de? 2 of operation can intercept and packet that passes

18port layer header

have a transport at the link layer which is à communication ne metwork segment of is used by DNS? 18port Rayes im Protocof) control motocol) ed by Ellp protocol? oes used by lettp

hask as a packet Shudied about at various

08 °18 a network

28 communication

all devices

Experiment No: 6 Expos correction at Data Link Layer

Aim: (day' (a) had) tomber ) with the acoust Waite a program to implement 2208 detection and correction using Hamming code concept. Make a test sun to input and data stream and verify expos correction parture.

tages correction at Data link Layer:

Hamming code is a set of exers correction codes that can be used to detect and correct the exposs that can occur when the data is transmitted from the sends to the Receives. It is a technique developed by R.W. Hamming for esses correction.

cleate sender program with below features.

1. Input to sender gile should be a text of any length. Program should convert the text to binary.

2. Apply hamming code concept on the bin as data and add redundant bits on it.

3. Save this output in a file called chons create receives program with below features:

1. Receives program should read the Esput from channel file.

2. Apply hamming code on the binary dat to check for easoes.

3. If there is an exer, display the position of the expon

4. Else somewhere semove the sodundant bits and convert the binary data to asai and display the output.

```
deg string-to-binary (input-string):
    Paggam code:
                                                     calculat
       string-to-binary

seturn ' join (pormat (ord (c), '08b') for con input-string)
                                                     Lamming
                                                     n= 200(2
   deg binary-to-string (binary-data);
                                                    19 9 8 gr
                                                         pale
       gos ? En sange (o, son (bin asy-data), 8).
       chass=[]
                                                         parit
           byte = binary - data [:: 1+8]
           chass append (chs (int (byte, 2)))
        setus ". goin (chass)
def calculate parity - bits (data):
                                                         ha
                                                        Return
     n = len (data)
        REQUIRE Halfinghost & off.
                                                        detec
        while, (2** 9) < (n+9+1):
                                                         hami
           9+=1. 2003 PA POINT
       geturn a
      Ensert-parity-bits (data, 2):
       m= len (data) sabras of high
       3=90 mbspar Stpass pro 1
        R=0
        m=n+9 + a paramond plant
        hamming-code=[]
        for ? in sange (1, m+1):
           ° == 2 * * j:
              hamming - code, append (0)
               j+=1
           else:
             lamming-code append (int (data(k))
      sotus ". join (map(sts, hamming-code))
```

```
def calculate-parity-values (hamming-code, 2):
'086') for c in
                      hamming _ code = list (map ( int, hamming _ code))
uf string)
                  n= len (hamming code)
wantsta
               parity - pos = 2 * * ?
M Japan
-data), 8).
                        parity-val = 0
                 809 j'in Range (1, n+1):
e, 2)))
                  if j so parity:pos and j!=pority-pos:
parity-val 1= hamming-code [j-i]
1913-0570-
                         hamming - code [ parity - pos - 1] = parity - val
                       setus ". join (map(sts, Romming - code))
office si
                   def detect_ and- correct_ error (framming_code, 9):
                      hamming_code = list (map (int, hamming_code))
                       n = Den (hamming_ code)
               200 position = 0
                      pa i in Range (2)
                   (alal parity-p.08 = 2 * * 1,
                 parity - val = 0

808 j in Range (1, n+1):
3) tx31
                  of sp parity-pos:
                   parity_val 1 = hamming_code[;-1]
               "g parity_val != 0:
                2209-position += parity-pos
                      of esser-position:
                  ... print [g" Exer at : { 2008 - position y")
 mers
                    print-lg "Binary pos: {60 (0000- position)[2:]
- (data[k])
                    Ramming-code [exter-position -1] 1=1
                     print 18" corrected code: {".join(map
code))
                                    (sts, hamming_ode)) 3")
```

print ("NO 8888.") setuen ', join (map (ste, hamming code)) deg extract-data-grom-hamming (hamming.code, Res i en sange (1, len (hamming code) +1). 28 :1=2 \*\*9. - data append (hamming-code [;-1]) return ! join (map(str. data)) del main: input- string = input ("Enter the string:") binary-data = string to binary (input-8/8ing) print (f"Binary: & binary: data 3") 9 = calculate - pasity bits (binasy - data) hamming-code = insert-parity-bits (binary-data, hamming code = calculate pasity values 1. ( hamming code, 9) print (f"Hamming code: f hamming code 3") redundant bits= {2\*\*; for in Range (8) } while True: print ("In Flip a bit for 2888...") 2208 - bit = int linput ( P" Flip bit 11-18en (hamming code)3): ")) · if sedundant bits: · print (\* Redundant bot. choose

another position")

hamming-"i!" fran else '0') 4 print ( "Har break hamming-code= corrected bin are corrected strin print ( ?" Final of \_\_name\_\_ = " main () output: Enter the stair Binary: 010010 Hamming code Flip a bit for Flip bit (1-21) Redundant bi Flip a bit f Flip bit (1-21 taren! Hamming cod E228 at 3 Binaly pos: 0 consected cod to tinal outpu for implement es successful

else:

hamming - code) ing Chamming code, g) mming code) +1). ing-code [;-1]) data) the steing ; ") Dy linput- 8 12 ing) ata 3") aly-data) 1- bits (binaly-data, ty-values . code, 2) nming-code 3") in Range (8) 3 ("... Bees Flip bil-1): "))

inf bits:

hoose

hamming. code = hamming.code [: 2828\_bit-1] + (i! if hamming\_code [essos\_bit\_i] = 'o' else 'o') + hamming code [ esses bit] print (g"Hamming code with exer. {hamming\_code 3") break hamming-code = detect- and cossect- 200 (hamming-code, 9) corrected binary-data = extract-data- from homming (hamming - code, 9) corrected string = binary = to \_ string corrected - binary. print (8" Final output: 'fcorrected-stringy' ") ef \_\_name\_\_ = "\_\_ main\_\_": and the famus (Brint the Enter the string: Hi Binary: 010010000 1101001 Hamming code: 0100100001101001 Flip a bit por exer... Flip bit (1-21): 2 Redundant bit Choose another position Flip a bit for 2008 .... Flip bit (1-21): 3 toxe! Pages - Robert holder off o Hamming code with 2228: 01101001100001100100 F8208 at: 3 Binaly pos: 00011 cossected code: 010010011000011001001 e final output: Hi Thus, the program for Hamming code for implementing exxer detection and coxection is successfully executed and output is verified.