Data Encoding

Encoding at the Physical Layer
Encoding at the Link Layer

Summary

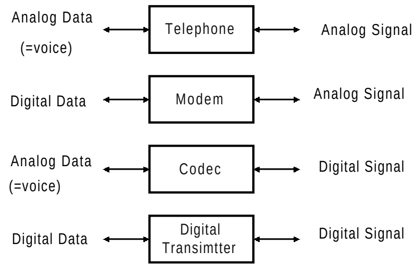
- What link bandwidth means – Fourier transform and frequency domain explanation
- What capacity means – How many bits per second I can send on a link of bandwidth H
- How noise affects link capacity

Data Encoding at the Physical Layer

- Data – Something that carries meaning
- Signal – Physical encoding of data

Digital Data Digital Signal	Digital Data Analog Signal
Analog Data Digital Signal	Analog Data Analog Signal

Analog vs. Digital Data/Signals



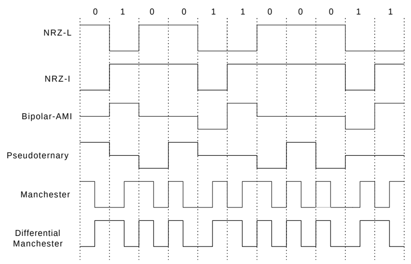
Analog Data (=voice) ↔ Telephone ↔ Analog Signal

Digital Data ↔ Modem ↔ Analog Signal

Analog Data (=voice) ↔ Codec ↔ Digital Signal

Digital Data ↔ Digital Transimtter ↔ Digital Signal

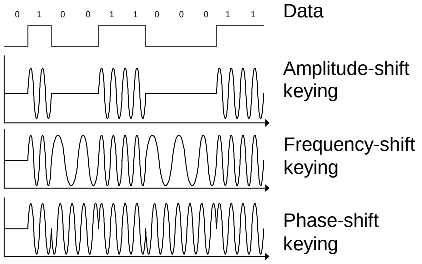
Digital Data / Digital Signal (Baseband – Short Distance)



Issues in Encoding

- Baseline wander
  - Too many consecutive 1s or 0s cause the receiver's baseline to drift
  - Compare NRZ, NRZI, Manchester?
- Clock synchronization
  - The absence of signal level transitions may cause clocks to drift
- Bit rate vs. Transition rate (Baud rate)
- 4B/5B Code.

Digital Data/Analog Signal (Broadband – Optical)



Analog Data/Digital Signal (Digital Mobile Phones)

- Pulse Code Modulation: PCM
- Example:
    - Voice has a bandwidth of about 4 kHz
    - Sampling rate must be 8000 samples/second

    - Typically, the sample size is 7-8 bits
    - Voice channel requires 56-64 kbps

    - PCM for voice is standardized as ITU-TS G.711
    - Note: ISDN B-channel has 64 kbps

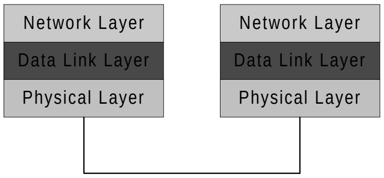
Analog Data/Analog Signal

- There are three types of modulation:
    - Amplitude Modulation (AM)
    - Frequency Modulation (FM)
    - Phase Modulation (PM)

Data Encoding at the Link Layer

Main Task of the data link layer:

- Provide error-free transmission over a link



Framing

- Problem: Identify the beginning and the end of a frame in a bit stream

01111110	Data	01111110

- Solutions:
    - Byte-oriented framing
    - Bit-oriented framing (HDLC)
    - Clock-based Framing (SONET)
- The frame identifying sequence must not occur in the data

Byte Oriented Framing

- Real-life example: PPP
- PPP Header:

01111110	Address FF	Control 03	protocol	payload	checksum	01111110

- Special code 01111110 (7E) denotes beginning and end of frame
    - 7E in the data is replaced by 7D-5E
    - 7D in the data is replaced by 7D-5D
- Receiver interprets 7D-5D as 7D, and 7D-5E as 7E.

Bit Oriented Framing (HDLC)

- HDLC = High-level Data Link Control
- Protocol views data as a stream of bits

01111110	header	body	CRC	01111110

- Special code 01111110 denotes the beginning and end of a frame
- If 11111 appears in body, a 0 is inserted
- At the receiver the extra 0 is removed

Clock-Based Framing (SONET)

- SONET is popular on optical backbone links of the OC-n category
- Full specs take entire book
- OC-1:
  - Each Frame is 810 bytes
  - Frame ends with special pattern
  - Receiver expects pattern to appear regularly every 810 bytes to lock on frame boundaries

Error Control

Two basic approaches to handle bit errors:

- Error-correcting codes
  - Used if retransmission of the data is not possible
  - Data are encoded with sufficient redundancy to correct bit errors
  - **Examples**: Hamming Codes, Reed Solomon Codes, etc.
- Error-detecting codes plus retransmission
  - Used if retransmission of corrupted data is feasible
  - Receiver detects error and requests retransmission of a frame.

Error Detection Techniques

- Error Detection Techniques:
  - Parity Checks
  - Cyclic Redundancy Check

Parity Checks

General Method:

- Append a parity bit to the end of each character in a frame such that the total number of '1' in a character is:
  - even (even parity) or
  - odd (odd parity)
- **Two-dimensional Parity**

Two-Dimensional Parity

0 1 1 0 1 1 0	0
0 1 0 0 1 0 0	0
1 1 0 1 0 1 1	1
1 1 1 1 0 0 0	0
0 1 1 1 0 0 1	0
0 0 0 1 0 0 0	1
0 0 1 0 1 0 0	0
0 1 0 0 1 0 0	0

Parity is made odd or even
In both dimensions

Single bit errors can be corrected

Two-bit errors can be detected

Odd-bit errors can be detected

What about 4,6,8,.. bit errors?

Cyclic Redundancy Check Background

- One’s complement arithmetic:
- Properties:
  - 1 – 1 = 1 + 1 = 0
  - 0 – 1 = 0 + 1 = 1
  - Implemented as an XOR operation
  - Y + 2 Z = Y + Z xor Z = Y

Cyclic-Redundancy Codes (CRC)

General Method:

- The transmitter generates an **n-bit check sequence number** from a given **k-bit frame** such that the resulting (k+n)-bit frame is divisible by some number
- The receiver divides the incoming frame by the same number
- If the result of the division does not leave a remainder, the receiver assumes that there was no error

CRC Encoding Method

- Let us view each block of data as a polynomial with binary coefficients:
  1 0 1 1 0 1   is viewed as $x^5 + x^3 + x^2 + 1$
- Let M(x) be the data block as a polynomial
- Let C(x) be the “Divisor Polynomial" which is known to both sender and receiver (degree of C(x) is k)

CRC Encoding Method

(I) Append k zeros to M(x), i.e., $M(x)\,x^k$

(II) Divide $M(x)\,x^k$ by C(x) and obtain:

$$M(x)\,x^k = Q(x)\,C(x) + R(x)$$

(III) Set $T(x) = M(x)\,x^k + R(x)$.

T(x) is the encoded message

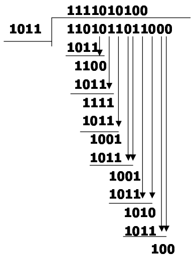
Note:

T(x) is divisible by C(x). Therefore, if the received message does not contain an error then it can be divided by C(x).

Example

- **Exercise:** Assume the divisor polynomial is $x^3 + x + 1$. Encode the frame:
  - **1 1 0 1 0 1 1 0 1 1**

Solution



```
          1111010100
1011    | 1101011011000
          1011
           1100
           1011
            1111
            1011
             1001
             1011
               1001
               1011
                1010
                1011
                   100
```

- **Encoded pattern = 1101011011100**

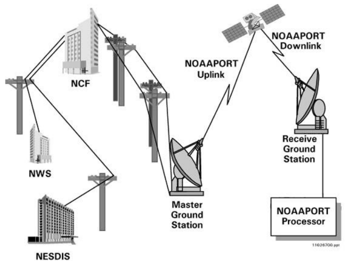
Common CRC Polynomials
(from Table 2.6)

- CRC-8: $x^8 + x^2 + x + 1$
- CRC-16: $x^{16} + x^{15} + x^2 + 1$
- CCITT: $x^{16} + x^{12} + x^5 + 1$
- CRC-32: $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$

Application Example

- National Oceanic and Atmospheric Administration (NOAA)

NOAA Communication Infrastructure: NOAAPORT



- NOAA Port uses a subset of HDLC for their link layer
  - Bit oriented framing
  - NRZ encoding
  - CRC-16

See http://205.156.54.206/noaaport/html/datalink.shtml

Error Correcting Codes

Hamming

Hamming Code

- Insert parity bits at positions 2^x, where $x=0,1,2,\ldots$
- Create parity groups. The kth group starts at bit k and includes every other group of k bits
- Each group includes only one parity bit. The bit is set to 1 or 0 to set group parity even.
- At the receiver, the parity of all groups is checked. Odd parity groups uniquely determine the position of the wrong bit.

Hamming Code by Example

- Encode 011110100

1	2	3	4	5	6	7	8	9	10	11	12	13
?	?	0	?	1	1	1	?	1	0	1	0	0

- Determine parity bit 1 (group 1)

1	2	3	4	5	6	7	8	9	10	11	12	13
?	?	0	?	1	1	1	?	1	0	1	0	0

↑
0

Example Continued

- Determine parity bit 2 (group 2)

1	2	3	4	5	6	7	8	9	10	11	12	13
0	?	0	?	1	1	1	?	1	0	1	0	0

1

- Determine parity bit 4 (group 4)

1	2	3	4	5	6	7	8	9	10	11	12	13
0	1	0	?	1	1	1	?	1	0	1	0	0

1

Example Continued

- Determine parity bit 8 (group 8)

1	2	3	4	5	6	7	8	9	10	11	12	13
0	1	0	1	1	1	1	?	1	0	1	0	0

0

- Encoded word:

1	2	3	4	5	6	7	8	9	10	11	12	13
0	1	0	1	1	1	1	0	1	0	1	0	0

At the Receiver

- If some groups have wrong parity, add up their numeric identifiers
- Result is the position of wrong bit