

Ex No: 6

DATE: 28-08-25

HAMMING CODE FOR ERRORS

Aim:

Write a program to implement error detection and correction using hamming code concept. Make a test to input data stream and verify error correction feature.

Error correction at data link layer:

Hamming code is a set of error correction codes that can be used to detect and correct the errors that can occur when the data is transmitted from the sender to the receiver. It is a technique developed by R.W Hamming for error correction.

Create sender program:

1. Input to select file should be a text of any length. Program should convert the text to binary.
2. Apply hamming code concept on the binary data and add redundant bits to it.
3. Save the output in a file called channel.

Create a receiver program:

1. Receiver program should read the input from channel.
2. Apply hamming code on binary data & check errors.
3. If there is an error, display the position of error.
4. Else remove the redundant bits and convert the binary data to ascii and display the output.

Hamming code for sender:

def char to binary:

```
return format(ord(ch), '08b')
```

```
def hamming_encode(data):
```

```
    d1, d2, d3, d4 = [int(bit) for bit in data]
```

```
    p1 = d1 ^ d2 ^ d4
```

```
    p2 = d2 ^ d3 ^ d4
```

```
    p4 = d2 ^ d3 ^ d4
```

```
    return f"{p1}{p2}{d1}{p4}{d2}{d3}{d4}"
```

```
text = input("Enter text: ")
```

```
with open("channel.txt", "w") as f:
```

```
    for ch in text:
```

```
        bin_ch = char_to_binary(ch)
```

```
        for i in range(0, 8):
```

```
            code = hamming_encode(bin_ch[i: i+8])
```

```
            f.write(code)
```

```
print("Data written to channel.txt with Hamming code")
```

Hamming code for receiver:

```
def hamming_decode(code):
```

```
    b = [0] * [int(bit) for bit in code]
```

```
    p1 = b[1] ^ b[3] ^ b[5] ^ b[7]
```

```
    p2 = b[2] ^ b[3] ^ b[6] ^ b[7]
```

```
    p4 = b[4] ^ b[5] ^ b[6] ^ b[7]
```

```
    error_pos = p1 * 1 + p2 * 2 + p4 * 4
```

```
    if error_pos != 0:
```

```
        print(f"Error detected at position {error_pos}.  
correcting...")
```

```
        b[error_pos] = 1 - b[error_pos]
```

```
        d1, d2, d3, d4 = b[3], b[5], b[6], b[7]
```

```
        return f"{d1}{d2}{d3}{d4}"
```

```
    binary_result = ""
```

```
    with open("channel.txt", "r") as f:
```

```
        code = f.read()
```

```
    for i in range(0, len(code), 8):
```

```
        binary_result += hamming_decode(code[i: i+8])
```

```
    text = ""
```

```
    for i in range(0, len(binary_result), 8):
```

```
        byte = binary_result[i: i+8]
```

```
        text += chr(int(byte, 2))
```

```
    print("Received text after error correction", text)
```

Input :

Enter 4 bit data: 1011

Sender side: 0010011

Receiver side: 0010011

Output :

Original data bits extracted: 1011

Result:

Sender and receiver program for hamming code concept was executed and got the output: