Announcements

Originality Declarations to lab demonstrators.

Lab allocations – If you really really need to change, ask demonstrator of the lab that you intend to join.

Lecture outline

Modulation
- Digital signals: Transmission coding
  - Analog signals:
    - Amplitude Shift Keying
    - Frequency Shift Keying
    - Phase Shift Keying

Framing
- Using length fields
- Using special characters
  - Stuffing
  - Bit stuffing
- Other framing techniques

References

Sections of Tanenbaum:
- Analog signals: 2.5.3
- Manchester coding: 4.3.2
- Framing: 3.1.2

(De)Modulation

Modulation is the process of translating the information that we want to transmit into signals that are suitable for the physical transmission medium.

Demodulation is the reverse process. Modems perform modulation and demodulation.

Types of translation:
- Analog → Digital: Done (audio/video sampling)
- Digital → "Digital": Baseband modulation ("encoding")
- Digital → "Analog": AC modulation ("keying")
Simple digital modulation

Amplitude indicates value
Level coding: Amplitude remains level for the duration of the sample period.
  e.g. Non-Return to Zero-Level (NRZ-L): 2 voltage levels, one indicates 0, other indicates 1. e.g. 0V=0, 5V=1
  (may prefer -5V=0, +5V=1 to distinguish transmission from disconnection/idle)

NRZ problems

With NRZ, the signal may remain at one level indefinitely, e.g. NRZ-L: string of 0s or 1s.
  • Clocking problems:
    o Receiver must sample signal at correct time, but
time can only be measured relative to last transition.
    o If receiver’s clock isn’t perfectly synchronised with
transmitter’s, then it may sample at the wrong
instant & interpret wrong value.
  • Eliminating DC component also helps:
    o Avoid corrosion in metallic conductors
    o Can use one conductor for both signal & power
    o Isolate transmitter and receiver through AC coupler
⇒ Manchester encoding: Force transitions within each symbol.
  † Alternative is avoid long strings of 0s or 1s ⇒ stuffing.

Differential encoding

Rather than use the value of a sample to convey information, use the difference between consecutive samples to convey the information.
  e.g. NRZ-Inverted: Transition indicates 1, no transition indicates 0.
  Eliminates DC when transmitting 1s, but no transitions when transmitting 0s

Manchester encoding

Each bit period divided into 2 intervals: One at one voltage, other at opposite voltage.

Ethernet† uses plain Manchester (-0.85V, +0.85V)

† NRZ in the sense that the signal can remain non-zero (long string of 1s for NRZ-L)
**Line coding**

Manchester essentially has 50% overhead: Encodes 1b as 2 levels. Resembles 1b of parity for each bit of data. Overhead can be reduced by grouping data bits and adding slight redundancy to the group for transmission e.g. 4b5b code: 4 bits of data + 1b of redundancy

**Choosing the form of redundancy:**
- Crude: Parity
- More sophisticated types: Better line variations and some symbols may not be needed to carry data ⇒ use for framing etc.

---

**4b5b code**

<table>
<thead>
<tr>
<th>Code</th>
<th>Name</th>
<th>Content</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>43210</td>
<td>0</td>
<td>00000</td>
<td>Data 0</td>
</tr>
<tr>
<td>10011</td>
<td>1</td>
<td>00011</td>
<td>Data 1</td>
</tr>
<tr>
<td>10101</td>
<td>2</td>
<td>00101</td>
<td>Data 2</td>
</tr>
<tr>
<td>01010</td>
<td>3</td>
<td>01010</td>
<td>Data 3</td>
</tr>
<tr>
<td>01011</td>
<td>4</td>
<td>01000</td>
<td>Data 4</td>
</tr>
<tr>
<td>01011</td>
<td>5</td>
<td>01011</td>
<td>Data 5</td>
</tr>
<tr>
<td>01110</td>
<td>6</td>
<td>01110</td>
<td>Data 6</td>
</tr>
<tr>
<td>01111</td>
<td>7</td>
<td>01111</td>
<td>Data 7</td>
</tr>
<tr>
<td>10010</td>
<td>8</td>
<td>10000</td>
<td>Data 8</td>
</tr>
<tr>
<td>10011</td>
<td>9</td>
<td>10010</td>
<td>Data 9</td>
</tr>
<tr>
<td>10110</td>
<td>A</td>
<td>10110</td>
<td>Data A</td>
</tr>
<tr>
<td>10111</td>
<td>B</td>
<td>10111</td>
<td>Data B</td>
</tr>
<tr>
<td>11010</td>
<td>C</td>
<td>11000</td>
<td>Data C</td>
</tr>
<tr>
<td>11011</td>
<td>D</td>
<td>11010</td>
<td>Data D</td>
</tr>
<tr>
<td>11100</td>
<td>E</td>
<td>11110</td>
<td>Data E</td>
</tr>
<tr>
<td>11101</td>
<td>F</td>
<td>11111</td>
<td>Data F</td>
</tr>
<tr>
<td>11111</td>
<td>I</td>
<td>undefined</td>
<td>IDLE: used as inter-stream fill code</td>
</tr>
<tr>
<td>11000</td>
<td>J</td>
<td>01010</td>
<td>Start-of-Stream Delimiter</td>
</tr>
<tr>
<td>10001</td>
<td>K</td>
<td>01011</td>
<td>Start-of-Stream Delimiter Ack &amp; Nack</td>
</tr>
<tr>
<td>01101</td>
<td>T</td>
<td>undefined</td>
<td>End-of-Stream Delimiter used in pairs</td>
</tr>
<tr>
<td>00111</td>
<td>R</td>
<td>undefined</td>
<td>End-of-Stream Delimiter</td>
</tr>
<tr>
<td>00100</td>
<td>H</td>
<td>undefined</td>
<td>Transmit Error; used to force signaling errors</td>
</tr>
<tr>
<td>11001</td>
<td>V</td>
<td>undefined</td>
<td>Invalid code</td>
</tr>
</tbody>
</table>

---

**Outline**

Motivation for AC signalling

- Square waves require broad bandwidth, e.g.:
  \[
  f(t) = \cos(2\pi f) - \frac{1}{2}\cos(2\pi f) + \frac{1}{4}\cos(2\pi f) - \frac{1}{8}\cos(2\pi f) + \ldots
  \]
- Alternative to DC signalling ("baseband modulation") is to modulate a carrier ("AC signalling")
- Carrier frequency can be chosen to match medium's frequency of low attenuation.
- May have multiple carriers at different frequencies.
- AC signalling is popular for modems operating over phone lines with narrow bandwidth.
**Multivalued signalling**

No need to restrict amplitude/frequency/phase to 2 values; can have more (noise levels permitting)

*e.g. Quadrature Phase Shift Keying: 4 different phase shifts (45, 135, 225, 315 degrees)*

Can also combine schemes, *e.g. PSK+ASK:*

*Modern symbol "constellations"*

\[ b^b: \text{V32} @ 9600b/s, \quad c^c: \text{V32bis} @ 14,400b/s \]

**Outline**

**Data rate vs baud rate vs bandwidth**

Units:
- **Data rate** (bits / second): Information carrying capacity
- **Baud rate** (symbols / second): How often the signal can vary
- **Bandwidth** (Hz): Range of frequencies that carry significant energy.

These measures are correlated, but are different:
- Each signal carries some bits ⇒ more signals = more bits
  *e.g. V32 modem: 2400 baud, each symbol carries 4b=9600b/s*
- Shannon's capacity formula relates data rate to bandwidth, but the relationship depends on the SNR.

The term "bandwidth" is often used to describe data rate, even though the technically correct term is "data rate".
### Framing

Physical layer provides a bit stream. Higher layers want to multiplex streams/flows of information flowing between different processes over a single physical layer. ⇒ need to identify where each flow starts/stops. ⇒ **Framing**: Divide a stream of bits transmitted by the Physical layer into "frames" (typically 10s to 1000s of bytes long)

Framing also helps with error control: more efficient to do error control over blocks of data (frames) than bits.

### Framing using length fields

First characters indicate how long frame is.
- Receiver can allocate buffer of appropriate size when first characters of frame arrive.
- Transmitter needs to know how long frame is before starting.
- Sensitive to transmission errors:
  - Error may change a length field, and so cause mis-interpretation of subsequent characters as length fields, continuing indefinitely.
  - Rarely used at link layer, but may be used at higher layers (e.g. IP) after link layer error detection.

#### Framing using special characters

Idea: Use a specific "flag" character (sequence) to indicate the start of a frame.
- Often also use a character at the end of the frame – so receiver can be sure about how to interpret signal after received bytes (e.g. continual 0V = string of 0s, or nothing?)

**e.g. ASCII character set:**
- printable characters (decimal 32-126)
- Start of Text (STX, 2)
- End of Text (ETX, 3)
- Often preface STX/ETX

Similarly, some Physics using line coding possess redundant symbols (e.g. Start-of-Stream and End-of-Stream Delimiters in 4b5b code)

† Sometimes called a "sentinel"  ‡ "Phy" is a common truncation of "Physical (Layer)"

### Stuffing

Problem with binary data: The frame may contain arbitrary strings of bits, including the flag string.
- e.g. FLAG H1 H2 A B C FLAG D E F FLAG

Don't want it to be interpreted as delineating 2 frames

**Solution:** **Stuffing:** When transmitting, stuff in extra characters where flag characters occur. Extra characters distinguish apparent flags in payload from genuine flags.
- e.g. stuff in an "Escape" character before apparent flag characters & before apparent escape characters.

**Issues:**
- Errors: Can convert payload into flag, or vice versa, but effect will only persist until the next flag.
- Extent to which stuffing expands the data depends on content
- Often scramble content to prevent long string of characters needing stuffing.
Byte stuffing

Byte stuffing: Readily manipulated by software → used for PPP

<table>
<thead>
<tr>
<th>FLAG</th>
<th>Header</th>
<th>Payload field</th>
<th>Trailer</th>
<th>FLAG</th>
</tr>
</thead>
</table>

\[ C \text{ programming analogy:} \]
\[ \text{printf}("\text{string}", \text{vars}...) \]
\[ = \text{flag character} \]
To include " within string, preface by escape character \\
\[ \text{Actually want to print escape character? Use escape-escape } \backslash \]

Often (in PPP and HDLC-like protocols) the flag character has the binary value 01111110

Bit stuffing

Goals: Want to avoid:
- Flag occurring within payload
- Long strings of 1s in payload (receiver may not observe transitions → may lose sync & sample at wrong instants)

- Preventing strings of 6 or more 1s will achieve both goals.
- Whenever transmitting 011111, follow this with 0 irrespective of next payload bit.
- Receiver: remove 0 immediately after 011111

Bit manipulations make this stuffing difficult in software → used in link layers intended for hardware implementation (e.g. HDLC, not PPP)

Original data: 011011111111111111110010

Data after stuffing (as transmitted): 011011111011110111115010010

Received data (after destuffing): 011011111111111111110010

† Similarly, might want to avoid long strings of 0s ⇒ (de)stuff 1s at appropriate times

Other framing techniques

Interframe spaces (Time gaps): Rarely used because clocks on different devices may differ slightly → hard to agree on what constitutes a sufficient gap.

Recurrence of redundancy, e.g. frames of known length L (e.g. ATM cell header always 5B):
1. Maintain running calculation of CRC over last L bits.
2. If looks like L bits contained valid CRC, probably found frame boundary.
3. Repeat a couple of times to be sure.

Often combine techniques, e.g. stuffing + character count to confirm