Chapter 3
Digital Transmission Fundamentals

- Digital Representation of Information
- Why Digital Communications?
- Digital Representation of Analog Signals
- Characterization of Communication Channels
- Fundamental Limits in Digital Transmission
- Line Coding
- Modems and Digital Modulation
- Properties of Media and Digital Transmission Systems
- Error Detection and Correction

Questions of Interest

- How long will it take to transmit a message?
  - How many bits are in the message (text, image)?
  - How fast does the network/system transfer information?
- Can a network/system handle a voice (video) call?
  - How many bits/second does voice/video require? At what quality?
- How long will it take to transmit a message without errors?
  - How are errors introduced?
  - How are errors detected and corrected?
- What transmission speed is possible over radio, copper cables, fiber, …?
A Transmission System

Transmitter
- Converts information into signal suitable for transmission
- Injects energy into communications medium or channel
  - Telephone converts voice into electric current
  - Modem converts bits into tones

Receiver
- Receives energy from medium
- Converts received signal into form suitable for delivery to user
  - Telephone converts current into voice
  - Modem converts tones into bits

Transmission Impairments

Communication Channel
- Pair of copper wires
- Coaxial cable
- Radio
- Light in optical fiber
- Light in air
- Infrared

Transmission Impairments
- Signal attenuation
- Signal distortion
- Spurious noise
- Interference from other signals
Analog Long-Distance Communications

Transmission segment

- Each repeater attempts to restore analog signal to its original form
- Restoration is imperfect
  - Distortion is not completely eliminated
  - Noise & interference is only partially removed
- Signal quality decreases with # of repeaters
- Communications is distance-limited
- Still used in analog cable TV systems
- Analogy: Copy a song using a cassette recorder

Analog vs. Digital Transmission

**Analog transmission**: all details must be reproduced accurately

**Digital transmission**: only discrete levels need to be reproduced

Simple Receiver: Was original pulse positive or negative?
Digital Long-Distance Communications

Transmission segment

- Regenerator recovers original data sequence and retransmits on next segment
- Can design so error probability is very small
- Then each regeneration is like the first time!
- Analogy: copy an MP3 file
- Communications is possible over very long distances
- Digital systems vs. analog systems
  - Less power, longer distances, lower system cost
  - Monitoring, multiplexing, coding, encryption, protocols...

Bit Rates of Digital Transmission Systems

<table>
<thead>
<tr>
<th>System</th>
<th>Bit Rate</th>
<th>Observations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Telephone twisted pair</td>
<td>33.6-56 kbps</td>
<td>4 kHz telephone channel</td>
</tr>
<tr>
<td>Ethernet twisted pair</td>
<td>10 Mbps, 100 Mbps</td>
<td>100 meters of unshielded twisted copper wire pair</td>
</tr>
<tr>
<td>Cable modem</td>
<td>500 kbps-4 Mbps</td>
<td>Shared CATV return channel</td>
</tr>
<tr>
<td>ADSL twisted pair</td>
<td>64-640 kbps in, 1.536-6.144 Mbps out</td>
<td>Coexists with analog telephone signal</td>
</tr>
<tr>
<td>2.4 GHz radio</td>
<td>2-11 Mbps</td>
<td>IEEE 802.11 wireless LAN</td>
</tr>
<tr>
<td>28 GHz radio</td>
<td>1.5-45 Mbps</td>
<td>5 km multipoint radio</td>
</tr>
<tr>
<td>Optical fiber</td>
<td>2.5-10 Gbps</td>
<td>1 wavelength</td>
</tr>
<tr>
<td>Optical fiber</td>
<td>&gt;1600 Gbps</td>
<td>Many wavelengths</td>
</tr>
</tbody>
</table>
Chapter 3
Digital Transmission Fundamentals

Properties of Media and Digital Transmission Systems

Fundamental Issues

- Propagation speed of signal
  - $c = 3 \times 10^8$ meters/second in vacuum
  - $\nu = c/\sqrt{\varepsilon}$ speed of light in medium where $\varepsilon > 1$ is the dielectric constant of the medium
  - $\nu = 2.3 \times 10^8$ m/sec in copper wire; $\nu = 2.0 \times 10^8$ m/sec in optical fiber
Twisted Pair

A twisted pair consists of two insulated copper wires, typically about 1mm thick

- More twists per cm leads to less crosstalk and better quality over longer distance

(a) Category 3 UTP (16 MHz).  (b) Category 5 UTP (100 MHz).

Twisted Pair Bit Rates

Table 3.5 Data rates of 24-gauge twisted pair

<table>
<thead>
<tr>
<th>Standard</th>
<th>Data Rate</th>
<th>Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>T-1</td>
<td>1.544 Mbps</td>
<td>16,000 feet, 5.5 km</td>
</tr>
<tr>
<td>DS2</td>
<td>6.312 Mbps</td>
<td>12,000 feet, 3.7 km</td>
</tr>
<tr>
<td>1/4 STS-1</td>
<td>12.960 Mbps</td>
<td>4500 feet, 1.4 km</td>
</tr>
<tr>
<td>1/2 STS-1</td>
<td>25.920 Mbps</td>
<td>3000 feet, 0.9 km</td>
</tr>
<tr>
<td>STS-1</td>
<td>51.840 Mbps</td>
<td>1000 feet, 300 m</td>
</tr>
</tbody>
</table>

- Twisted pairs can provide high bit rates at short distances
- Asymmetric Digital Subscriber Loop (ADSL)
  - High-speed Internet Access
  - Lower 3 kHz for voice
  - Upper band for data
  - 64 kbps inbound
  - 640 kbps outbound
- Much higher rates possible at shorter distances
  - Strategy for telephone companies is to bring fiber close to home & then twisted pair
  - Higher-speed access + video
**Ethernet LANs**

- Category 3 unshielded twisted pair (UTP): ordinary telephone wires
- Category 5 UTP: tighter twisting to improve signal quality
- Shielded twisted pair (STP): to minimize interference; costly
- 10BASE-T Ethernet
  - 10 Mbps, Baseband, Twisted pair
  - Two Cat3 pairs
  - Manchester coding, 100 meters
- 100BASE-T4 Fast Ethernet
  - 100 Mbps, Baseband, Twisted pair
  - Four Cat3 pairs
  - Three pairs for one direction at-a-time
  - 100/3 Mbps per pair;
  - 3B6T line code, 100 meters
- Cat5 & STP provide other options

**Coaxial Cable**

- A good combination of high bandwidth and excellent interference immunity
  - Higher bandwidth than twisted pair
  - Cable TV distribution
  - Long distance telephone transmission
  - Original Ethernet LAN medium
Coaxial Cable (Cont.)

- A constant bit rate video stream requires transmitting 30 screen images (frames) per second. The screen is 480 X 640 pixels, each pixel being 24 bits. How much bandwidth is needed for a coaxial cable?

Optical Fiber

- Light sources (lasers, LEDs) generate pulses of light that are transmitted on optical fiber
  - Very long distances (>1000 km)
  - Very high speeds (>40 Gbps/wavelength)
  - Nearly error-free (BER of $10^{-15}$)
- Profound influence on network architecture
  - Dominates long distance transmission
  - Distance less of a cost factor in communications
  - Plentiful bandwidth for new services
Optical Fiber Properties

Advantages
- Very low attenuation
- Noise immunity
- Extremely high bandwidth
- Security: very difficult to tap without breaking
- No corrosion
- More compact & lighter than copper wire

Disadvantages
- New types of optical signal impairments & dispersion
- Wavelength dependence
- Limited bend radius
- If physical arc of cable too high, light lost or won’t reflect
- Will break
- Difficult to splice
- Mechanical vibration becomes signal noise

Communication Satellites

Communication satellites and some of their properties, including altitude above the earth, round-trip delay time and number of satellites needed for global coverage.

Where are the 24 GPS satellites?
Chapter 3
Digital Transmission Fundamentals

Error Detection and Correction

Error Control

- Digital transmission systems introduce errors
- Applications require certain reliability level
  - Data applications require error-free transfer
  - Voice & video applications tolerate some errors
- Error control used when transmission system does not meet application requirement
- Error control ensures a data stream is transmitted to a certain level of accuracy despite errors
- Two basic approaches:
  - Error detection & retransmission
  - Error correction
**Codeword and Hamming Distance**

- A \( n \)-bit codeword: a frame of \( m \)-bit data plus \( k \)-bit redundant check bits (\( n = m + k \))
- The number of bit positions in which two codewords differ is called the Hamming distance.
  - Example: 10001001 and 10110001

**Key Idea**

- All transmitted data blocks (“codewords”) satisfy a pattern
  - If received block doesn’t satisfy pattern, it is in error
  - Redundancy(\( r \))
- Blindspot: when channel transforms a codeword into another codeword
Error Detecting Codes – Single Parity bit

- Parity bit: to make the number of 1 bits in a codeword even or odd ($k = 1$)
  - Examples

  Can a parity bit used to detect a single-bit error in a codeword?
  Can a parity bit used to detect a double-bit error in a codeword? Triple…?
  What is the hamming distance of the two codewords (before and after error)?
  Can a parity bit used to correct a single-bit error in a codeword?

  Parity bit used in ASCII code

How good is the single parity check code?

- **Redundancy**: Single parity check code adds 1 redundant bit per $m$ information bits:
  overhead = $1/(m + 1)$
- **Coverage**: all error patterns with odd # of errors can be detected
  - An error pattern is a binary $(m + 1)$-tuple with 1s where errors occur and 0’s elsewhere
  - Of $2^{k+1}$ binary $(m + 1)$-tuples, ½ are odd, so 50% of error patterns can be detected
- Is it possible to detect more errors if we add more check bits?
- Yes, with the right codes
What if bit errors are random?

- Many transmission channels introduce bit errors at random, independently of each other, and with probability $p$.
- Some error patterns are more probable than others:

$$P[10000000] = p(1-p)^7 = (1-p)^8 \left( \frac{p}{1-p} \right)$$

and

$$P[11000000] = p^2(1-p)^6 = (1-p)^8 \left( \frac{p^2}{1-p} \right)^2$$

- In any worthwhile channel $p < 0.5$, and so $(p/(1-p) < 1)$.
- It follows that patterns with 1 error are more likely than patterns with 2 errors and so forth.
- What is the probability that an undetectable error pattern occurs?

<table>
<thead>
<tr>
<th>Single parity check code with random bit errors</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Undetectable error pattern if even # of bit errors:</td>
</tr>
<tr>
<td>$$P[\text{error detection failure}] = P[\text{undetectable error pattern}] = P[\text{error patterns with even number of 1s}] = \binom{n}{2} p^2(1-p)^{n-2} + \binom{n}{4} p^4(1-p)^{n-4} + \ldots$$</td>
</tr>
<tr>
<td>- Example: Evaluate above for $n = 32, p = 10^{-3}$</td>
</tr>
<tr>
<td>$$P[\text{undetectable error}] = \binom{32}{2} (10^{-3})^2 (1-10^{-3})^{30} + \binom{32}{4} (10^{-3})^4 (1-10^{-3})^{28}$$</td>
</tr>
<tr>
<td>$$\approx 496 (10^{-6}) + 35960 (10^{-12}) \approx 4.96 (10^{-4})$$</td>
</tr>
<tr>
<td>- For this example, roughly 1 in 2000 error patterns is undetectable</td>
</tr>
</tbody>
</table>
Two-Dimensional Parity Check

- More parity bits to improve coverage
- Arrange information as columns
- Add single parity bit to each column
- Add a final “parity” column
- Used in early error control systems

![Parity Check Matrix]

What is its error-detecting capability? How about its error-correction capability?

Bottom row consists of check bit for each column

Other Error Detection Codes

- Many applications require very low error rate
- Need codes that detect the vast majority of errors
- Single parity check codes do not detect enough errors
- Two-dimensional codes require too many check bits
- The following error detecting codes used in practice:
  - Internet Check Sums
  - CRC Polynomial Codes
**Internet (IP) Checksum**

- Let *IP header* consist of \( L \), 16-bit words, \( b_0, b_1, b_2, \ldots, b_{L-1} \)
- The algorithm appends a 16-bit checksum \( b_L \)
  - The checksum \( b_L \) is calculated as follows:
    - Treating each 16-bit word as an integer, find \( x = (b_0 + b_1 + b_2 + \ldots + b_{L-1}) \mod 2^{16}-1 \)
    - The checksum is then given by: \( b_L = -x \)
  - Thus, the headers must satisfy the following pattern:
    \[ 0 = (b_0 + b_1 + b_2 + \ldots + b_{L-1} + b_L) \mod 2^{16}-1 \]
- The checksum calculation is carried out in software using one’s complement arithmetic
Internet Checksum Example

Assume 4-bit words
Use mod 2^4-1 arithmetic

\[ b_0 = 1100 = 12 \]
\[ b_1 = 1010 = 10 \]

Use Modulo Arithmetic

Use Binary Arithmetic

Note 16 mod 15 = 1
So: 10000 mod 15 = 0001
leading bit wraps around

\[ b_0 + b_1 = 1100 + 1010 \]
\[ = 10110 \]
\[ = 10000 + 0110 \]
\[ = 0001 + 0110 \]
\[ = 0111 \]
\[ = 7 \]

Take 1's complement
\[ b_2 = -0111 = 1000 \]
Polynomial Codes

- Polynomials instead of vectors for codewords
- Polynomial arithmetic instead of check sums
- Implemented using shift-register circuits
- Also called cyclic redundancy check (CRC) codes
- Most data communications standards use polynomial codes for error detection
- Polynomial codes also basis for powerful error-correction methods

Binary Polynomial Arithmetic

- Binary vectors map to polynomials

\[(i_{k-1}, i_{k-2}, \ldots, i_2, i_1, i_0) \rightarrow i_{k-1}x^{k-1} + i_{k-2}x^{k-2} + \ldots + i_2x^2 + i_1x + i_0\]

Binary Addition:

\[(x^7 + x^6 + 1) + (x^6 + x^5)\]

Binary Multiplication:

\[(x + 1) (x^2 + x + 1)\]
Binary Polynomial Arithmetic

- Binary vectors map to polynomials

\[(i_{k-1}, i_{k-2}, \ldots, i_2, i_1, i_0) \rightarrow i_{k-1}x^{k-1} + i_{k-2}x^{k-2} + \ldots + i_2x^2 + i_1x + i_0\]

Addition:

\[(x^7 + x^6 + 1) + (x^6 + x^5) = x^7 + x^6 + x^6 + x^5 + 1\]

\[= x^7 + (1+1)x^6 + x^5 + 1\]

\[= x^7 + x^5 + 1 \text{ since } 1+1 \mod 2 = 0\]

Multiplication:

\[(x + 1) (x^2 + x + 1) = x(x^2 + x + 1) + 1(x^2 + x + 1)\]

\[= (x^3 + x^2 + x) + (x^2 + x + 1)\]

\[= x^3 + 1\]

Error-Detecting Codes

- CRC base

- Cyclic Redundancy Check (CRC) use polynomial code, which is based on treating bit strings as representation of polynomials with coefficients of 0 and 1 only.

- A \(k\)-bit frame is regarded as the coefficient list for a polynomial with \(k\) terms, ranging from \(x^{k-1}\) to \(x^0\). Such a polynomial is said to be of degree \(k-1\)

  Example: 110001
  What is its degree?
  What are its polynomial and coefficients?

- Polynomial arithmetic is done by per-bit XOR

  Example: 10011011 + 11001010
  \[11110000 - 10100110\]
CRC Idea

- Both the sender and the receiver agree upon a generator polynomial $G(x)$ as $1$ xxx…x $1$ in advance. Given a frame of $m$ bits (a polynomial $M(x)$), the idea of CRC is to append a checksum to the end of the frame in such a way that the polynomial represented by the checksummed frame is divisible by $G(x)$. When the receiver gets the checksummed frame, it tries dividing it by $G(x)$. If there is a reminder, there has been a transmission error.

What kind of errors can be detected?
How the checksum is calculated?

CRC Algorithm

- Let $r$ be the degree of $G(x)$. Append $r$ 0s to the low-order end of the frame, resulting in $x^rM(x)$.
- Divide the bit string of $G(x)$ into the bit string of $x^rM(x)$, using modulo 2 division.
- Subtract the remainder from the bit string of $x^rM(x)$ using modulo 2 subtraction. The result is the checksummed frame to be transmitted, called $T(x)$.

$T(x)$ must be divisible by $G(x)$!
CRC Example

- Frame: 1101011011
- Generator: 10011

What is the generator polynomial?

What is the actual frame to be transmitted?

If the third/fourth bit from the left is inverted during transmission, how this error is detected (or not detected) at the receiver’s end?

CRC Algorithm

Frame: 1101011011
Generator: 10011
Message after 4 zero bits are appended: 11010110110000
CRC Analysis

- What kind of errors will be detected?
- Imagine that a transmission error occurs, so that instead of $T(x)$ arriving, $T(x) + E(x)$ arrives. Each 1 bit in $E(x)$ corresponds to a bit that has been inverted.
  
  Example: 11001 (sent) ---- > 10101 (received)

  If $E(x)$ is divisible by $G(x)$, the error will slip by! So, how we select $G(x)$?!

Designing good polynomial codes

- Select generator polynomial so that likely error patterns are not multiples of $g(x)$
- Detecting Single Errors
  - $e(x) = x^i$ for error in location $i + 1$
  - If $g(x)$ has more than 1 term, it cannot divide $x^i$
- Detecting Double Errors
  - $e(x) = x^i + x^j = x^j(x^{j-i} + 1)$ where $j > i$
  - If $g(x)$ has more than 1 term, it cannot divide $x^i$
  - If $g(x)$ is a primitive polynomial, it cannot divide $x^{m+1}$ for all $m < 2^{n-k} - 1$ (Need to keep codeword length less than $2^{n-k} - 1$)
    - $x^{15} + x^{14} + 1$ won’t divide $x^k + 1$ for $k < 32, 768$
  - Primitive polynomials can be found by consulting coding theory books
Designing good polynomial codes

- **Detecting Odd Numbers of Errors**
  - Suppose all codeword polynomials have an even number of 1s, then all odd numbers of errors can be detected.
  - As well, \( b(x) \) evaluated at \( x = 1 \) is zero because \( b(x) \) has an even number of 1s.
  - This implies \( x + 1 \) must be a factor of all \( b(x) \).
  - Pick \( g(x) = (x + 1) p(x) \) where \( p(x) \) is primitive.

Standard Generator Polynomials

- **CRC-8:**
  \[ = x^3 + x^2 + x + 1 \]
  ATM

- **CRC-16:**
  \[ = x^{16} + x^{15} + x^2 + 1 \]
  \[ = (x + 1)(x^{15} + x + 1) \]
  Bisync

- **CCITT-16:**
  \[ = x^{16} + x^{12} + x^5 + 1 \]
  HDLC, XMODEM, V.41

- **CCITT-32:**
  \[ = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^7 + x^6 + x^5 + x^4 + x^2 + x + 1 \]
  IEEE 802, DoD, V.42
Error Correcting Codes – An Error-correcting Code

- Given a complete list of the valid codewords, the minimum hamming distance of any two codewords is the hamming distance of the complete code.
- Example: a complete code with four legal codewords of 0000000000, 0000011111, 1111100000, 1111111111
  - What is the hamming distance of the code?
  - How many error-bits at most can it correct?
  - How many error-bits at most can it detect?
  - What is the hamming distance of a code with a parity bit?
  - What is the relationship between the hamming distance and the number of error-bits to be detected and corrected?

Error Correcting Codes – Low Limit on k

- A $n$-bit codeword: a frame of $m$-bit data plus $k$-bit redundant check bits ($n = m + k$).
- What is the lower limit on the number of bits $k$ needed to correct single-bit errors in a $n$-bit codeword?
  \[(n+1) 2^m \leq 2^n\]
Hamming Method

- A \(n\)-bit codeword: a frame of \(m\)-bit data plus \(k\)-bit redundant check bits (\(n = m + k\))
- Use of a Hamming code to detect & correct a single-bit error in a codeword
  - The bits that are powers of 2 are used as check bits.
  - The rest are filled up with the data bits
  - Each check bit forces the parity of some collection of bits, including itself, to be even (or odd)
  - To see which check bits the data bit in position \(k\) contributes to, rewrite \(k\) as a sum of powers of 2
  - A bit is checks by just those check bits occurring in its expansion (\(11 = 1 + 2 + 8\))

Hamming Example

- Example: a \(n\)-bit codeword containing a 7-bit data 1001000

\[
1001000 \rightarrow 00110010000 \text{ (even-parity used)}
\]

How to correct it if 00100010000 is received instead?

How to correct it if 00110010001 is received instead?

How many check bits needed to d&c a single error in a 10-bit message?
Error-Correcting Codes – Burst Errors

- What to do if errors come in burst, instead of isolated single-bit errors?

<table>
<thead>
<tr>
<th>Char.</th>
<th>ASCII</th>
<th>Check bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>H</td>
<td>1001000</td>
<td>00110010000</td>
</tr>
<tr>
<td>a</td>
<td>1100001</td>
<td>10111001001</td>
</tr>
<tr>
<td>m</td>
<td>1101101</td>
<td>11101010101</td>
</tr>
<tr>
<td>i</td>
<td>1101001</td>
<td>01101011001</td>
</tr>
<tr>
<td>n</td>
<td>1101110</td>
<td>01110101100</td>
</tr>
<tr>
<td>g</td>
<td>1100111</td>
<td>01110011111</td>
</tr>
<tr>
<td>c</td>
<td>1100011</td>
<td>10011000000</td>
</tr>
<tr>
<td>o</td>
<td>1101111</td>
<td>11110000011</td>
</tr>
<tr>
<td>d</td>
<td>1100100</td>
<td>10101011111</td>
</tr>
<tr>
<td>e</td>
<td>1100101</td>
<td>00111000010</td>
</tr>
</tbody>
</table>

Order of bit transmission

What is the maximum length of a burst that can be corrected in a sequence of k codewords?

Error Detecting Codes vs. Error Correcting Codes

- Consider a channel on which errors are isolated and the error rate is $10^{-6}$. Let the block size ($m$) be 1000.

How many total bits required to provide single-bit error corrections for 1 MB data?

How many total bits required to provide the error detection + retransmission?

Why wireless networks prefer error correction while wired networks may go for error detection and retransmission?

What kind of applications prefer error correction instead of detection?