Lecture 3:
Signaling and Clock Recovery

Stefan Savage
Today

- How to send data from point A to point B?
10,000 foot review: digital vs analog

- Data can be represented continuously as an analog wave

- Or as a series of discrete digital symbols

- Why is the digital representation so powerful?
Essence of data signaling

- Take digital data (usually binary symbols)
- Encode it into an analog medium (called modulation) so it can travel distance
- Decode it from the analog medium (called demodulation) back into digital data
Simple example of data signaling

- Medium: Sound pressure in air
- Modulation: binary yodel modulation
  - “Yodeley-yoohoo” = 1
  - “Yodeley-hooyou” = 0
Properties of different mediums

- Bandwidth (frequency response)
- Noise (junk in the background)
- Attenuation (how “far” will signal go)
- Distortion (how is signal changed)
- Shared vs isolated
- Cost
Copper

- Typical examples
  - Category 5 Twisted Pair 10-100Mbps 100m
  - Coaxial Cable 10-100Mbps 200m
Fiber Optics

- Typical examples
  - Multimode Fiber 100Mbps 2km
  - Single Mode Fiber 100-2400Mbps 40km

Cheaper to drive (LED vs laser) & terminate

Longer distance (low attenuation)
Higher data rates (low dispersion)
Wireless

- Different frequencies have different properties
- Signals subject to atmospheric/environmental effects

![Frequency Spectrum Diagram](image)

- AM
- FM
- Coax
- Microwave
- TV
- Satellite
- Fiber
- Twisted Pair

Freq (Hz): 10^4 10^6 10^8 10^10 10^12 10^14

Radio  Microwave  IR  Light  UV
Wirless (2)

- Shared medium -> regulated use
Aside: Wireless spectrum

- Policy approach forces spectrum to be allocated like a fixed spatial resource (e.g. land, disk space, etc)
- Reality is that spectrum is time and power shared
- Measurements show that fixed allocations are poorly utilized
Challenges

- No channel is perfect and the original signal gets modified along the way
  - Attenuation: signal power absorbed by medium
  - Distortion: frequency, phase changes
  - Noise: random background “signals”

- Different mediums distort different signals differently
- Note: that here “bandwidth” means frequency over which signals cannot pass through channel
Impact of distortion & noise

Impact of distortion & noise

Impact of distortion & noise

Impact of distortion & noise

Impact of distortion & noise
Sampling

- To reconstruct signal we need to sample it
Sampling limitations: Nyquist limit

- Channel with bandwidth (max frequency) $B$
- Can’t sample distinct “symbols” faster than $2B$/sec
- Intuition why?
  - Of course… we could send more bits per symbol
    - Instead of encoding 0 and 1, encode 8 levels per waveform
    - Why stop there… can’t we encode an infinite # of bits into a waveform?
Sampling limitations: Shannon’s limit

- Nope... **noise** stops you you
  - At most $\log (S/2N)$ bits per symbol
- Example:
  - Encode 4 values using voltage
    - 3V, 2V, 1V and 0V
  - What if noise is 0.5V?
    - If you get line level of 2.5V then what symbol is it? 2 or 3?
- Combine with Nyquist to get capacity of channel
  - $C = 2B \log (S/2N)$ bits per second ($S =$ signal power)
  - Actual bound is: $C = B \log (1 + S/2N)$
    (this is so the EE profs won’t kill me)
- Real life: telephone lines (pre 1990s)
  - $B = 3000\text{Hz}$, $S/N = 30\text{dB} = 1000$
  - $C = 3000 \times \log(1001) \approx 30\text{kbps}$
  - And this is why modems of the day were 28.8kbps
How to get the digital signal onto the analog medium?

- More concretely: how to represent a 0 and a 1?

- Baseband modulation: send the “bare” signal
  - E.g. +5Volts for 1, -5Volts for 0
  - Needs wide frequency range

- Broadband modulation
  - Use the signal to modulate a narrow high frequency signal (carrier).
  - Can be viewed as the product of the two signals

- Lots of options for modulation
  - Amplitude, frequency, phase, combinations of these…
Example: Amplitude Shift Keying

Amplitude Signal Carrier

Frequency

Amplitude Modulated Carrier

Amplitude

Signal

Carrier Frequency

Modulated Carrier
Other forms of modulation

- **binary signal**

- **Amplitude modulation**

- **Frequency modulation**

- **Phase modulation**
**Why pick one form of modulation over the other?**

- Properties of channel and desired application
  - AM vs FM for analog radio

- Efficiency
  - Some modulations can encode many bits for each symbol (subject to Shannon limit)

- Aiding with error detection
  - Dependency between symbols… can tell if a symbol wasn’t decoded correctly

- Complexity
How does the receiver know when to sample the signal?

- Sampling rate: How often to sample?
- Sampling phase:
  - When to start sampling? (getting in phase)
  - How to adjust sampling times (saying in phase)
How often to sample?

- Sender and receiver must agree on signaling rate, otherwise…

Which of these is correct?
Question

- If sender and receiver both sample at the same rate isn’t that enough?

- Why not?
Sampling Phase
First try: 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
- Use start bit to adjust local clock to sample at correct points
Example: RS232 serial lines

- Uses two voltage levels (+15V, -15V), to encode single bit binary symbols
- Limited to ~19.2kbps in practice (need long idle time)
Synchronous coding

- Encode many bits (thousands) together
  - Amortize cost of learning clock information from start bits (preamble) and stop bits (trailer)
  - Continuously “learn” clock from data stream
- Examples
  - NRZ
  - NRZI
  - Manchester
  - 4B/5B
  - Many others…
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

Bits: 0 0 1 0 1 1 1 1 0 1 0 0 0 0 1 0

NRZ: 0 0 1 0 1 1 1 1 0 1 0 0 0 0 1 0

Courtesy Robin Kravets
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 ⇒ 1
  - Low to high transition ⇒ 0

- Comments
  - Solves clock recovery problem
  - Only 50% efficient (½ bit per transition)
  - Still need preamble (typically 0101010101… trailing 11 in Ethernet)
**4B/5B (100Mbps Ethernet)**

- Goal: address inefficiency of Manchester encoding, while avoiding long periods of low signals
- Solution:
  - Use five bits to encode every sequence of four bits
  - No 5 bit code has more than one leading 0 and two trailing 0’s
  - Use NRZI to encode the 5 bit codes
  - Efficiency is 80%

<table>
<thead>
<tr>
<th>4-bit</th>
<th>5-bit</th>
<th>4-bit</th>
<th>5-bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>11110</td>
<td>1000</td>
<td>10010</td>
</tr>
<tr>
<td>0001</td>
<td>01001</td>
<td>1001</td>
<td>10011</td>
</tr>
<tr>
<td>0010</td>
<td>10100</td>
<td>1010</td>
<td>10110</td>
</tr>
<tr>
<td>0011</td>
<td>10101</td>
<td>1011</td>
<td>10111</td>
</tr>
<tr>
<td>0100</td>
<td>01010</td>
<td>1100</td>
<td>11010</td>
</tr>
<tr>
<td>0101</td>
<td>01011</td>
<td>1101</td>
<td>11011</td>
</tr>
<tr>
<td>0110</td>
<td>01110</td>
<td>1110</td>
<td>11100</td>
</tr>
<tr>
<td>0111</td>
<td>01111</td>
<td>1111</td>
<td>11101</td>
</tr>
</tbody>
</table>
Summary

- Signaling
  - Transforming digital signal to and from analog representation
  - Fundamental limits (Nyquist/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

- This is the most EEish lecture you will ever receive from me – its all downhill from here
For Next Time...

- We’ll cover framing and error detection
- Read 2.3 and 2.4