Lecture 3: Coding and Error Control
Coping with Data Transmission Errors

- **Error detection codes**
  - Detects the presence of an error

- **Automatic repeat request (ARQ) protocols**
  - Block of data with error is discarded
  - Transmitter retransmits that block of data

- **Error correction codes, or forward correction codes (FEC)**
  - Designed to detect and correct errors
Error Detection Probabilities

Definitions

- $P_b$: Probability of single bit error (BER)
- $P_1$: Probability that a frame arrives with no bit errors
- $P_2$: While using error detection, the probability that a frame arrives with one or more undetected errors
Error Detection Probabilities

- With no error detection

\[ P_1 = (1 - P_b)^F \]
\[ P_2 = 1 - P_1 \]

- F = Number of bits per frame
Error Detection Process

- **Transmitter**
  - For a given frame, an error-detecting code (check bits) is calculated from data bits
  - Check bits are appended to data bits

- **Receiver**
  - Separates incoming frame into data bits and check bits
  - Calculates check bits from received data bits
  - Compares calculated check bits against received check bits
  - Detected error occurs if mismatch
Error Detection Process

(a) Sender

(b) Receiver

Figure 8.1 Error Detection Process
Parity Check

- Parity bit appended to a block of data
- Even parity
  - Added bit ensures an even number of 1s
- Odd parity
  - Added bit ensures an odd number of 1s
- Example, 7-bit character [1110001]
  - Even parity [11100010]
  - Odd parity [11100011]
Cyclic Redundancy Check (CRC)

- **Transmitter**
  - For a \( k \)-bit block, transmitter generates an \((n-k)\)-bit frame check sequence (FCS)
  - Resulting frame of \( n \) bits is exactly divisible by predetermined number

- **Receiver**
  - Divides incoming frame by predetermined number
  - If no remainder, assumes no error
CRC-Binary Check Digit Method

- **Check Digit Method**
  Make number divisible by 9
- **Example: 823 is to be sent**
  1. Left-shift: 8230
  2. Divide by 9, find remainder: 4
  3. Subtract remainder from 9: 9-4=5
  4. Add the result of step 3 to step 1: 8235
  5. Check that the result is divisible by 9.
- Detects all single-digit errors: 7235, 8335, 8255, 8237
- Detects several multiple-digit errors: 8765, 7346
- Does not detect some errors: 7335, 8775, ...
CRC using Modulo 2 Arithmetic

- Exclusive-OR (XOR) operation
- Parameters:
  - $T = n$-bit frame to be transmitted
  - $D = k$-bit block of data; the first $k$ bits of $T$
  - $F = (n - k)$-bit FCS; the last $(n - k)$ bits of $T$
  - $P = \text{pattern of } n-k+1 \text{ bits}; \text{this is the predetermined divisor}$
  - $Q = \text{Quotient}$
  - $R = \text{Remainder}$
CRC using Modulo 2 Arithmetic

- For $T/P$ to have no remainder, start with

- Divide $2^{n-k}D$ by $P$ gives quotient and remainder

$$T = 2^{n-k}D + F$$

- Use remainder as FCS

$$\frac{2^{n-k}D}{P} = Q + \frac{R}{P}$$

$$T = 2^{n-k}D + R$$
CRC using Modulo 2 Arithmetic

- Does $R$ cause $T/P$ have no remainder?

  Substituting,
  \[
  \frac{T}{P} = \frac{2^{n-k}D + R}{P} = \frac{2^{n-k}D}{P} + \frac{R}{P}
  \]

- No remainder, so $T$ is exactly divisible by $P$

  \[
  \frac{T}{P} = Q + \frac{R}{P} + \frac{R}{P} = Q + \frac{R + R}{P} = Q
  \]
**Binary Check Digit Method**

Make number divisible by \( P = 110101 \) \((n-k+1=6 \text{ bits})\)

**Example:** \( D = 1010001101 \) is to be sent

1. Left-shift \( D \) by \( n-k \) bits \( 2^{(n-k)} \) \( D = 101000110100000 \)

2. Divide \( 2^{(n-k)} \) \( D \) by \( P \), find remainder: \( R = 01110 \)

3. Subtract remainder from \( P \)

4. Add the result of step 2 to step 1:
   \( T = 101000110101110 \)

5. Check that the result \( T \) is divisible by \( P \).
Modulo 2 Division

\[ Q = \overline{1101010110} \]
\[ P = \overline{110101} ) \overline{1010001110100000} = 2^n M \]

\[
\begin{array}{c}
110101 \\
110101 \\
011101 \\
000000 \\
111010 \\
110101 \\
011111 \\
000000 \\
111110 \\
110101
\end{array}
\quad \begin{array}{c}
010110 \\
000000 \\
101100 \\
110101 \\
110010 \\
110101 \\
001110 \\
000000 \\
011110 = \text{R}
\end{array}
\]
Checking at The Receiver

\[
\begin{align*}
1101010110 \\
110101)101000110101110 \\
110101 \\
111011 \\
110101 \\
011101 \\
000000 \\
111010 \\
110101 \\
011111 \\
000000 \\
111110 \\
110101
\end{align*}
\]
Practice

- $P=1100101$?
Wireless Transmission Errors

- Error detection requires retransmission

- Detection inadequate for wireless applications
  - Error rate on wireless link can be high, results in a large number of retransmissions
  - Long propagation delay compared to transmission time
Block Error Correction Codes

Transmitter
- Forward error correction (FEC) encoder maps each k-bit block into an n-bit block codeword
- Codeword is transmitted; analog for wireless transmission

Receiver
- Incoming signal is demodulated
- Block passed through an FEC decoder
Forward Error Correction Process

(a) Sender

(b) Receiver

Figure 8.5  Forward Error Correction Process
FEC Decoder Outcomes

- No errors present
  - Codeword produced by decoder matches original codeword

- Decoder detects and corrects bit errors

- Decoder detects but cannot correct bit errors; reports uncorrectable error

- Decoder detects no bit errors, though errors are present
Block Code Principles

- Hamming distance – for 2 $n$-bit binary sequences, the number of different bits
  - E.g., $v_1=011011; v_2=110001; d(v_1, v_2)=3$

- Redundancy – ratio of redundant bits to data bits

- Code rate – ratio of data bits to total bits

- Coding gain – the reduction in the required $E_b/N_0$ to achieve a specified BER of an error-correcting coded system
Error Correction Example

- 2-bit words transmitted as 5-bit/word

<table>
<thead>
<tr>
<th>Data Codeword</th>
<th>Transmitted</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>00000</td>
</tr>
<tr>
<td>01</td>
<td>00111</td>
</tr>
<tr>
<td>10</td>
<td>11001</td>
</tr>
<tr>
<td>11</td>
<td>11110</td>
</tr>
</tbody>
</table>

- Received = 00100 ⇒ Not one of the code words ⇒ Error
- Distance (00100, 00000) = 1 Distance (00100, 00111) = 2
  Distance (00100, 11001) = 4 Distance (00100, 11110) = 3
- a. Most likely 00000 was sent. Corrected data = 00
- b. Received = 01010 Distance(.,00000) = 2 = Distance(.,11110)
  Error detected but cannot be corrected
- c. Three bit errors will not be detected. Sent 00000, Received 00111
Hamming Code

- Designed to correct single bit errors

- Family of \((n, k)\) block error-correcting codes with parameters:
  - Block length: \(n = 2^m - 1\)
  - Number of data bits: \(k = 2^m - m - 1\)
  - Number of check bits: \(n - k = m\)

- Single-error-correcting (SEC) code
  - SEC double-error-detecting (SEC-DED) code
Hamming Code Process

- **Encoding**: $k$ data bits + $(n - k)$ check bits

- **Decoding**: compares received $(n - k)$ bits with calculated $(n - k)$ bits using XOR
  - Resulting $(n - k)$ bits called *syndrome word*
  - Syndrome range is between 0 and $2^{(n-k) - 1}$
  - Each bit of syndrome indicates a match (0) or conflict (1) in that bit position
For positive pair of integers $m$ and $t$, a $(n, k)$ BCH code has parameters:

- Block length: $n = 2^m - 1$
- Number of check bits: $n - k \leq mt$

- Correct combinations of $t$ or fewer errors
- Flexibility in choice of parameters
  - Block length, code rate
Reed-Solomon Codes

- Subclass of nonbinary BCH codes
- Data processed in chunks of $m$ bits, called symbols
- An $(n, k)$ RS code has parameters:
  - Symbol length: $m$ bits per symbol
  - Block length: $n = 2^m - 1$ symbols = $m(2^m - 1)$ bits
  - Data length: $k$ symbols
  - Size of check code: $n - k = 2t$ symbols = $m(2t)$ bits
Automatic Repeat Request

- Mechanism used in data link control and transport protocols
  - Relies on use of an error detection code (such as CRC)
  - Flow Control
  - Error Control
Flow Control

- Assures that transmitting entity does not overwhelm a receiving entity with data
- Protocols with flow control mechanism allow multiple PDUs in transit at the same time
- PDUs arrive in same order they’re sent
- Sliding-window flow control
  - Transmitter maintains list (window) of sequence numbers allowed to send
  - Receiver maintains list allowed to receive
Flow Control

- Reasons for breaking up a block of data before transmitting:
  - Limited buffer size of receiver
  - Retransmission of PDU due to error requires smaller amounts of data to be retransmitted
  - On shared medium, larger PDUs occupy medium for extended period, causing delays at other sending stations
Figure 8.17  Sliding-Window Depiction

(a) Sender's perspective

- PDU sequence number
- Last PDU acknowledged
- Last PDU transmitted

(b) Receiver's perspective

- Last PDU acknowledged
- Last PDU received

Window shrinks from trailing edge as PDUs are transmitted.
Window expands from leading edge as ACKs are received.

PDUs already transmitted
PDUs already received
Window of PDUs that may be transmitted
Window of PDUs that may be accepted
Error Control

- Mechanisms to detect and correct transmission errors
- Types of errors:
  - Lost PDU: a PDU fails to arrive
  - Damaged PDU: PDU arrives with errors
Error Control Requirements

- **Error detection**
  - Receiver detects errors and discards PDUs

- **Positive acknowledgement**
  - Destination returns acknowledgment of received, error-free PDUs

- **Retransmission after timeout**
  - Source retransmits unacknowledged PDU

- **Negative acknowledgement and retransmission**
  - Destination returns negative acknowledgment to PDUs in error
Go-back-N ARQ

- **Acknowledgments**
  - RR = receive ready (no errors occur)
  - REJ = reject (error detected)

- **Contingencies**
  - Damaged PDU
  - Damaged RR
  - Damaged REJ
Go-back-N ARQ