Questions

- Why are some links faster than others?
- What limits the amount of information we can send on a link?
- How can we increase the capacity of a link?
Signals: Analog vs. Digital

- Signal: a function $s(t)$ that varies with time ($t$ stands for time)
- Analog: varies continuously
  - Example: voltage representing audio (analog phone call)

- Digital: discrete values; varies abruptly
  - Example: voltage representing 0s and 1s

Signals: Periodic vs. Aperiodic

- Period: repeat over and over again, once per period
  - Period ($T$) is the time it takes to make one complete cycle
  - Frequency ($f$) is the inverse of period, $f = 1/T$; measured in hz

- Aperiodic: don’t repeat according to any particular pattern
Data vs. Signal

- Links become slower with distance because of signal attenuation
- Amplifiers and repeaters can help
Noise

- A signal $s(t)$ sent over a link is generally
  - Distorted by the physical nature of the medium
    - This distortion may be known and reversible at the receiver
  - Affected by random physical effects
    - Fading
    - Multipath effects
  - Also interference from other links
    - Wireless
    - Crosstalk
- Dealing with noise is what communications engineers do

Noise Limits the Link Rate

- Suppose there were no noise
  - Then, if send $s(t)$ always receive $s(t+\Delta)$
  - Take a message of $N$ bits say $b_1b_2\ldots b_N$, and send a pulse of amplitude of size $0.b_1b_2\ldots b_N$
  - Can send at an arbitrarily high rate
  - This is true even if the link distorts the signal but in a known way
- In practice the signal always gets distorted in an unpredictable (random) way
  - Receiver tries to estimate the effects but this lowers the effective rate
Physical Layer Functions

Adaptor: convert bits into physical signal and physical signal back into bits

- Functions
  1. Encode bit sequence into analog signal
  2. Transmit bit sequence on a physical medium (Modulation)
  3. Receive analog signal
  4. Convert Analog Signal to Bit Sequence

Block Diagram

- Channel Coding
- Modulation
- Medium
- Demodulation
- Channel Decoding
**Modulation**

- The function of transmitting the encoded signal over a link, often by combining it with another (carrier signal)
  - E.g., Frequency Modulation (FM)
    - Combine the signal with a carrier signal in such a way that the frequency of the received signal contains the information of the carrier

- E.g., Frequency Hopping (OFDM)
  - Signal transmitted over multiple frequencies
  - Sequence of frequencies is pseudo random

---

**Outline**

- Relation between bandwidth and link rate
  - Fourier transform
    - Nyquist’s Theorem
    - Shannon’s Theorem
- Encoding
- Framing
Any periodic signal \( g(t) \) with period \( T (=1/f) \) can be constructed by summing a (possibly infinite) number of sines and cosines.

\[
g(t) = \frac{1}{2} c + \sum_{n=1}^{\infty} a_n \sin(2\pi ft) + \sum_{n=1}^{\infty} b_n \cos(2\pi ft)
\]

To construct signal \( g(t) \) we need to compute the values \( a_0 \), \( a_1 \), ..., \( b_0 \), \( b_1 \), ..., and \( c \)!

But it’s an infinite series...

- Often the magnitude of the \( a_n \)'s and \( b_n \)'s get smaller as the frequency \((n \times 2\pi f)\) gets higher.
- Key point: a “reasonable reconstruction” can be often be made from just the first few terms (harmonics)
  - Tough the more harmonics the better the reconstruction...

**Fourier Transform: Example**

\[
sin(2\pi ft) + \frac{1}{3} \sin(6\pi ft) = g_3(t)
\]

*Note: \( f = 1/T \)
Bandwidth & Data Rate

- Physical media attenuate (reduce) different harmonics at different amounts
- After a certain point, no harmonics get through.
- Bandwidth: the range of frequencies that can get through the link
- Example:
  - Voice grade telephone line 300Hz – 3300Hz
  - The bandwidth is 3000Hz
- Data rate: highest frequency at which hardware can send signal

Outline

- Signal study
  - Fourier transform
    - Nyquist’s Theorem
  - Shannon’s Theorem
- Encoding
- Framing
Nyquist’s Theorem (aka Nyquist’s Limit)

- Establish the connection between data rate and bandwidth (actually the highest frequency) in the absence of noise
  - Developed in the context of analog to digital conversion (ACDs)
- Say how often one needs to sample an analog signal to reproduce it faithfully
- Suppose signal s(t) has highest frequency $f_{max}$
  - Assume $B = f_{max}$ i.e., lowest frequency is 0

- Then, if $T \leq \frac{1}{2B}$ it is possible to reconstruct $s(t)$ correctly
- Nyquist’s Theorem: [Data rate (bits/sec) $\leq 2^B$ (Hz)]

Why Double the Frequency?

- Assume a sine signal, then
  - We need two samples in each period to identify sine function
  - More samples won’t help
Nyquist’s Theorem (cont’d)

- Can you do better than Nyquist’s limit?
  - Yes, if clocks are synchronized sender and receiver, we only need one sample per period
  - This is because the synchronized starting sample counts as one of the two points

Outline

- Signal study
  - Fourier transform
  - Nyquist’s Theorem
    - Shannon’s Theorem
- Encoding
- Framing
Shannon Theorem

- Establish the connection between bandwidth and data rate in the presence of noise
- Noisy channel
  - Consider ratio of signal power to noise power.
  - Consider noise to be super-imposed signal
  - Decibel (dB) = 10 log₁₀ (S/N)
  - S/N of 10 = 10 dB
  - S/N of 100 = 20 dB
  - S/N of 1000 = 30 dB

Shannon Theorem (cont’d)

- Data rate in the presence of S/N is bounded as follows

\[ \text{Data rate} \leq B \log_2 (1 + S/N) \]

- Example:
  - Voice grade line: S/N = 1000, B=3000, C=30Kbps
  - Technology has improved S/N and B to yield higher speeds such as 56Kb/s
- Higher bandwidth \(\rightarrow\) higher rate; Intuition:
  - Signal has more space to “hide” from noise
  - Noise gets “diluted” across frequency space
Outline

- Signal study
  - Fourier transform
  - Nyquist’s Theorem
  - Shannon’s Theorem
- Encoding
- Framing

Encoding

- Specify how bits are represented in the analog signal
  - This service is provided by the physical layer
- Challenges: achieve:
  - Efficiency – ideally, bit rate = clock rate
  - Robust – avoid de-synchronization between sender and receiver when there is a large sequence of 1’s or 0’s
Assumptions

- We use two discrete signals, high and low, to encode 0 and 1
- The transmission is synchronous, i.e., there is a clock used to sample the signal
  - In general, the duration of one bit is equal to one or two clock ticks
- If the amplitude and duration of the signals is large enough, the receiver can do a reasonable job of looking at the distorted signal and estimating what was sent.

Non-Return to Zero (NRZ)

- 1 → high signal; 0 → low signal
- Disadvantages: when there is a long sequence of 1’s or 0’s
  - Sensitive to clock skew, i.e., difficult to do clock recovery
  - Difficult to interpret 0’s and 1’s (baseline wander)
Non-Return to Zero Inverted (NRZI)

- 1 → make transition; 0 → stay at the same level
- Solve previous problems for long sequences of 1’s, but not for 0’s

Manchester

- 1 → high-to-low transition; 0 → low-to-high transition
- Addresses clock recovery and baseline wander problems
- Disadvantage: needs a clock that is twice as fast as the transmission rate
  - Efficiency of 50%
4-bit/5-bit (100Mb/s Ethernet)

- Goal: address inefficiency of Manchester encoding, while avoiding long periods of low signals
- Solution:
  - Use 5 bits to encode every sequence of four bits such that 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>

Outline

- Signal study
  - Fourier transform
  - Nyquist’s Theorem
  - Shannon’s Theorem
- Encoding
  - Framing
Framing

- Specify how blocks of data are transmitted between two nodes connected on the same physical media
  - This service is provided by the data link layer
- Challenges
  - Decide when a frame starts/ends
  - If use special delimiters, differentiate between the true frame delimiters and delimiters appearing in the payload data

Byte-Oriented Protocols: Sentinel Approach

- STX – start of text
- ETX – end of text
- Problem: what if ETX appears in the data portion of the frame?
- Solution
  - If ETX appears in the data, introduce a special character DLE (Data Link Escape) before it
  - If DLE appears in the text, introduce another DLE character before it
- Protocol examples
  - BISYNC, PPP, DDCMP
**Byte-Oriented Protocols: Byte Counting Approach**

- Sender: insert the length of the data (in bytes) at the beginning of the frame, i.e., in the frame header
- Receiver: extract this length and decrement it every time a byte is read. When this counter becomes zero, we are done

**Bit-Oriented Protocols**

- Both start and end sequence can be the same
  - E.g., 01111110 in HDLC (High-level Data Link Protocol)
- Sender: in data portion inserts a 0 after five consecutive 1s
- Receiver: when it sees five 1s makes decision on the next two bits
  - If next bit 0 (this is a stuffed bit), remove it
  - If next bit 1, look at the next bit
    - If 0 this is end-of-frame (receiver has seen 01111110)
    - If 1 this is an error, discard the frame (receiver has seen 01111111)
Clock-Based Framing (SONET)

- SONET (Synchronous Optical NETwork)
- Developed to transmit data over optical links
  - Example: SONET ST-1: 51.84 Mbps
  - Many streams on one link
- SONET maintains clock synchronization across several adjacent links to form a path
  - This makes the format and scheme very complicated

SONET Multiplexing

- STS-3c has the payloads of three STS-1’s byte-wise interleaved.
- STS-3 is a SONET link w/o multiplexing
- For STS-N, frame size is always 125 microseconds
  - STS-1 frame is 810 bytes
  - STS-3 frame is 810x3 = 2430 bytes
STS-1 Frame

- First two bytes of each frame contain a special bit pattern that allows to determine where the frame starts
- No bit-stuffing is used
- Receiver looks for the special bit pattern every 810 bytes
  - Size of frame = 9x90 = 810 bytes

Clock-Based Framing (SONET)

- Details:
  - Overhead bytes are encoded using NRZ
  - To avoid long sequences of 0’s or 1’s the payload is XOR-ed with a special 127-bit pattern with many transitions from 1 to 0
What do you need to know?

- Concept of bandwidth and data rate
- Nyquist's Theorem
- Shannon's Theorem
- Encoding
  - Understand (not memorize) NRZ, NRZI, Manchester, 4/5 bit
- Framing
  - Understand framing for bit/byte oriented protocols and clock based framing