EE 122: Encoding And Framing

Ion Stoica
September 9, 2002
Overview

- Encoding
  - Framing
Encoding

- Goal: send bits from one node to another node on the same physical media
  - This service is provided by the physical layer
- Problem: specify a robust and efficient encoding scheme to achieve this goal

Adaptor: convert bits into physical signal and physical signal back into bits
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.
Non-Return to Zero (NRZ)

- 1 $\rightarrow$ high signal; 0 $\rightarrow$ 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
4-bit/5-bit

- Goal: address inefficiency of Manchester encoding, while avoiding long periods of low or high 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

<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>1111</td>
<td>01111</td>
<td>1111</td>
<td>11101</td>
</tr>
</tbody>
</table>
Overview

- Encoding
  - Framing
Framing

- Goal: send a block of bits (frames) between nodes connected on the same physical media
  - This service is provided by the data link layer
- Use a special byte (bit sequence) to mark the beginning (and the end) of the frame
- Problem: what happens if this sequence appears in the data payload?
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: 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)
- Example: SONET ST-1: 51.84 Mbps
Clock-Based Framing (SONET)

- 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
Summary

- Encoding – specify how bits are transmitted on the physical media
- Challenge – 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
- Framing – specify how blocks of data are transmitted
- Challenge
  - Decide when a frame starts/ends
  - Differentiate between the true frame delimiters and delimiters appearing in the payload data