Lecture 2: Links and Signaling

CSE 123: Computer Networks
Alex C. Snoeren

Project 1 out Today, due 10/26
Lecture 2 Overview

- **Signaling**
  - Types of physical media
  - Shannon’s Law and Nyquist Limit

- **Encoding schemes**
  - Clock recovery
  - Manchester, NRZ, NRZI, etc.

- A lot of this material is **not in the book**

- **Caveat**: I am not an EE Professor
Today’s Goal: Send bits

- A three-step process
  - Take an input stream of bits (digital data)
  - Modulate some physical media to send data (analog)
  - Demodulate the signal to retrieve bits (digital again)
  - Anybody heard of a modem (Modulator-demodulator)?
A Simple Signaling System

SEMAPHORE (marine alphabet 1 of 3)

- A or 1
- B or 2
- C or 3
- D or 4
- E or 5
- F or 6
- G or 7
- H or 8
- I or 9
- J or letters follow

www.infovisual.info
Signals and Channels

- A signal is some form of energy (light, voltage, etc)
  - Varies with time (on/off, high/low, etc.)
  - Can be continuous or discrete
  - We assume it is periodic with a fixed frequency

- A channel is a physical medium that conveys energy
  - Any real channel will distort the input signal as it does so
  - How it distorts the signal depends on the signal
Channel Challenges

- Every channel degrades a signal
  - Distortion impacts how the receiver will interpret signal

![Diagram showing frequency response](image)
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
Copper

- Typical examples
  - Category 5 Twisted Pair: 10-1Gbps, 50-100m
  - 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)
Common Link Speeds

- Copper based off of old phone-line provisioning
  - Basic digital service was 64-Kbps ISDN line
  - Everything else is an integer multiple
    » T-1 is 24 circuits 24 * 64 = 1.544 Mbps
    » T-3 is 28 T-1s, or 28 * 1.544 = 44.7 Mbps

- Optical links based on STS standard
  - STS is electrical signaling, OC is optical transmission
  - Base speed comes from STS-1 at 51.84 Mbps
  - OC-3 is 3 * 51.84 = 155.25 Mbps

- Move to asymmetric link schemes
  - Your service at home is almost surely ADSL
Wireless

- Widely varying channel bandwidths/distances
- Extremely vulnerable to noise and interference
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

Whitespaces, anyone?
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
  - “What in the world am I saying” – needs punctuation and pacing
  - Helps to know what language I’m speaking
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
  - Fourier analysis can tell us how signal changes
  - But not in this class…
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

- Input Signal
- Amplitude Shift Keying (ASK)
- Frequency Shift Keying (FSK)
- Phase Shift Keying (PSK)

Phase changes
Why Different Schemes?

- 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

- Transmitter/receiver Complexity
Intersymbol Interference

- 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)

- OK, so just how fast can we send symbols?
In a channel bandlimited to $f$, we can send at maximum symbol (baud) rate of $2f$ without ISI.
Multiple Bits per Symbol

- OK, but why not send multiple bits per symbol
  - E.g., multiple voltage levels instead of just high/low
  - Four levels gets you two bits, \( \log L \) in general
  - Could we define an infinite number of levels?

- Channel noise limits bit density
  - Intuitively, need level separation
  - Only get \( \log(S/2N) \) bits per symbol

- Can combine this observation with Nyquist
  - \( C < 2B \log(S/2N) \) in a perfect channel, but…
Shannon considered noisy channels and derived

\[ C = B \log (1 + S/N) \]

- Gives us an upper bound on any channel’s performance regardless of signaling scheme
- Old school modems approached this limit
  - \( B = 3000\)Hz, \( S/N = 30\)dB = 1000
  - \( C = 3000 \times \log(1001) \approx 30\)kbps
  - 28.8Kbps, anyone?
Sampling at the Receiver

- Need to determine correct sampling frequency
  - Signal could have multiple interpretations

Which of these is correct?
Nyquist Revisited

- Sampling at the correct rate ($2f$) yields actual signal
  - Always assume lowest-frequency wave that fits samples

- Sampling too slowly yields aliases
The Importance of Phase

- Need to determine when to START sampling, too

![Diagram showing input and output signals with ideal sampling points at the receiver]
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 Looks (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

![NRZ Bit Sequence](Image)
Non-Return to Zero Inverted (NRZI)

- Signal to Data
  - Transition $\Rightarrow 1$
  - Maintain $\Rightarrow 0$

- Comments
  - Solves series of 1s, but not 0s

NRZI

NRZ

Bits 0 0 1 0 1 1 1 1 0 1 0 0 0 0 1 0

Courtesy Robin Kravets
Manchester Encoding

- **Signal to Data**
  - XOR NRZ data with senders clock signal
  - High to low transition \( \Rightarrow 1 \)
  - Low to high transition \( \Rightarrow 0 \)

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

![Manchester Encoding Diagram](image-url)
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>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>11110</td>
</tr>
<tr>
<td>0001</td>
<td>01001</td>
</tr>
<tr>
<td>0010</td>
<td>10100</td>
</tr>
<tr>
<td>0011</td>
<td>10101</td>
</tr>
<tr>
<td>0100</td>
<td>01010</td>
</tr>
<tr>
<td>0101</td>
<td>01011</td>
</tr>
<tr>
<td>0110</td>
<td>01110</td>
</tr>
<tr>
<td>0111</td>
<td>01111</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>4-bit</th>
<th>5-bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>10010</td>
</tr>
<tr>
<td>1001</td>
<td>10011</td>
</tr>
<tr>
<td>1010</td>
<td>10110</td>
</tr>
<tr>
<td>1011</td>
<td>10111</td>
</tr>
<tr>
<td>1100</td>
<td>11010</td>
</tr>
<tr>
<td>1101</td>
<td>11011</td>
</tr>
<tr>
<td>1110</td>
<td>11100</td>
</tr>
<tr>
<td>1111</td>
<td>11101</td>
</tr>
</tbody>
</table>
Summary

- Two basic tasks: send and receive
  - The trouble is the channel distorts the signal

- Transmission modulates some physical carrier
  - Lots of different ways to do it, various efficiencies

- Receiver needs to recover clock to correctly decode
  - All real clocks drift, so needs to continually adjust
  - The encoding scheme can help a lot
For Next Class

- Read 2.3
- Log into Moodle; let me know if you have problems
- Get started on Project 1!