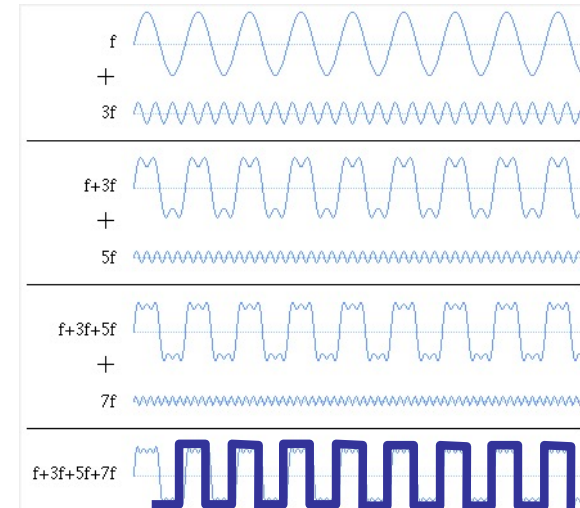
- Encode 1's and 0's on a wire
 - ◆ +5 volts = 1
 - ◆ -5 volts = 0





The Magic of Sine Waves

- All periodic signals can be expressed as sine waves
 - ◆ Component waves are of different frequencies
- Sine waves are “nice”
 - ◆ Phase shifted or scaled by most channels
- “Easy” to analyze
 - ◆ But not in this class...
- The higher the frequency, the “sharper” the edges





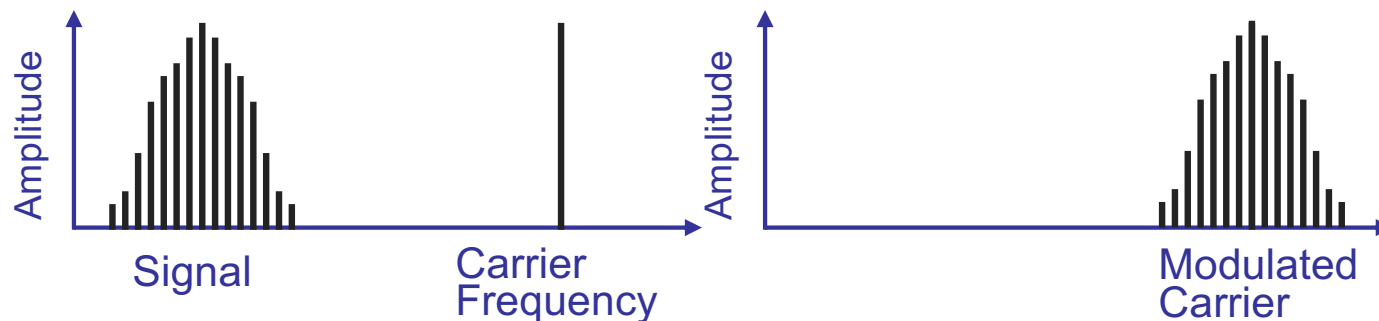
Channel Properties

- **Bandwidth**-limited
 - ◆ *Range* of frequencies the channel will transmit
 - ◆ Means the channel is slow to react to change in signal
- Power **attenuates** over distance
 - ◆ Signal gets softer (harder to “hear”) the further it travels
 - ◆ Different frequencies have different response (**distortion**)
- Background **noise** or interference
 - ◆ May add or subtract from original signal
- Different physical characteristics
 - ◆ Point-to-point vs. shared media
 - ◆ Very different price points to deploy

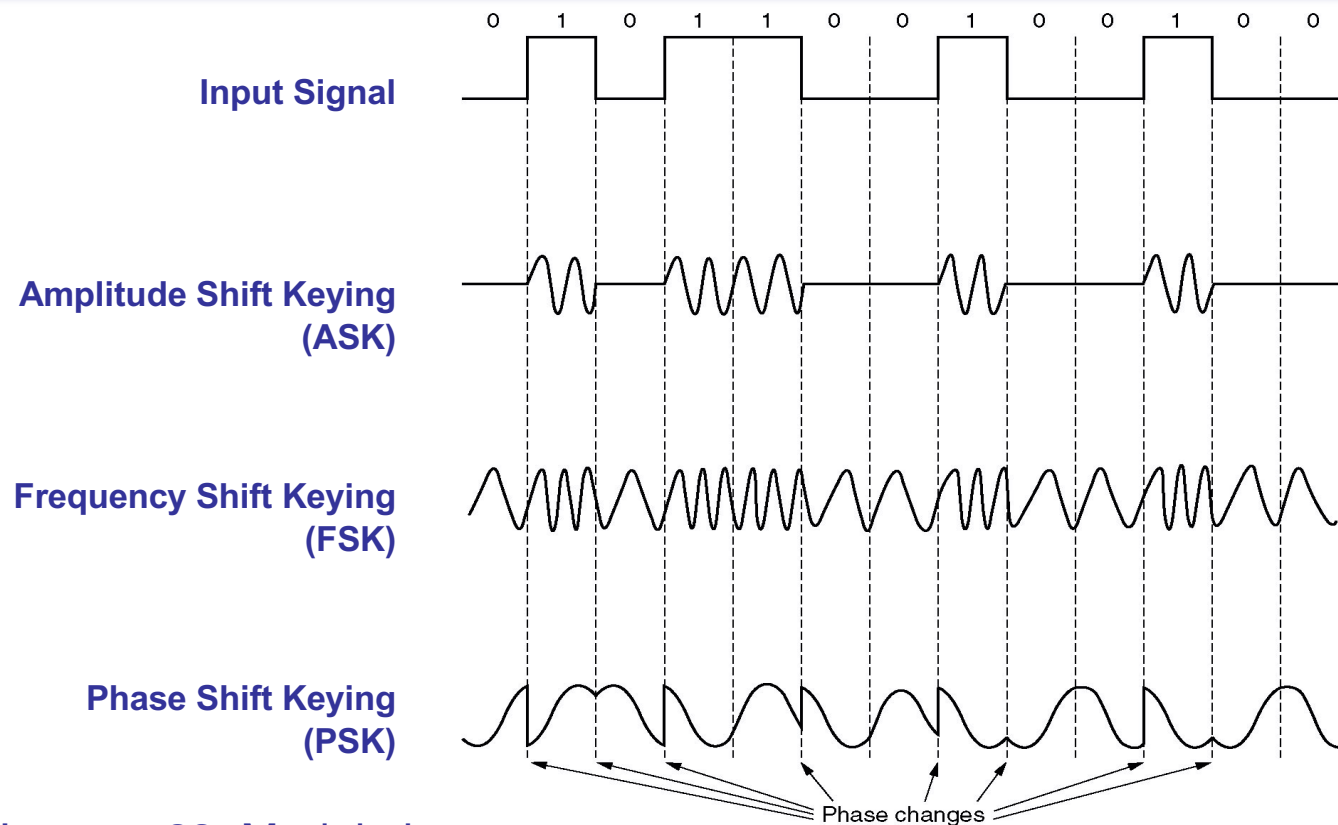
Carrier Signals



- **Baseband** modulation: send the “bare” signal
 - ♦ E.g. +5 Volts for 1, -5 Volts for 0
 - ♦ All signals fall in the same frequency range
- **Broadband** modulation
 - ♦ Use the signal to modulate a high frequency signal (**carrier**).
 - ♦ Can be viewed as the product of the two signals



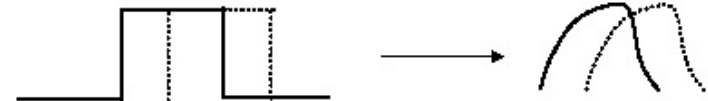
Forms of Digital Modulation





Nyquist rate

- Bandlimited channels cannot respond faster than some maximum frequency f
 - ♦ Channel takes some time to settle
- Attempting to signal too fast will mix symbols
 - ♦ Previous symbol still “settling in”
 - ♦ Mix (add/subtract) adjacent symbols
 - ♦ Leads to **intersymbol interference** (ISI)
- Nyquist says in a channel bandlimited to f , we can send at maximum symbol (**baud**) rate of $2f$ without ISI



Hartley's Law



- Nyquist limits the number of symbols per second we can send, but doesn't talk about the information content in each symbol
- Couldn't we send *multiple* bits per symbol?
 - ◆ E.g., multiple voltage levels instead of just high/low
 - ◆ Four levels gets you two bits, $\log_2 M$ in general (M levels)
- Can combine this observation with Nyquist
 - ◆ *Channel capacity: $C < 2 B \log_2(M)$*

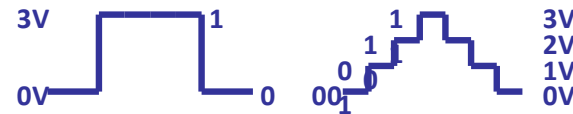
What limits the number of levels we can send?

- A. Sender power**
- B. Receiver fidelity**
- C. Channel properties**
- D. All of the above**

Noise matters



- Real channels are **noisy**... noise creates measurement challenges
- Example:
 - ♦ Encode 4 values using voltage
 - » 2 bits per symbol
 - » Symbols at 3V, 2V, 1V and 0V
 - ♦ What if noise is 0.5V?
 - » If you get line level of 2.5V then what symbol is it? 11 or 10?
- Limited to $\sim \log_2 (S/2N)$ bits per symbol (S = signal power, N = Noise)
 - ♦ Previous example: $S = 3V - 0V = 3V$, $N = 0.5V$, so we can have $\log_2(3/1) = 1.58$ bits per symbol



Shannon's Law



- Shannon considered noisy channels and derived

$$C = B \log_2 (1 + S/N)$$

- Gives us an upper bound on any channel's performance regardless of signaling scheme
- Old school modems approached this limit
 - ♦ $B = 3000\text{Hz}$, $S/N = 30\text{dB} = 1000$
 - ♦ $C = 3000 \times \log(1001) \approx 30\text{kbps}$
 - ♦ 28.8Kbps – anyone remember dialup?

The sounds of modulation



**3.
V.34 (1994):
28.8K - 33.6K**

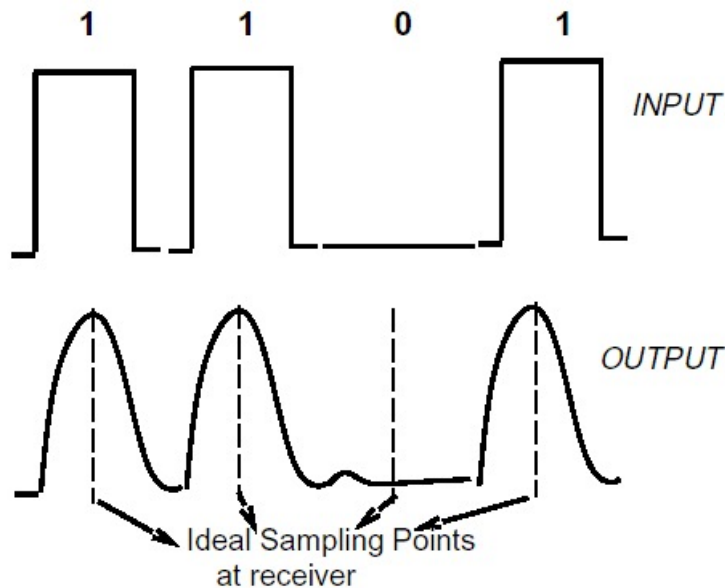
- Uses an automated smart training sequence**
 - Superior echo cancellation**
 - Adapts automatically to poor lines**
- Modulation: TCM**

<https://www.youtube.com/watch?v=ckc6XSSh52w>

Sampling



- To reconstruct signal we need to sample it



Which of the following are potential interpretations of the signal at left?

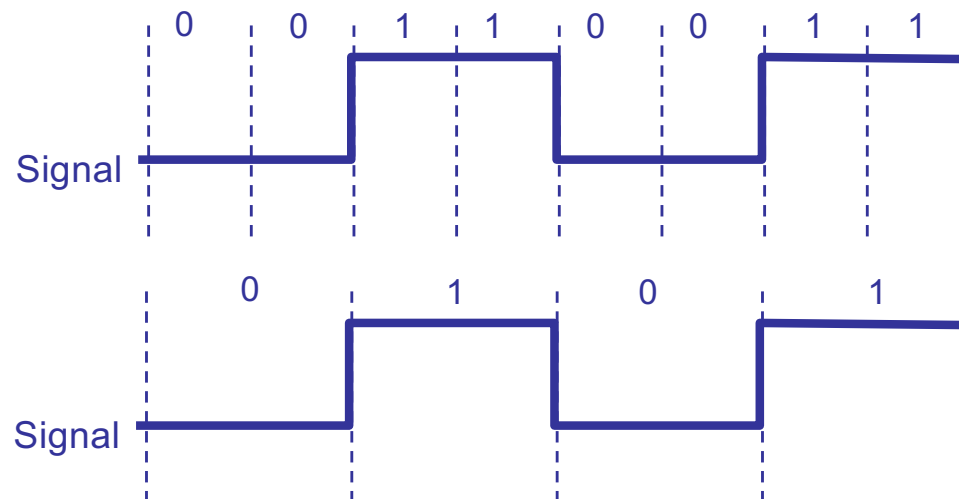
- A. 1101
- B. 11110011
- C. 0000
- D. All of the above

Why sampling rate matters...



- Signal could have multiple interpretations

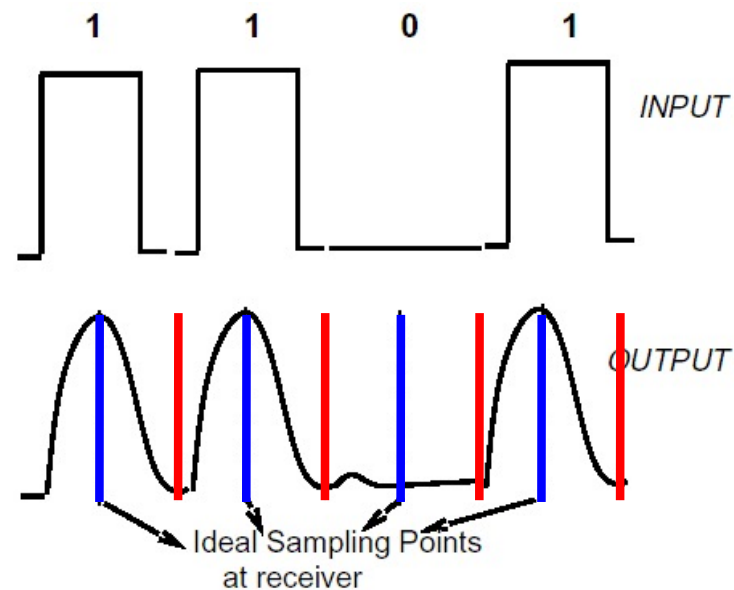
Which of these is correct?





The Importance of Phase

- Need to determine when to START sampling, too





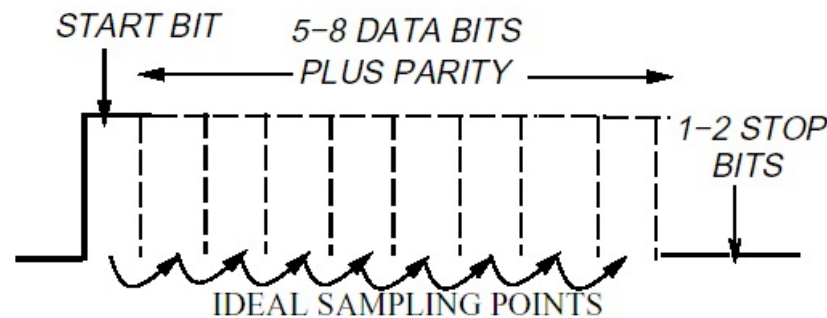
Clock Recovery

- Using a training sequence to get receiver lined up
 - ◆ Send a few, known **initial training bits**
 - ◆ Adds inefficiency: only m data bits out of n transmitted
- Need to combat clock drift as signal proceeds
 - ◆ Use **transitions** to keep clocks synched up
- Question is, how often do we do this?
 - ◆ Quick and dirty every time: **asynchronous** coding
 - ◆ Spend a lot of effort to get it right, but amortize over lots of data: **synchronous** coding



Asynchronous Coding

- Encode several bits (e.g. 7) together with a leading “start bit” and trailing “stop bit”
- Data can be sent at any time

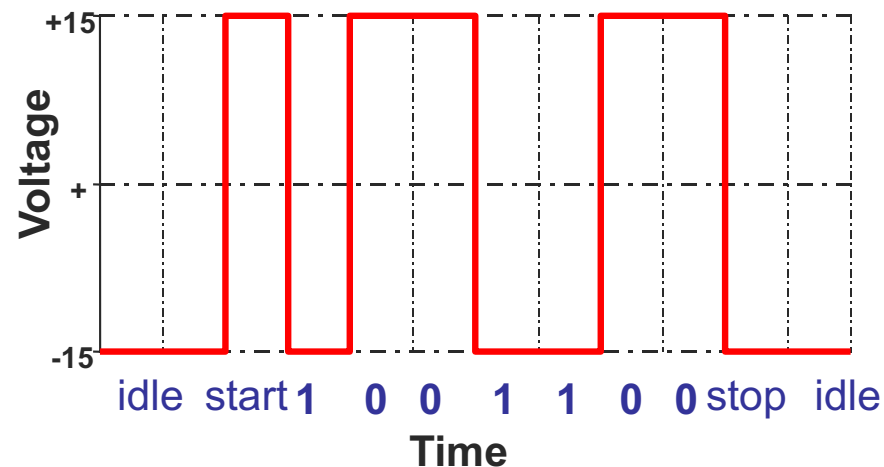


- Start bit transition kicks off sampling intervals
 - Can only run for a short while before drifting



Example: RS232 serial lines

- Uses two voltage levels (+15V, -15V), to encode single bit binary symbols
- Needs long idle time – limited transmit rate



Synchronous Coding

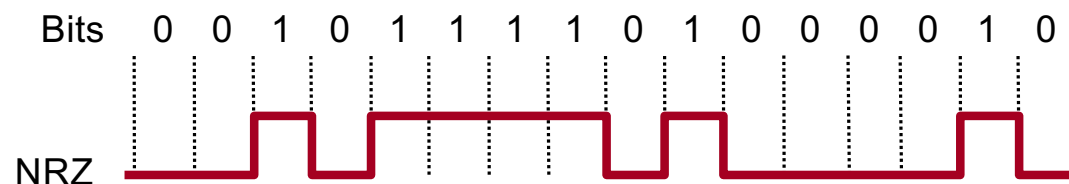


- Asynchronous receiver phase locks each symbol
 - ♦ Takes time, limiting transmission rates
- So, start symbols need to be extra slow
 - ♦ Need to fire up the clock, which takes time
- Instead, let's do this training once, then just keep sync
 - ♦ Need to continually adjust clock as signal arrives
 - ♦ Ever hear of Phase Lock Loops (PLLs) ?
- Basic idea is to use **transitions** to lock in



Non-Return to Zero (NRZ)

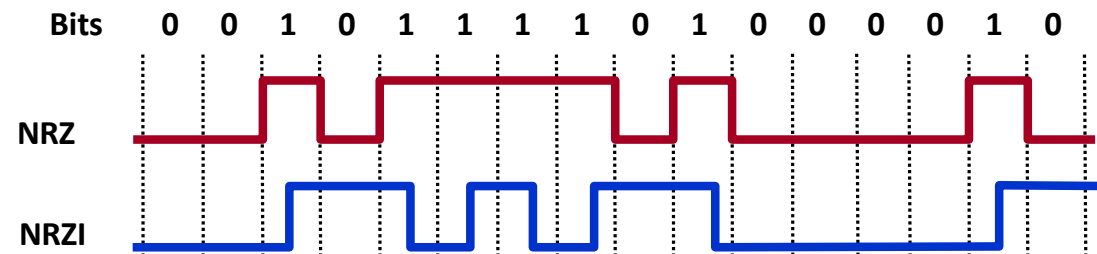
- Signal to Data
 - ♦ High $\Rightarrow$ 1
 - ♦ Low $\Rightarrow$ 0
- Comments
 - ♦ Transitions maintain clock synchronization
 - ♦ Long strings of 0s confused with no signal
 - ♦ Long strings of 1s causes *baseline wander*
 - » We use average signal level to infer high vs low
 - ♦ Both inhibit clock recovery





Non-Return to Zero Inverted (NRZI)

- Signal to Data
 - ♦ Transition $\Rightarrow$ 1
 - ♦ Maintain $\Rightarrow$ 0
- Comments
 - ♦ Solves series of 1s, but not 0s



Manchester Encoding (10Mbps Ethernet)

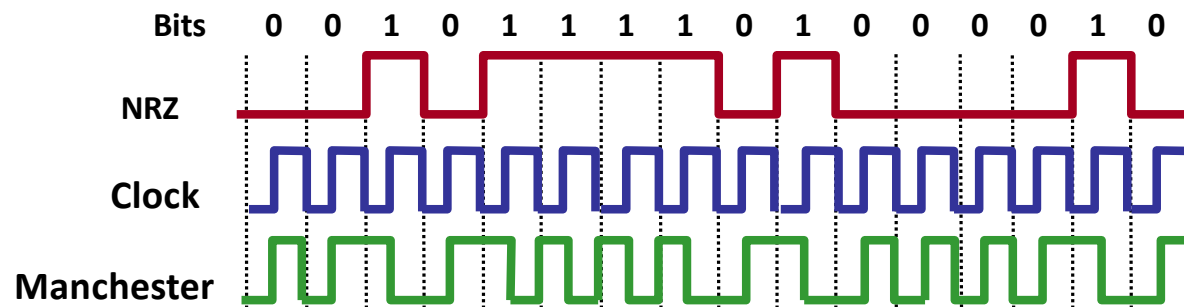


- Signal to Data

- ♦ XOR NRZ data with sender's clock signal
- ♦ High to low transition $\Rightarrow$ 1
- ♦ Low to high transition $\Rightarrow$ 0

- Comments

- ♦ Solves clock recovery problem
- ♦ Only 50% efficient ($\frac{1}{2}$ bit per transition)
- ♦ Still need preamble (typically 0101010101... trailing 11 in Ethernet)



Encoding Summary



- Signaling & Modulation
 - ◆ Transforming digital signal to and from analog representation
 - ◆ Fundamental limits (Shannon)
 - ◆ Lots of ways to encode signal (modulation) onto a given medium
- Clock recovery
 - ◆ Receiver needs to adjust its sampling times to best extract signal from channel
 - ◆ Sender can code signal to make it far easier to do this

For Next Time



- Read 2.6 in P&D
- HW 4 due Friday
- Keep going on the project...